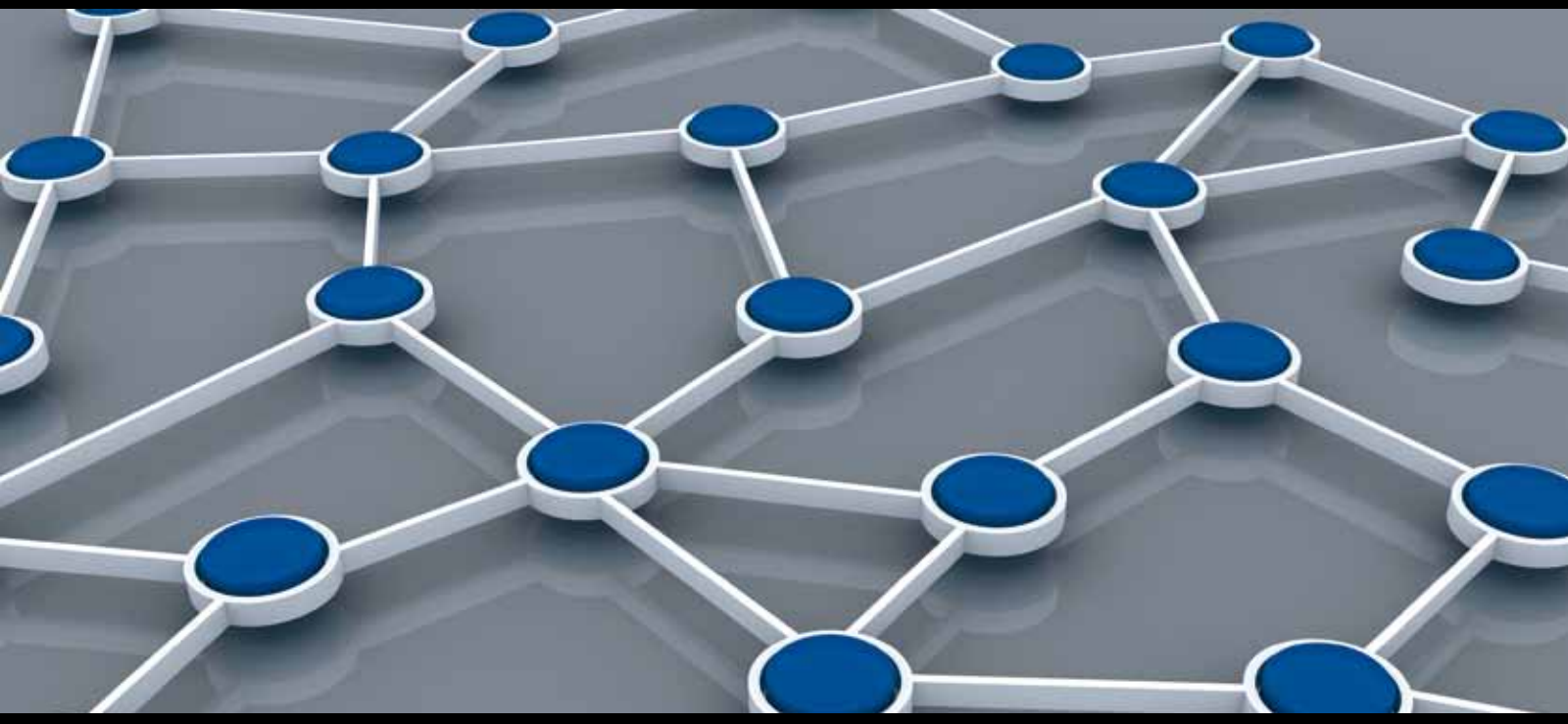
International Journal of Distributed Sensor Networks

Big Data in Ubiquitous Wireless Sensor Networks

Guest Editors: Fu Xiao, Chongsheng Zhang, and Zhijie Han



Big Data in Ubiquitous Wireless Sensor Networks

International Journal of Distributed Sensor Networks

Big Data in Ubiquitous Wireless Sensor Networks

Guest Editors: Fu Xiao, Chongsheng Zhang, and Zhijie Han

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 781729, 2 pages
http://dx.doi.org/10.1155/2014/781729

Editorial
Big Data in Ubiquitous Wireless Sensor Networks

Fu Xiao,[1] Chongsheng Zhang,[2] and Zhijie Han[3]

[1] *School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2] *Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim 7491, Norway*
[3] *School of Electronics, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*

Correspondence should be addressed to Fu Xiao; xiaof@njupt.edu.cn

Received 24 December 2013; Accepted 24 December 2013; Published 12 March 2014

Big data through cloud computing is an excellent way to offload significant amounts of computation and data from both data centers and terminal devices, due to its flexibility, scalability, and significant economic savings. However, cloud computing may not be suitable for all applications such as WSN (Wireless Sensor Network) for its high requirement on real time latency and response and geographically distributed and mobility support. For WSN, targets areas are close to the physical world, while cloud moves part of data storage and computation toward the edge of the network. This special issue aims at providing researchers and practitioners from the academia, industry, and government an opportunity to reflect upon these new developments of recent advances for big data in ubiquitous Wireless Sensor Networks and focus on the computation and storage, data analysis and mining, and distributed algorithms for data and computation placement, guaranteeing a global standard of quality and system integrity in Wireless Sensor Networks. We have accepted a few papers that address the state of the art and the future directions of these research and application areas for big data in ubiquitous Wireless Sensor Networks in this issue.

The paper "*Sparse signal recovery by stepwise subspace pursuit in compressed sensing*" describes an algorithm named stepwise subspace pursuit (SSP) for sparse signal recovery. This approach eliminates useless information from the candidate through threshold processing at first and then recovers the signal through the largest correlation coefficients. Experiments demonstrate that SSP significantly outperforms conventional techniques in recovering sparse signals.

The paper "*Coverage optimization algorithm based on sampling for 3D underwater sensor networks*" addresses the problem of maximizing the sensor field coverage for specific number of sensors while minimizing the distance traveled by the sensor nodes. Movement task as an optimization problem is defined to involve the adjustment of sensor node positions and coverage optimization algorithm based on sampling to enhance the coverage of 3D underwater sensor networks. A quadratic programming mathematical model is established to optimize the line segment coverage according to the intersection between sensing circles and line segments while minimizing the moving distance of the nodes.

The paper "*A coding and postprocessing framework of multiview distributed video for wireless video sensor networks*" proposes a new multiview video coding and postprocessing framework for multiview videos. The subsequent decoded videos of postprocessing simulation prove that the proposed postprocessing scheme can provide an additional improvement to the decoded video sequences.

The paper "*Hybrid ant algorithm based query processing with multiagents in sensor networks*" investigates query processing problem in resource-constrained Wireless Sensor Networks with two-tier architecture and multiple query agents, where multiple nodes of query agents are configured in the networks and corresponding source cluster-heads send collected events to only one optimum query agent. To reduce the energy consumption and shorten the delivery delay, an efficient query processing algorithm inspired by swarm intelligence of ants is introduced.

The paper "*A novel image mosaicking algorithm for wireless multimedia sensor networks*" focuses on how to take advantage of the cooperation among sensor nodes to obtain the panoramic information of scenic spots. And an image mosaic algorithm based on phase correlation and weighted average (IMBPW) is proposed. The IMBPW algorithm uses

the phase correlation based on Fourier transform to achieve registration of translation, rotation, and scaling images; adaptive weighted average algorithm is proposed to do the image fusion.

The paper "*Fast endmember extraction for massive hyperspectral sensor data on GPUs*" aims at proposing a method of parallel optimization for the well-known N-FINDR algorithm on graphics processing units (NFINDR-GPU) which is proposed to realize fast endmember extraction for massive hyperspectral sensor data. Experimental results show the effectiveness of NFINDR-GPU.

The paper "*A sociability-based spray and forward scheme for opportunistic network*" studies the nodes moving patterns when designing data transmission strategy. Based on the analysis of node moving characteristics in opportunistic social networks, social rang and social frequency are introduced and sociability-based Spray and Forward (SSF) scheme is proposed. Simulation results show that SSF can effectively improve data delivery ratio and reduce latency.

The paper "*A new metric for modeling the uneven sleeping problem in coordinated sensor node scheduling*" analyses the uneven sleeping problem (USP) happening in coordinated node scheduling in Wireless Sensor Networks. By analysing key factors leading to USP, a new metric is designed to measure and evaluate USP.

The paper "*Uncertainty-aware sensor deployment strategy in mixed wireless sensor networks*" studies the challenging problem of sensor deployment in a mixed sensor network where the mobile and static nodes work collaboratively to perform deployment optimization task. Gaussian white noise in environment is considered and a centralized algorithm is presented by using detection model based on false alarm and moves the mobile nodes based on bipartite graph matching.

The paper "*Research on migration strategy of mobile agent in wireless sensor networks*" investigates the migration schemes for mobile agents, determines core factors of migration strategy, and proposes SMLA and DMLA algorithms; advantages and disadvantages of the algorithms are evaluated by simulations.

The paper "*An approach based on chain key predistribution against Sybil attack in wireless sensor networks*" presents a chain key predistribution based approach against Sybil attack. Lightweight hash function and node-to-node chain key based authentication and exchange (CK-AE) protocol are proposed. Results show that this approach can not only enhance the ability of resilience to Sybil attack, but also reduce the communication overhead significantly at the cost of a certain amount of computational overhead.

The paper "*A new scheme based on pixel-intense motion block algorithm for residual distributed video coding*" proposes a new approach of residual coding combined with IMB (intense motion block) extraction algorithm with Low Density Parity Check (LDPC) and Baum-Welch iterative decoding algorithm. Experimental results show that the proposed scheme achieves better rate-distortion performance compared to traditional DVC and RDVC scheme.

The paper "*Multisensor data fusion for water quality evaluation using Dempster-Shafer evidence theory*" proposed a multisensor data fusion approach for water quality evaluation using Dempster-Shafer evidence theory. Each sensor measurement is considered as a piece of evidence. Based on quality parameters measured by sensor node, evidence from each sensor is given a reliability discounting and then combined with others by D-S rule. According to the decision rule, the class of water quality can be determined.

The paper "*Resilient multipath routing mechanism based on clustering with intrusion tolerance*" introduces the key management scheme after analyzing the security threats of LEACH protocol. Several related mechanisms to protect the communication among the sensor nodes and the confidentiality of the data transmission are adopted. Also, mixed multipath mechanism among the clusters is used to improve the security of the network and strengthen the intrusion tolerance.

The paper "*Data deduplication in wireless multimedia monitoring network*" applies data deduplication technology in Wireless Multimedia Monitoring Network. It eliminated redundant data from raw data to exploit network bandwidth efficiently. Moreover, a chunking algorithm with low computation complexity is presented.

The paper "*Research and design of an airfield runway FOD detection system based on WSN*" analyses Foreign object debris (FOD) surveillance system and introduces a designed method of airport runway FOD detection system based on WSN, and function of various image acquisition sensors is analyzed.

The paper "*A novel wireless sensor networks structure based on the SDN*" addresses the SDN (Software Defined Network) technology based on OpenFlow, and a new solution for WSNS in accordance with the OpenFlow technology is proposed.

The Paper "*Discovering human presence activities with smartphones using nonintrusive Wi-Fi sniffer sensors: the big data prospective*" describes an optimized Wi-Fi channel detection and selection method to switch between the best channels automatically to aggregate the Wi-Fi messages. Efficiency is demonstrated by experiment deploying in real-world office environment.

Fu Xiao
Chongsheng Zhang
Zhijie Han

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 874047, 7 pages
http://dx.doi.org/10.1155/2014/874047

Research Article

A Novel Wireless Sensor Networks Structure Based on the SDN

Zhi-jie Han and Wanli Ren

College of Computer & Information Engineering, Henan University, Kaifeng, Henan 475001, China

Correspondence should be addressed to Zhi-jie Han; hanzhijie@qq.com

Received 18 July 2013; Accepted 13 December 2013; Published 6 March 2014

Academic Editor: Chongsheng Zhang

Since the SDN (Software Defined Network) technology based on OpenFlow was born, domestic and foreign research institutions have had increasing attention to this emerging field. The OpenFlow technology which is known for standardization and compatibility network virtualization has brought a lot of innovative solutions in the field of research. As the wireless sensor network node is complex and difficult to move and the network topology is changing rapidly with time, this requires a routing protocol for it to be able to adapt to this high topology change. Because traditional WSNS protocol communication needs to consider energy consumption, load balancing and other factors, it is difficult to find one to take into account all aspects of network algorithms. In this paper, we put forward a new solution for WSNS in accordance with the OpenFlow technology research in the application of traditional network.

1. Introduction

With the development of wireless communications, electronics and sensing technology, wireless sensor network (WSN) has attracted widespread attention [1]. WSNs are composed of the sensor with functions of sensing, data processing, and short-distance wireless communication. In military defense, disaster relief, environmental monitoring, biological and commercial applications, and other fields have broad application prospects [2].

1.1. WSN Routing Technology. Routing protocol is one of key technologies in wireless sensor network and is presently one of the hotspot research problems. In multiple hops network, source node cannot send the data packet to the destination node directly and only leans on the help of intermediate nodes to forward these grouped data. In the network intermediate nodes (including source node) must correctly determine which neighboring node to send the received data packets to and to find the shortest path to enable it to reach destination along the shortest path [3].

Traditional ad hoc network focused on high-quality and efficient communication; therefore, the design goal of routing protocol is to find the shortest communication delay between two nodes, to improve the utilization of network bandwidth, and thus to avoid traffic congestion caused by routing loops and at the same time to control the number of intermediate forwarding about multihop communication and balance network traffic.

WSN have the traditional characteristics of the wireless network does not exist. It is far from the center of the network, in the architecture, addressing method, the communication mode, and routing structures which are different from the traditional network. Compared with common networks, sensor networks have more nodes. The nodes is very difficult to move but its network topology showed a strong nonstationary, which requested that its high routing protocol must be able to adapt to the topology changes [4].

WSN node has no energy supplement and its energy is constrained. In order to improve the network's lifetime, routing protocols must be involved in reducing energy consumption and load balancing. According to the literature [5–13] we classify routing protocols. Based on wireless sensor networks and the various similarities and differences between traditional networks, the existing routing protocols for wireless sensor networks have the following characteristics.

(1) Limited Energy. Wireless sensor network node is numerous, mobility is weak, and energy supplement cannot be timely, so the energy consumption is one of the challenges of wireless sensor network.

(2) Local Topology Information. Because the energy is limited, how to save energy and ensure the efficient of routing mechanism is a big challenge for WSN.

(3) High Topology Changes. Compared with common networks, sensor networks have more nodes. The nodes is very dofficult to move but its network topology showed a strong nonstationary, which requested that its high routing protocol must be able to adapt to the topology changes.

(4) Big Data Redundancy. In the existing wireless sensor network, due to the random distribution of nodes, Node has no special logo, multiple nodes simultaneously to collect the same data redundancy information is great.

(5) Application Specific. Applications of wireless sensor network environment vary widely, so there is no routing mechanism suitable for all applications: different applications need to adopt different systems.

1.2. Classification of WSN Routing Protocols

(1) According the number of paths in the transmission process, Routing protocols can be divided into single-path routing protocol and multipath routing protocol.

(2) According to the node differences in the hierarchy and the role during the routing process, Routing protocols can be divided into flat routing protocols and hierarchical routing protocols.

(3) According to the relationship between the route when established and data transmission Routing protocols can be divided into proactive routing protocols and hybrid routing protocols.

(4) According to whether the location is identified based on the destination and if the route calculation uses the location information, Routing protocols can be divided into location-based routing protocol and non-location-based routing protocol.

(5) According to whether they identify the destination by the daa, can be divided into routing protocols based on data and the routing protocols not based on data.

(6) According to whether the node addressing, whether to identify the destination addresses the routing protocols can be divided into data aggregation and non-data aggregation.

(7) According to the choice of routing whether considering the QoS constraints, the routing protocols can be divided into guaranteed QoS routing protocols and not guaranteed QoS routing protocols.

(8) According to whether the process of data transmission deals with aggregation, the routing protocols can be divided into data aggregation and non-data aggregation.

(9) According to whether the routing is designated by the source node, the routing protocols can be divided into source station and passive station.

(10) According to whether the routing time is involved in the query, the routing protocols can be divided into query driven routing protocols and non-query-driven routing protocols.

Because of the great number of sensor nodes and high density, sensor node energy, computing power, and storage capacity are limited; network topology changes frequently and has the self-organizing ability. When choosing wireless sensor network routing protocol we need to take into account the node energy, the communication load balancing, routing protocol fault tolerance, routing protocol security mechanisms, and so on. It is difficult to design a kind that has the most efficient wireless sensor network routing protocol.

2. The Key Components and Architecture of OpenFlow Based on SDN

The control logic of traditional Internet and data forwarding tight coupling device on the network, resulting in network control plane management of complex, but also makes updating and development of the network control plane is difficult to directly deploy new technologies on existing networks, flexibility and scalability difficult to adapt to the rapid development of the network. Network control forward separation architecture proposed by proprietary equipment deployment of high-level strategy, under the guidance of senior policy data forwarding, can be done. And it reduces the network equipment of many complicated functions, and at the same time, it can also improve the flexibility and operability of the network based on the new technology.This control logic control and data forwarding thinking separation technology research foundation SDN. Initially OpenFlow is put forward as a prototype of SDN [14], it mainly has two parts, OpenFlow switch and controller. OpenFlow switch forwarding packets according to the flow table represents the data forwarding plane; controller achieves the control function through the network view. The control logic represents the control plane [15].

The rapid development of the Internet can be attributed to the TCP/IP architecture and open application layer software design. However, due to the particularity of hardware devices, WSN is largely closed. In wireless sensor network, we put forward a model of wireless sensor network based on OpenFlow/SDN. The control function of the traditional distributed network equipment will be migrated to the controlled computing devices. The underlying network infrastructure can be abstracted by upper-layer network services and applications [15]. At last, it through the open programmable software pattern realizes the automation of network control function. This model consists of three basic roles: master node, center node, normal node. The master node as the controller of the network structure (programmable control unit of the central executive) according to the control unit (including topology, transmission capacity, and routing restrictions) will decide how to achieve synergy and interaction between nodes for themselves. The function of center node is similar to the OpenFlow switch. It is responsible for matching and

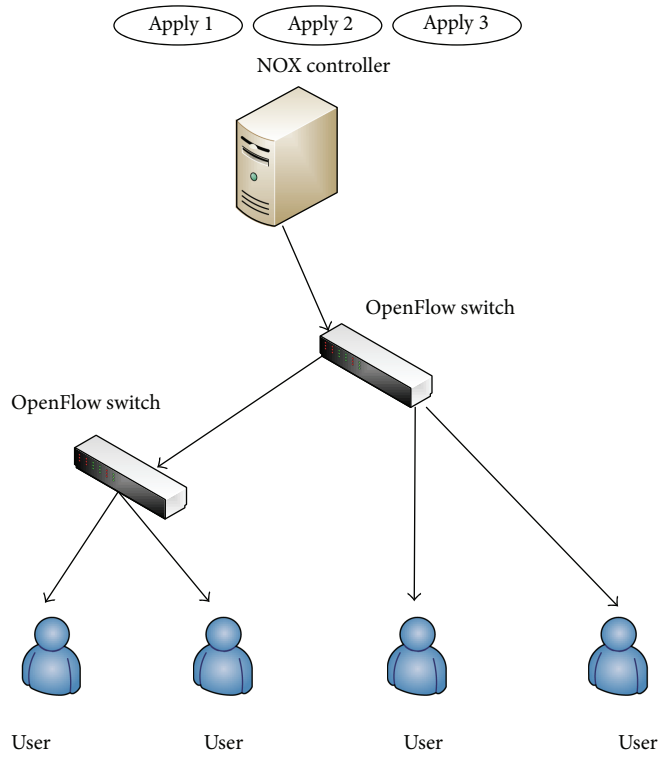


Figure 1: NOX-based OpenFlow network.

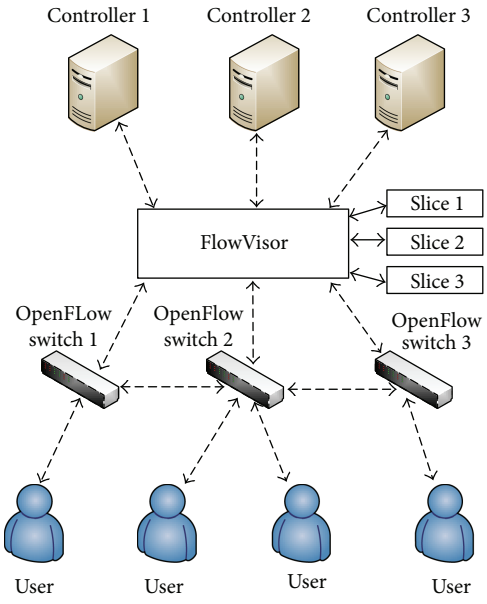


Figure 2: FlowVisor based OpenFlow virtualization.

forwarding data stream in the wireless sensor network. The responsibility of normal node is only to receive data flow [16].

2.1. Master Node/Controller. In the Software Defined Network, the controller is the core of the network NOS (network operating system) to realize control logic function. NOX is the programmable controlled central execution unit of the OpenFlow network. In fact, as shown in the Figure 1 in this case, the NOX is the control software of SDN. It can achieve different logic control functions by running different applications on NOX.

In wireless sensor network (WSN), we give the master node as a controller. In the WSN based on the OpenFlow/SDN, master node is the core of the network, and center node is the operating entity, NOX, by maintaining the basic information of the network view to maintain the entire network. Like topology, the network unit provides services. The application runs on the top of NOX by calling the global data network view to operate center node to the entire network. In the Software Defined Network [4].

In order to make the controller able to be directly deployed in real networks and solve the problem of multiple controllers to OpenFlow switch control sharing, at the same time, as shown in Figure 2 the controller should meet the real needs of network virtualization and FlowVisor has implemented the network virtualization layer based on OpenFlow between the controller and OpenFlow switch. It makes hardware forwarding plane able to be shared by multiple network logical slices. Each section has a different logical strategy forward. In this slice mode, multiple controllers can manage a switch at the same time, and network administrator can control network in parallel. Therefore, normal network traffic can run on a separate slice mode, thus ensuring the normal flow uninterrupted. In traditional WSN routing protocol, the node is also facing resources implosion problem (one node receives multiple copies of data from neighbor nodes almost

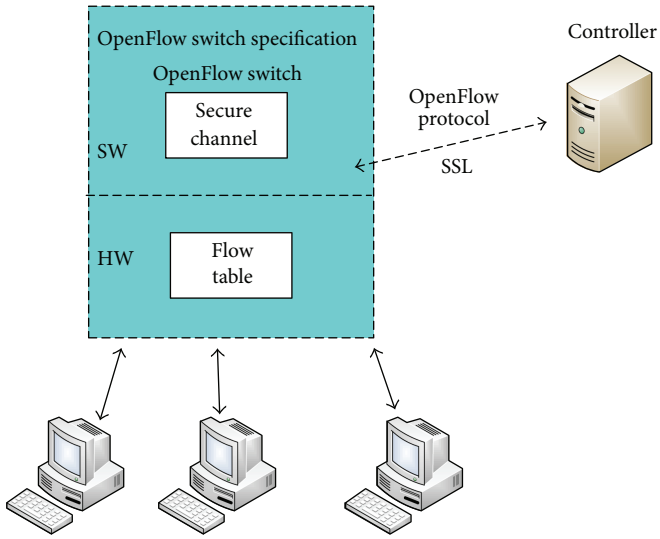


Figure 3: OpenFlow infrastructure.

at the same time) and overlapping (node has received the same data from the same area). The problems of center node can be solved based on this biopsy strategy.

2.2. Center Node/OpenFlow Switch. As shown in Figure 3 OpenFlow switch is responsible for data forwarding function. Main technical details are made up of three parts: flow table, secure channel, and OpenFlow protocol.

Each OpenFlow switch processing unit consists of flow table, and each flow table is made up of many flow table entries. Flow table entry represents the forwarding rules. In order to enhance the efficiency of the query traffic,the current flow table query through multistage flow table and pipeline mode.Flow table item is mainly composed of match fields,counters and instructions. The structure of matching field contains a lot of matchs, as the link layer,network layer and transport layer identifies. The counter is used for statistical basic data stream. Action represents the data packets which have been matched with the stream table entry and should perform the next step. Secure channel is an interface that can connect OpenFlow switches to controllers. Through this interface, we can configure and manage OpenFlow switches by the format specified of OpenFlow protocol. Besides matching flow table item, center node needs to identify various existing WSNS routing protocols, by reserved interface it is convenient for extensions in the future [17].

3. A Typical WSN Based on OpenFlow Network Structure

As shown in Figure 4 The main ideas of this network architecture are that center node can be controlled by the master node, the specific forwarding strategy entirely depends on the master node, the whole message link is completely controllable, when normal nodes have some abnormal conditions such as energy depletion and fault conditions, the master node can adjust strategy or change the forwarding path.

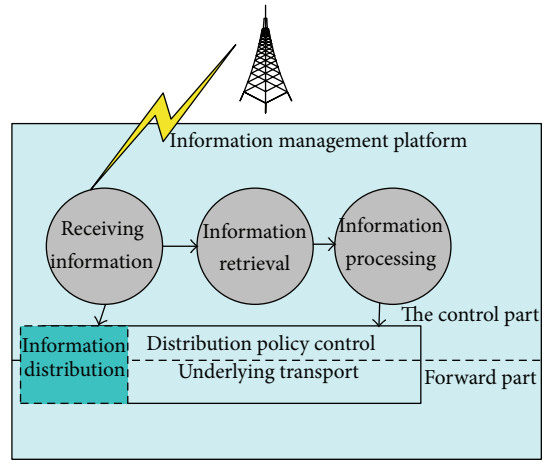


Figure 4: A typical WSN based on OpenFlow network structure.

The core of this framework is the information dissemination strategy. The master node can establish a transmission path for the information according to the QoS of the information. The master node through the secure channel among the center node and master node can generate flow table.

(1) When the center node first received the information forwarding requests, the center node matches the flow table information and will feed the node information back to the master node.

(2) After it receives the feedback information, the master node begins to construct an optimal forwarding policy based on the information of QoS and the network topology maintained by the center node.

(3) When the master node has chosen a path, the center node will be informed through the flow table to be added to it. When the information arrives at the center node, the message can be forwarded according to

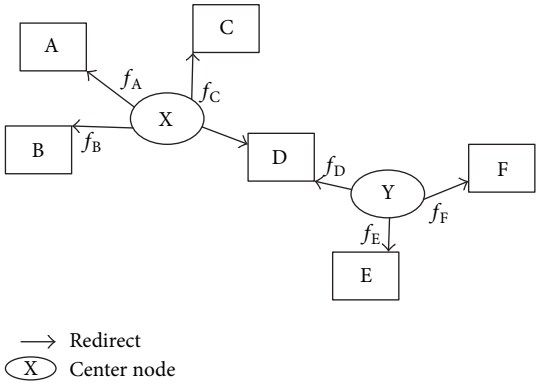


Figure 5: The relative position to determine.

the strategy that has been built. Because the forwarding strategy is the optimal path chosen by the QoS, node status, and the network topology, the message forwarding efficiency will be very high.

(4) When the information is sent over and the next flow of information arrives, the center node will check the flow table at first. If the information can be matched with the last, it means the QoS, node status, and the network topology are consistent with the previous one. There is no need to construct a new forward path. If the flow table exists, flow table entry cannot be forwarded, center node will feed the information back to the master node and request the construction of a new path.

Since every information forwarding has selected the optimal strategy and fully controllable, the stability and efficiency of the network are much higher than traditional WSN routing protocol.

4. WSN Network Center Node Deployed

In order to avoid the problem that the location of the center node unreasonable such as concentration or in the network edge, literature [18] suggests an algorithm. The center node location improvement by a distance-aware route algorithm for cloud computing is based on CAN. As shown in Figure 5 In order to measure the distance between two points, we introduce the cosine similarity to compare and calculate the similarity of two primary frequency plots, and then we can calculate the distance between two points. Frequency plot is the quantitative expression of the relative position based on the CDN network. It is a set of n-dimensional vector. It recorded the redirection information in a fixed time interval from the CDN network.

In accordance with the cosine similarity formula, we put the vector a and the vector b, respectively, into the formula; we obtained two primary frequency figures a and b as follows:

$$\text{Cos sim}(a,b) = \frac{\sum_{i\in I_a}(u_{a,i}\cdot u_{b,i})}{\sqrt{\sum_{i\in I_a} u_{a,i}^2 \cdot \sum_{i\in I_b} u_{b,i}^2}}. \quad (1)$$

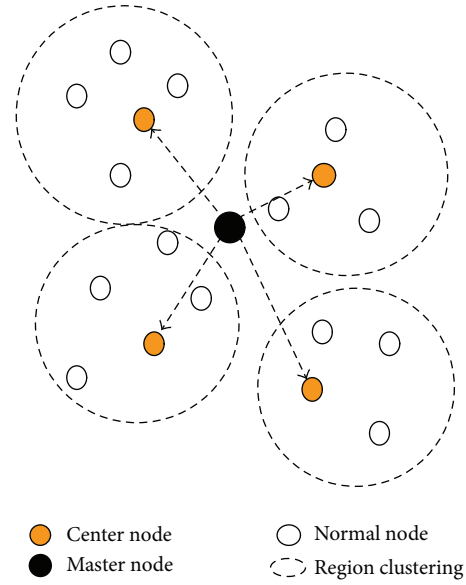


Figure 6: A master node in its control unit structure.

From the above theory we can get into the rough distance between the nodes corresponding to primary frequency plot formula. Based on the division of Land Mark, for M a Land Mark, the n-dimensional space of CAN can be divided into $M!$ Parts which follow the physical distance. We set up center node in these clustering. Area represented by the clustering structure is as follows.

(1) Center node. As shown in Figure 6 Center node is the physical center of the area representative clustering; center node can locate areas by junior or senior frequency diagram, based on the measurement system, only when the physical distance between the nodes and center node is less than certain value. This node is in center node area.

(2) Regional Unified Virtual Radius. When nodes are less than MaxPeer already in the area, we need to set a uniform distance d, with a constant value as a screening normal node reference. Regional unified virtual radius d is a variable, and it can be set according to the size of the network and the demand.

5. Experiment Results

In order to verify the effectiveness of the network structure in this paper, we use a professional experiment tool MATLAB and put the results compared with the traditional routing protocol. The sense for the experiment as nodes distributed in an area 100 m × 100 m; the node number is 100; the master node/base station is located in the center region and the energy can be supplied:

initial energy $E = 0.5$ J,

message size 4000 bit,

$\mathscr{E}$fs = 10 pJ/bit/m^2,

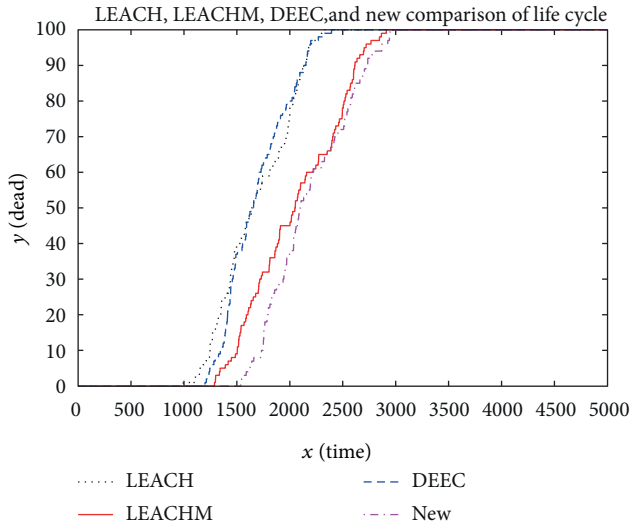


Figure 7

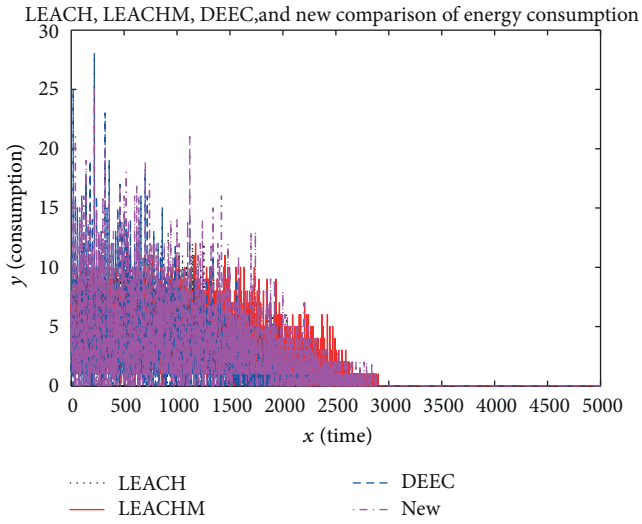


Figure 8

$\mathscr{E}mp = 0.0013\,pJ/bit/m^4$,

$EDA = 5\,nJ/bit/message$.

In order to facilitate the comparison with LEACH, LEACHM, and DEEC routing protocol, we ignore the influence of random factors such as wireless channel interference. We can get Figure 7 when the network is performed 5000 times and we compared the WSN structure based on the technology of SDN with the LEACH, LEACHM, and DEEC.

From Figure 7 we can compare the difference of the number of death nodes between the new structure and the traditional routing protocols. The death node appeared when LEAC, LEACHM, and DEEC protocol execution are 1000–1300 times. When the system runs 2500 times, all of the nodes are dead and when it uses DEEC protocol, the system runs 3000 times. The death node appeared when the system ran the new structure and all of the nodes died when it ran 3000 times. So we can see that, compared with traditional routing protocol, the new structure is more stable.

From Figure 8 we can see the changes of energy consumption, the new structure is relatively stable and remains on a lower power consumption level on the node. When the network topology changes, some nodes has high instantaneous energy consumption, but the network has a long life. The traditional LEACH protocol energy consumption is relatively stable in the early times, with the selection of cluster head and changes of the network topology; the energy consumption was exhausted when it runs 2500 times. We can see that though some nodes have high instantaneous energy consumption, the life of the network is longer than that of the traditional routing protocols.

From Figure 9 we can see the load balancing of the new structure is lower than others, the highest load balancing rate of the new structure is 1, and the rate of LEACH protocol is 1.5, another two rates between 1 and 1.5.

From Figure 10 we can see that the new structure has a great advantage in the data transmission. In the process of work, LEACH and LEACHM protocol need to select the

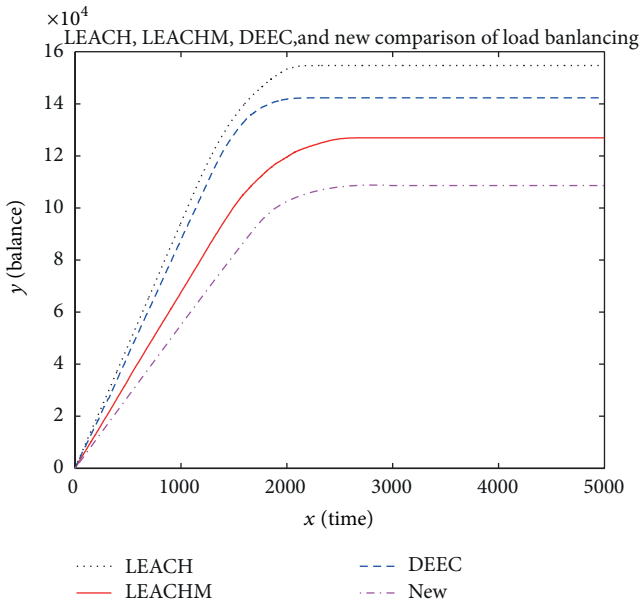


Figure 9

clusters heads several times; due to the limitations of the protocol they cannot guarantee the efficiency of the network and the nodes. In the early stages of the network lifetime, the traditional LEACH protocol and DEEC protocol can maintain a higher coverage. but with the energy depletion of the nodes, the position of cluster head has changed. Network coverage decreased rapidly, so the whole life of the network is very short. The WSNS structure is based on the SDN. Because center nodes have relatively fixed position and they can flexibly choose routing protocol according to the change of network topology, the network coverage can maintain a high level. The whole network maximum to avoid the waste of energy. From Figure 7 we can see that, compared with the traditional routing protocols, the new structure has brought very big rise.

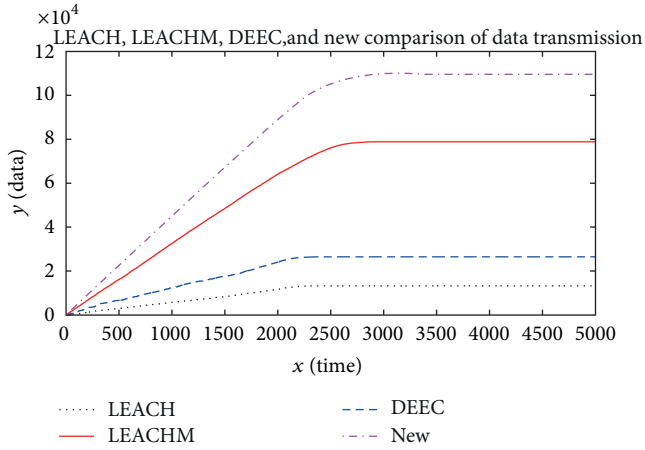


Figure 10

6. Conclusion

Because many factors should be considered when we design routing protocols such as node energy, load balancing, and network topology changes, under the original design concept, it is difficult for us to design a routing protocol that can be suitable for all situations. In this paper, a kind of WSN structure based on the technology of SDN is a good solution to these questions. However, we have some problems to solve such as the energy limit of center node and master node, normal nodes are missing, node zoning and so forth. Next step We will design algorithm to solve these problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The work supported by the National Natural Science Foundation of China (Grants no. 61103195, and no. 61373137).

References

[1] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[2] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufmann, Boston, Mass, USA, 1st edition, 2004.

[3] M. O. Pervaiz, M. Cardei, and M. Wu, "Routing security in ad hoc wireless networks," in *Network Security*, pp. 117–142, Springer, New York, NY, USA, 2010.

[4] J. P. Rohrer, A. Jabbar, E. K. Cetinkaya, E. Perrins, and J. P. G. Sterbenz, "Highly-dynamic cross-layered aeronautical network architecture," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 4, pp. 2742–2765, 2011.

[5] Y. Tang, M.-T. Zhou, and X. Zhang, "Overview of routing protocols in wireless sensor networks," *Journal of Software*, vol. 17, no. 3, pp. 410–421, 2006.

[6] J. Kulik, W. R. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, no. 2-3, pp. 169–185, 2002.

[7] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.

[8] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 22–31, Atlanta, Ga, USA, September 2002.

[9] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 243–254, Boston, Mass, USA, August 2000.

[10] D. Niculescu and B. Nath, "Trajectory based for warding and its applications," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 260–272, San Diego, Calif, USA, 2003.

[11] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, vol. 1, pp. 350–355, Orlando, Fla, USA, 2002.

[12] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pp. 3005–3014, Maui, Hawaii, USA, 2000.

[13] S. Lindsey and C. S. Raghavendra, "PEGASIS: power-efficient gathering insensor information systems," in *Proceedings of the IEEE Aerospace Conference*, pp. 1125–1130, Big Sky, Mont, USA, 2002.

[14] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[15] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufmann, Boston, Mass, USA, 1st edition, 2004.

[16] M. O. Pervaiz, M. Cardei, and M. Wu, "Routing security in ad hoc wireless networks," in *Network Security*, pp. 117–142, Springer, New York, NY, USA, 2010.

[17] S. K. Singh, M. Singh, and D. Singh, "Routing protocols in wireless sensor networks-a survey," *International Journal of Computer Science & Engineering Survey*, vol. 1, no. 2, pp. 63–83, 2010.

[18] Z.-J. Han, L.-S. Huang, R.-C. Wang, L.-J. Sun, and M. Sheng, "Distance aware route algorithm for cloud computing based on CAN," *Journal on Communications*, vol. 32, no. 7, pp. 29–39, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 927940, 12 pages
http://dx.doi.org/10.1155/2013/927940

Research Article

Discovering Human Presence Activities with Smartphones Using Nonintrusive Wi-Fi Sniffer Sensors: The Big Data Prospective

Weijun Qin,[1] Jiadi Zhang,[2] Bo Li,[2] and Limin Sun[1]

[1] *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, No. 89, Minzhuang Road, Haidian District, Beijing 100093, China*
[2] *School of Software and Microelectronics, Peking University, Beijing 102600, China*

Correspondence should be addressed to Weijun Qin; qinweijun@iie.ac.cn

Received 18 July 2013; Accepted 10 October 2013

Academic Editor: Fu Xiao

With the explosive growth and wide-spread use of smartphones with Wi-Fi enabled, people are used to accessing the internet through Wi-Fi network interfaces of smartphones. Smartphones periodically transmit Wi-Fi messages, even when not connected to a network. In this paper, we describe the *Mo-Fi* system which monitors and aggregates large numbers of continuous Wi-Fi message transmissions from nearby smartphones in the area of interest using nonintrusive Wi-Fi sniffer sensors. In this paper, we propose an optimized Wi-Fi channel detection and selection method to switch the best channels automatically to aggregate the Wi-Fi messages based on channel data transmission weights and human presence activity classification method based on the features of human dwell duration sequences in order to evaluate the user engagement index. By deploying in the real-world office environment, we found that the performance of Wi-Fi messages aggregation of CAOCA and CACFA algorithms is over 3.8 times higher than the worst channel of FCA algorithms and about 76% of the best channel of FCA algorithms, and the human presence detection rate reached 87.4%.

1. Introduction

Big data is leading a new prospective of data computation, storage, analysis, and mining in the recent years [1–3]. With the explosive growth and wide-spread use of smartphones with Wi-Fi enabled, people are used to accessing the internet, for example, watching videos on *Youtube* and chatting on *Facebook* through Wi-Fi network interfaces of smartphones in the area where Wi-Fi hotspots are deployed in order to save network traffic costs. In the meanwhile, about 40% to 70% of people always turn on Wi-Fi network interface of smartphones instead of turning it off for energy savings. Smartphones with Wi-Fi enabled periodically transit Wi-Fi probe messages, even when not connected to a Wi-Fi network [4]. When smartphones detect and connect to Wi-Fi hotspots in the area of interest, the background programs and services of operating systems in the smartphones can generate large numbers of data transmissions, for example, Apple iOS message push notification service and Android message notification service. As Wi-Fi network interface in the smartphones has the unique MAC address, it is possible to identify the smartphone's owner and distinguish his presence in the area of interest.

By deploying Wi-Fi sniffer sensors in the area of interest, it is possible to capture Wi-Fi message transmissions without disturbing the normal daily use of smartphones, and to analyze the situation of humans stay and even the coarse-grained location traces of the smartphones from the big data prospective. With the features of Wi-Fi sniffing, it can be widely used in the public places, for example, shopping mall, office buildings to analyze the dwelling of smartphones as well as the owners, even the features of human presence activities and classify varied user groups. In the shopping mall, for instance, user or customer engagement index [5] is a significant index of business decision for shop owner, for example, customer traffic, returning customers ratio, dwell duration, visit frequency, and so forth. In this paper, we design and implement the *Mo-Fi* system and investigate the feasibility of nonintrusive Wi-Fi sniffing for smartphones and human presence

activity patterns by deploying in the real-world office environment.

There are two key challenges to discover human presence activity patterns using Wi-Fi message sniffing on smartphones without disturbing the normal daily use of smartphones. The first challenge is the fact that smartphones with Wi-Fi enabled work on 14 channels, and the area of interest may deploy no Wi-Fi access point or multiple access points working on different channels. It is difficult to select the most active working channel in order to aggregate the Wi-Fi probe or data messages as much as possible in the real-world deployment. The second challenge is how to extract the features of human presence activities from the original Wi-Fi message sequence with timestamp.

The primary contributions of this paper are as follows.

(i) Non-intrusive Wi-Fi sniffing approach to aggregate Wi-Fi packages from smartphones.

(ii) Optimized channel selection algorithm to switch Wi-Fi sniffer sensor to work on those busy channels.

(iii) An activity classification algorithm to discover human presence behaviors in the area of interest.

(iv) Evaluation of human presence activity patterns in the office environment with the real-world deployment.

With the real-world deployment of *Mo-Fi* system in the office environment, we found that smartphones with Wi-Fi enabled generate numerous and continuous Wi-Fi probe message transmissions, even when not connected to a wireless network nearby or smartphones turn the screen off. During the data aggregation with one deployed Wi-Fi sniffer sensor for one month, we totally collected over 1,380,330 Wi-Fi messages from 12,496 mobile devices with Wi-Fi enabled via the IEEE OUI Registry [6], with the average of 46,011 messages per day. Among those monitored 12,496 devices, 1,437 of them are recognized as the visitors whose dwell duration is over 10 minutes. From the results of our classification methods, we dug out four types of human presence activity patterns, including the *outside*, *walkbys*, *bounced*, and *engaged*.

The remainder of the paper is structured as follows. Section 2 starts with the research backgrounds and requirements for the *Mo-Fi* system. We describe the overall operation of *Mo-Fi* system in Section 3. In Section 4, our optimized Wi-Fi sniffing channel selection algorithm is described to meet the first challenge, followed by human presence activity classification method in Section 5. Then Section 6 gives the *Mo-Fi* system deployment in our experimental office environment and performance evaluations. Finally, related works and some concluding remarks are made.

2. Motivation

From the retailers perspective, human presence activity can be virtual gold mine [7]. In the shopping mall, for instance, the shop owners are eager for the information about the customers behaviors after entering into the store, for example, which goods shelf or area where shoppers spend long time and monthly returning customer visit ratio. Tracking this information, the shop owner can predict the shoppers, purchase intention or needs and adjust their business decision, for example, more attractive discounts. Researchers have investigated several approaches to achieve the offline analytics for retailers on vision recognition approaches [7, 8].

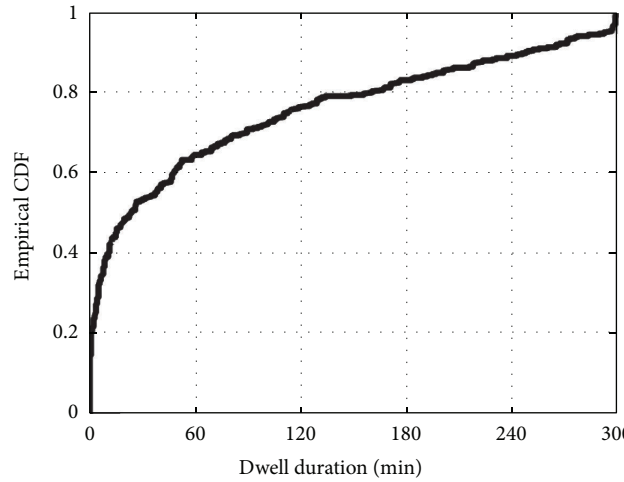


Figure 1: Clients at a university cafe exhibit varied dwell times, reflecting multiple patterns of human presence.

In [2], the experimental results distinguished that smartphones with Wi-Fi enabled transmit Wi-Fi probe messages periodically, even not to associate with a network. In [9], they set up a Wi-Fi enabled laptop as a traffic sniffer monitoring on three strongest APs in the cafe and used `tcpdump` command to monitor the devices connecting to the APs. Figure 1 shows the CDF of people dwell times at a university cafe. They found that more than 30% of the devices dwelled for less than 10 minutes (e.g., the user had a coffee/food to go); and more than 20% of the devices stayed at least two hours (e.g., web browsing).

From the real-world use scenario (e.g., shopping mall, cafe, and outdoor show), we can elicit the following requirements that support discovering human presence activity with smartphones using non-intrusive Wi-Fi sniffer sensors.

(i) *R*1: minimum assumption on smartphones: One of the first ideal features is that smartphones do not need to install any specialized Apps software in order to aggregate Wi-Fi packages.

(ii) *R*2: non-intrusive to the existing wireless networking: another ideal feature is that Wi-Fi message sniffing is non-intrusive to the existing wireless networking in the area of interest.

(iii) *R*3: redundant message elimination and compress: as smartphones with Wi-Fi can provide numerous probe message transmissions, it is necessary to eliminate numerous messages with the same timestamp, and uplink distinguished Wi-Fi messages with varied timestamps.

(iv) *R*4: capture the behavior of users stay: as smartphones have unique MAC address of Wi-Fi network

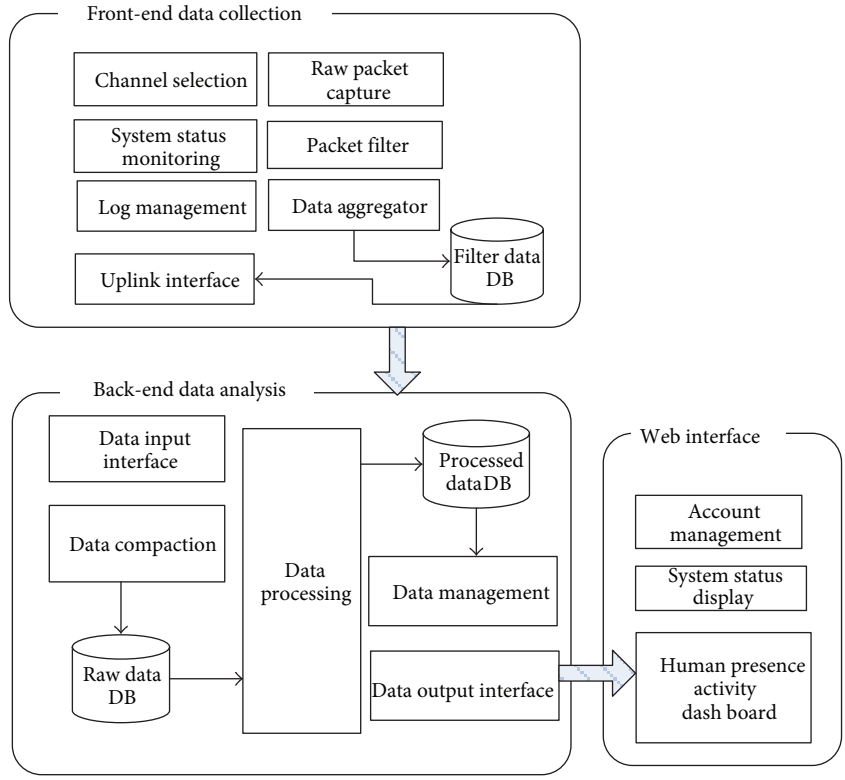


Figure 2: System architecture of *Mo-Fi* system with front-end Wi-Fi sniffer sensor and back-end data analysis services.

interface, the behaviors of the owners stay could be distinguished and extracted from the Wi-Fi messages sequence from Wi-Fi sniffer sensors.

(v) *R*5: human presence activity classification: to evaluate the user engagement index, varied human presence activity patterns should be classified from the features of the users dwelling behaviors, for example, walkbys and engaged customers.

(vi) *R*6: privacy protection against unsecure exposed Wi-Fi sniffing: when the user enables Wi-Fi network interfaces of his smartphones, user privacy should be guaranteed so that he might choose not to reveal his identity during passive Wi-Fi sniffing.

To address the abovementioned requirements, we propose and implement *Mo-Fi* system which can monitor and aggregate large numbers of ubiquitous Wi-Fi messages from smartphones using non-intrusive Wi-Fi sniffer sensors and discover human presence activities, as the unique MAC address could be used as an identity. The Wi-Fi sniffer sensors can detect the best active working channels and capture the device's MAC address with one-way hashing encryption and uplink the encrypted MAC IDs with extracted timestamp to the back-end big data analysis. The back-end functions provide large-scale storage and process the data, extract the stay of smartphones and classify the human presence activities, and display the statistics results on dashboard in the web interface.

3. System Overview

This section presents the system overview of *Mo-Fi* system before we discuss the Wi-Fi channel detection and selection and human presence activity classification issues.

3.1. System Architecture. As shown in Figure 2, the architecture of *Mo-Fi* system is divided into front-end Wi-Fi sniffer devices and back-end data analysis services and web interfaces.

The front-end Wi-Fi sniffer sensor is an ARM-based embedded device, which detects Wi-Fi channels, aggregates Wi-Fi messages from nearby smartphones, and uplinks to back-end clouds. *Channel Detection and Selection* module detects and selects better active sniffing channel due to the user's configuration. *Data Filtering and Compression* module filters out redundant, duplicated, or non-mobile-device data packets. *Data Uploading* module uploads the sniffed messages to back-end cloud RESTful API. *Log Management* module keeps the runtime status of all modules and stores them in local database. *System Status Monitoring* module monitors the parameters of CPU, memory, and disk usage in the Wi-Fi sniffer sensor for device diagnosis.

The back-end data analysis service is running on cloud servers, provides RESTful API to push/pop filtered and compressed Wi-Fi messages, and stores data in the database. *Data Input* module receives data from front-end devices and stores them into NoSQL-based database. *Data Processing* module loads raw data and analyzes data to human presence activities

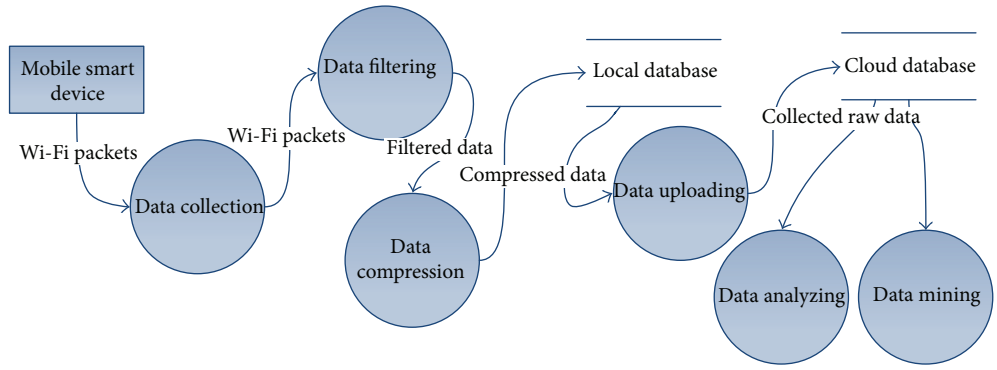


Figure 3: Workflow of *Mo-Fi* system modules from Wi-Fi message sniffing to human presence activity recognition and classification.

using *Analyzing and Mining* module and then stores the analyzed statistics results. *Data Output* module provides Open API for data fetching of third-party modules.

The web interface demonstrates human presence activities and device status with charts, bars, and tables, which could display analysis data in real time.

3.2. Workflow. As the workflow of *Mo-Fi* system shown in Figure 3, the data processing procedure contains three phases of data fetching phase, data storage phase, and data analyzing phase. Data fetching method is designed to aggregate as much data messages from smart devices, which is embedded on Wi-Fi channel detection and selection algorithms; data storage phase is designed to provide better persistence for big data Wi-Fi data analysis, which includes data filtering and data compression for eliminating redundant messages; data analyzing phase is designed to extract the useful information from Wi-Fi messages (e.g., RSSI, SRC MAC address, DEST MAC address, PROBE/DATA message type, and the timestamp), analyze the dwell duration of the human behaviors, and finally dig out and classify the human presence activities.

4. Optimized Channel Detection and Selection

In this section, we discuss the Wi-Fi channel detection and selection methods to enhance the efficiency of Wi-Fi message sniffing and aggregation.

4.1. Channel Detection and Selection Algorithms. The purpose of channel detection and selection method is to monitor and aggregate Wi-Fi messages from smartphones including probe message and data message as much as possible. Smartphones transmit periodically Wi-Fi probe messages to all 14 channels and send Wi-Fi data messages in the fixed working channel of connected Wi-Fi network when associated with a Wi-Fi AP. Therefore, it is key and significant to detect and find the working channel to monitor in the area of interest. Traditional sniffing method uses *Fixed Channel Allocation* (FCA) method to pick up one fixed channel randomly. The front-end Wi-Fi sniffer sensors of *Mo-Fi* system are designed not only to scan the working channels but also to switch to the one in order to capture message packets more efficiently. Besides the FCA methods, other three channel detection and selection methods are provided in the *Mo-Fi* system as described below.

(i) Average Time Channel Allocation (ATCA). The ATCA method allocates the unified length of sniffing time slices for all 14 Wi-Fi channels. The *Mo-Fi* system switches the sniffing channel in roll-polling way.

(ii) Channel Activeness Based Channel Fair Allocation (CACFA). The CACFA method as Algorithm 1 described allocates the varied length of sniffing time slices for all 14 Wi-Fi channels based on the received message number and weights of each channel. The method sorts each channel message counting list in ascending order in the first 5 minutes. Then it separates the remainder 55 minutes into numbers of time slices by 15 seconds interval (default) and dispatches the slices to each channel in the counting list due to the packets percentage of total packets number. Each channel on list would at least get one time slice.

(iii) Channel Activeness Based Optimized Channel Allocation (CAOCA). The CAOCA method is derived from the CACFA method which discards those channels with extremely low level activeness and tries to switch the channel with high quality activeness and allocates more time slices as much as possible. Instead of sorting the list in ascending order, the CAOCA algorithm shown in Algorithm 2 sorts the channel's weights in descending order in the first 5 minutes in the detection phase. Then it separates the reminder 55 minutes into time slices by 15 seconds interval (default) and allocates the time slices to those high-quality channels first in the list according to the packets percentage of total packets number. When over 95% time slices have been allocated, the algorithms stop. Eventually, one or a few channels with high-quality from total 14 channels have been selected.

Table 1 gives an example of the message counting results of each Wi-Fi channel. Normally, CH.6 is the default configured working channel on the most Wi-Fi APs, which has the best high-quality activeness. The detection and selection procedure is divided into a 5-minute detection phase and 55-minute selection and monitoring phase. Totally, the 720 5-second time slices are divided and allocated to those channels. The packet counting result shows that the CAOCA algorithm

```
Input: channel_list: an array list of detection result,
       including channel number and packet count
       total_data_count: the sum of packet count
       during detection total_time_slice: the total
       number of time slices divided by threshold
Output: channel_list.time_slice
(1)  sort channel_list in ascending order by packet count
(2)  for i = 1; i ≤ channel_list.length; i + + do
(3)      percentage = channel_list[i]/total_data_count;
(4)      time_slice = percentage ∗ sum(time_slice);
(5)      if time_slice ≤ 1 then
(6)          time_slice = 1;
(7)      end
(8)      channel.time_slice = time_slice;
(9)  end
(10) return channel_list.time_slice;
```

Algorithm 1: The CACFA algorithm.

```
Input: channel_list: an array list of detection result,
       including channel number and packet count
       total_data_count: the sum of packet count
       during detection total_time_slice: the total
       number of time slices divided by threshold
Output: channel_list.time_slice
(1)  sort channel_list in descending order by packet count
(2)  total_percentage = 0
(3)  for i = 1; i ≤ channel_list.length; i + + do
(4)      percentage = channel_list[i]/total_data_count;
(5)      time_slice = percentage ∗ sum(time_slice);
(6)      channel.time_slice = time_slice;
(7)      total_percentage+ = percentage;
(8)      rest_time_slices− = time_slice
(9)      if total_percentage ≥ 0.95 then
(10)         break;
(11)     end
(12)     channel.time_slice = time_slice;
(13) end
(14) channel_list[0].time_slice+ = rest_time_slices
(15) return channel_list.time_slice;
```

Algorithm 2: The CAOCA algorithm.

is a little better than the CACFA algorithm theoretically, and these two algorithms are much better than the ATCA algorithm. The FCA algorithm could have the best performance if the best high-quality working channel is chosen occasionally, while the inactive channel could get worse performance. The expectation value of FCA more or less equals the ATCA.

4.2. Redundant Message Elimination and Compression. Smartphones with Wi-Fi enabled can produce large amounts of Wi-Fi message transmissions in seconds [2]. According to our experimental records, one smartphone can send 3 to 5 probe requests in average and receive the same amount. In Table 1, an example of time slice allocation of FCA, ATCA, CACFA, and CAOCA algorithms of probe response within 1 hour is shown. The total amounts of uncompressed and unfiltered packets could be up to more than 10 millions per day, which is a big burden to data analysis or data mining, in which we found that the messages contain tons of duplicated data. The front-end sensors of *Mo-Fi* system enable redundant message elimination and compression for sniffed packets. The algorithm shown as Algorithm 3 has two steps: elimination and compression. It maintains a MAC library of all the brands of smartphones and a dictionary of existing devices Wi-Fi modules MAC address with packet type within a certain time threshold (the default threshold is 1 second, which means if a Wi-Fi message is recorded, the algorithm would discard those other same messages within the threshold window). When a Wi-Fi message is sniffed, the algorithm

Table 1: Example of time slice allocations of FCA, ATCA, CACFA, and CAOCA algorithms.

CH.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Total
Packets in 5 s	500	10	10	200	10	3000	300	10	10	10	10	10	10	10	4100
Percentage (%)	12.2	0.2	0.2	4.9	0.2	73.2	7.3	0.2	0.2	0.2	0.2	0.2	0.2	0.2	100
Algorithms	Dispatched time slices														Count
FCA (best CH.)	0	0	0	0	0	720	0	0	0	0	0	0	0	0	2160000
FCA (worst CH.)	0	720	0	0	0	0	0	0	0	0	0	0	0	0	7200
ATCA	52	52	52	52	52	52	51	51	51	51	51	51	51	51	212830
CACFA	80	1	1	32	1	491	48	1	1	1	1	1	1	1	1551635
CAOCA	80	0	0	32	0	500	48	0	0	0	0	0	0	0	1578535

Input: *mac_dict*: a hash table records MAC of smartphones: *data_dict*: a hash table records existed data packet information with time threshold; *last_timestamp*: the time stamp of the last recorded
Output: *stored_data*
(1) Get data from sniffer thread
(2) **if** !*mac_dict.has_key*(*data.mac_md*5) **then**
(3) discard data
(4) **end**
(5) **if** *data.timestamp* ≠ *last_timestamp* **then**
(6) delete expired data from *data_dict*;
last_timestamp = *data.timestamp*
(7) **end**
(8) **if** *data_dict.has_key*(*data.mac_md*5) **then**
(9) **if** *data_dict*[*data.mac_md*5].*type* == *data.type* **then**
(10) discard data
(11) **end**
(12) **end**
(13) add data to *data_dict*;
(14) return *stored_data*;

Algorithm 3: The data filtering and compression algorithms.

would filter out packets of nonmobile smart devices based on the IEEE OUI Registry.

5. Human Presence Activity Classification

In this section, we discuss the human presence activity recognition and classification method from the collection of the aggregated Wi-Fi messages to the behaviors of human presence activities.

5.1. Feature Extraction on Wi-Fi Message Sequence. While Wi-Fi sniffer sensors detect and capture message transmissions with all types of Wi-Fi packages, *Mo-Fi* system filters out three types of all, including Wi-Fi `probe request`, `probe response`, and `data messages`. In the experiment, we found smartphones transmit Wi-Fi probe request or response messages periodically and send or receive data messages when associated with Wi-Fi APs. By processing the timestamp of Wi-Fi message sequence from the unique targets identified MAC address, the system calculates the unique MAC IDs dwell duration, as well as the start time and the end time. We design a TS2VD algorithm to convert the discrete Wi-Fi message sequence to unique devices or users sequential dwell duration. As the probe request period is fixed, we design a time difference value (TDV), greater than probe request period, to distinguish between two unique visits. Any time interval of two message packets with the unique MAC address which is smaller than TDV belongs to the same dwell duration, as Algorithm 4 has shown in detail.

The *Mo-Fi* system regards the unique devices or users dwell duration as the feature of the behaviors of visitor's stay in the monitored area of interests.

5.2. Human Presence Activity Classification on Dwell Duration. Varied time lengths of dwell duration indicate varied human presence activity patterns. In order to classify the patterns, the *Mo-Fi* system uses K-means clustering methods [10] to classify the human groups based on the time threshold features of dwell duration (DD), for example, capture time value (CTV), inside time value (ITV), and engaged time value (ETV). As a result, the users are divided into four groups, for example, *outside*, *walkbys*, *bounced*, and *engaged* patterns,

```
Input: WPList: Packet List of a WiFi with
       timestamps ordered by timestamp, TDV: time
       difference value
Output: VDList: dwell duration list
(1)  pCount = countofWPList
(2)  vd: Initialize visit duration data, start time, end time
     and duration
(3)  vd.startTime = WPList [0]. timestamp;
(4)  vd.endTime = WPList [0].timestamp;
(5)  vd.duration = vd.endTime − vd.startTime;
(6)  if pCount > 1 then
(7)     for i = 1; i ≤ pCount; i + + do
(8)        interval = WPList [i].timestamp;
           WPList[i − 1]. Timestamp;
(9)        if interval ≤ TDV then
(10)          vd.duration = vd.duration + interval;
(11)          vd.endTime ≤ −WPList[i].timestamp;
(12)       end
(13)       else
(14)          push vd to VDList;
(15)          reset vd object and do initialization;
(16)          vd.startTime = WPList[i].timestamp;
(17)          vd.endTime = WPList[i].timestamp;
(18)          vd.duration =
              vd.endTime − vd.startTime;
(19)       end
(20)    end
(21)    else
(22)       push vd to VDList;
(23)    end
(24) end
(25) return VDList;
```

Algorithm 4: Converting Wi-Fi message sequence to dwell duration.

as Figure 4 has shown. In the case of the experiment in the laboratory, the average total traffic is 1,676 per day, of which the capture rate of outside pattern is 68%, walkbys pattern is 8.4%, bounced pattern is 17.4%, and engaged pattern is 6.2%.

6. Deployment and Performance Evaluation

Our experiment evaluation focuses on the following.

(i) The portability of front-end packet sniffing program, including channel detection and selection algorithm and data filtering and compression algorithm.

(ii) The performance of channel detection and selection algorithm. The experiment would demonstrate how channel detection and selection algorithm affects Wi-Fi package sniffing.

(iii) The performance of redundant message elimination and compression. The experiment would demonstrate the comparison between using and not using filtering and compression algorithm.

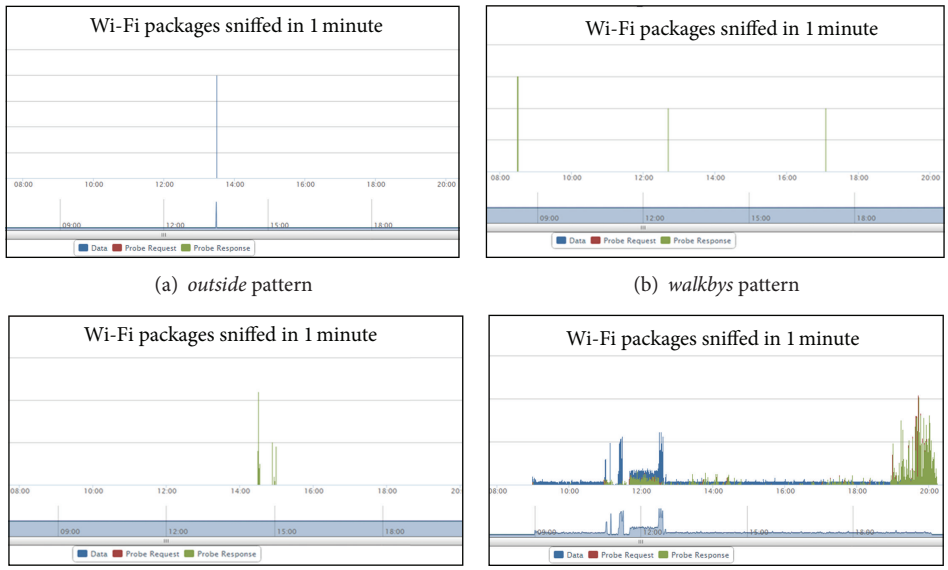
(iv) The comparison between analyzed human presence and the real circumstance in the real-world deployment in the office environment.

6.1. Prototype Implementation and Real-World Deployment

(1) Prototype Implementation of Mo-Fi System. On one side, we design and implement the front-end functions of *Mo-Fi* system in Raspberry PI [11] ARM-based embedded device with a *Ralink* Wi-Fi USB card as shown in Figure 5(a). The modules of channel detection and selection and data filtering and compression mentioned above are implemented in Python which are running on *Raspbian* operating system in Raspberry PI device. While the sniffing procedure starts, the Wi-Fi messages are sniffed and aggregated into local database as Figure 5(b) shows.

On the other side, we design and implement the functions of human presence activity recognition and classification in Python in back-end server. Also the *Mo-Fi* system provides a web portal where the human presence activities are drawn in a figure dashboard. In addition, the users can monitor device status in real time and manage the Wi-Fi sniffer sensors as shown in Figure 6.

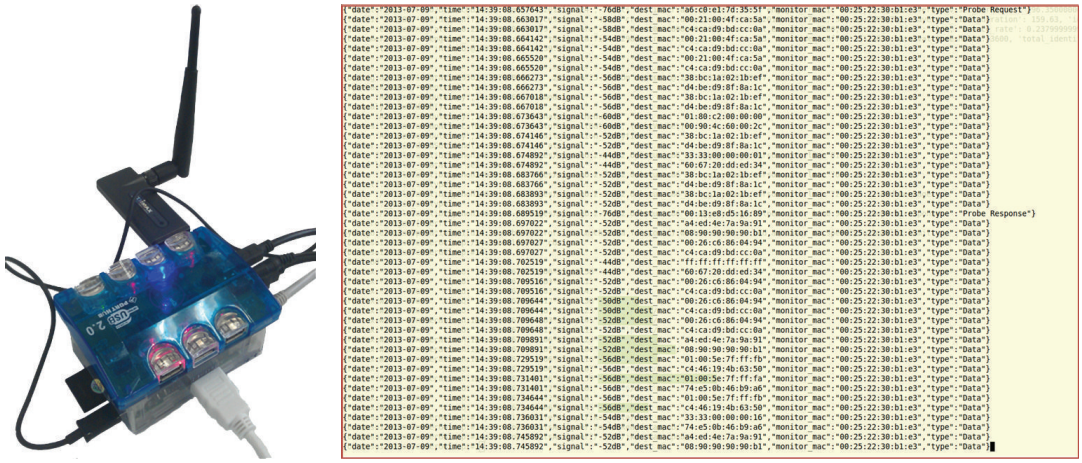
(2) Deployment in the Real-World Office Environment. As Figure 7 has shown, the *Mo-Fi* system is deployed in the real-world office environment in the Institute of Information Engineering, Chinese Academy of Sciences. The Wi-Fi sniffer



(a) *outside* pattern (b) *walkbys* pattern

(c) *bounced* pattern (d) *engaged* pattern

FIGURE 4: Four types of human presence activity patterns in the real-world testing office environment.



(a) Prototype of Wi-Fi sniffer (b) Result display of Wi-Fi message sniffing

FIGURE 5: The prototype of Wi-Fi sniffer sensor and result display screen of Wi-Fi message sniffing.

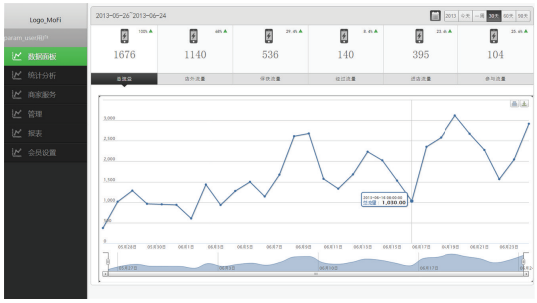


FIGURE 6: Web interfaces of *Mo-Fi* system.

sensors capture Wi-Fi probe and data packages within about 100 meters range. According to the structure of the institutes, five sniffer sensors are deployed in each corner and entrance of the building.

6.2. Performance Results of Channel Detection and Selection Algorithms. To evaluate the performance of the Wi-Fi channel detection and selection algorithms, we use 5 Wi-Fi sniffer sensors to run the different algorithms at the same place and the same time in the testing office environment as follows.

(i) *Fixed Channel Allocation* algorithms on the best high-quality active channel, for example, CH. 6.

(ii) *Fixed Channel Allocation* algorithms on the worst high-quality active channel.

(iii) *Average Time Channel Allocation* algorithms.

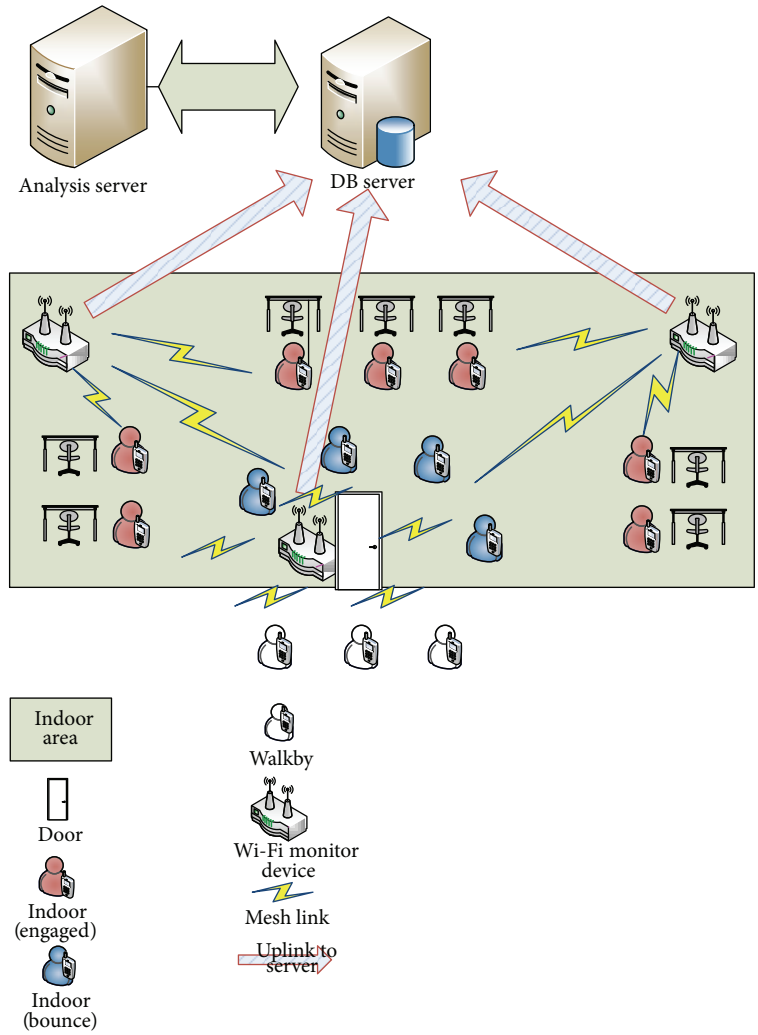


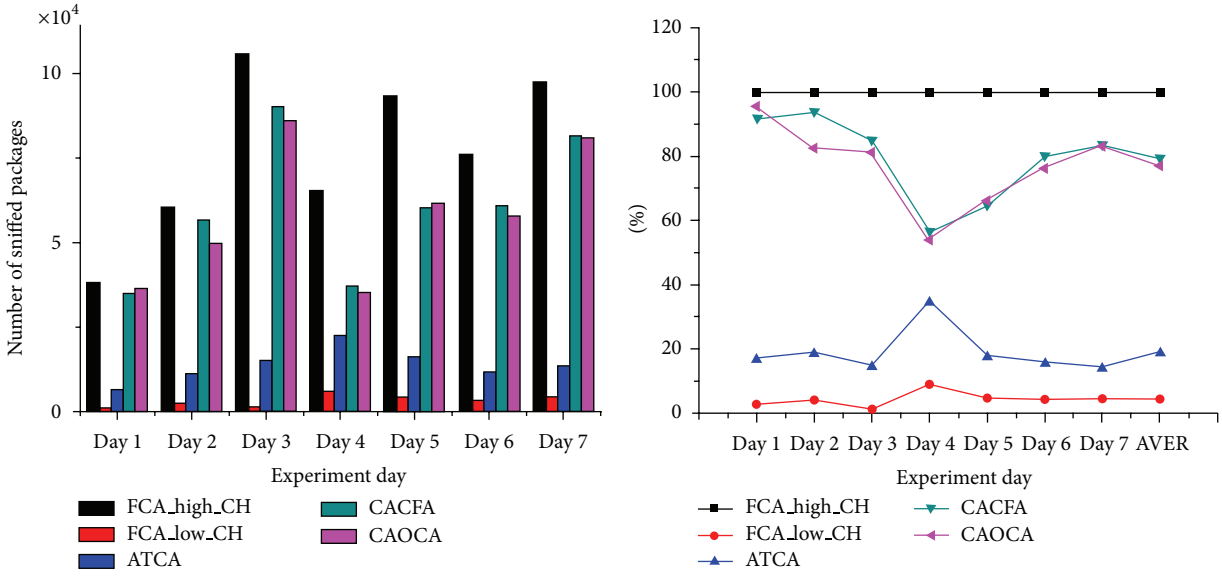
FIGURE 7: The deployment of *Mo-Fi* system.

(iv) *Channel Activeness based Channel Fair Allocation* algorithms.

(v) *Channel Activeness based Optimized Channel Allocation* algorithms.

As shown in Figure 8, the different lines demonstrate number of sniffed packets using different channel detection and selection algorithms. The sniffer sensor with *Fixed Channel Allocation* algorithms on the best high-quality active channel, for example, CH. 6, received the most numbers of Wi-Fi packages. The result shows that about 80% of routers were working on CH. 6, the channel with the best high-quality activeness. On the other hand, the sniffer sensor with *Fixed Channel Allocation* algorithms on the worst high-quality active channel works worst, only 4.2% of the best one. It proved that the sniffer sensor should detect and select the best high-quality to improve the performance of Wi-Fi sniffing. Actually, as we do not know the deployment of Wi-Fi APs in the area of interest and the most active channel, the sniffer sensor just only has 1/14 chances to hit the best channel. For the *Average Time Channel Allocation* algorithms, the performance improved to reach 18% of the best one. For the *Channel Activeness Based Channel Fair Allocation* and *Channel Activeness Based Optimized Channel Allocation* algorithms, the performance improved highly further to reach 78.6% and 76.1%. As shown in the experiment, there is no distinct difference between *Channel Activeness based Channel Fair Allocation* and *Channel Activeness based Optimized Channel Allocation* algorithms opposite, not as we expected. We supposed two possibilities but not confirmed yet; one is the performance discrepancy between the sniffer sensors hardware on channel detection, and the other is the improper configuration value of channel hopping interval (5 seconds by default).

To summarize, the channel detection and selection algorithms can facilitate the Wi-Fi sniffer sensors to aggregate Wi-Fi messages in an easy way with a different performance. By deployment on CACFA or CAOCA methods, the performance is highly improved to the normal FCA method.



(a) Numbers of sniffed packages with FCA, ATCA, CACFA, and CAOCA algorithms in 7 days

(b) Percentage of package numbers with FCA (low CH), ATCA, CACFA, and CAOCA algorithms compared with FCA (high CH)

FIGURE 8: Performance results and comparison of 4 channel detection and selection algorithms, for example, FCA, ATCA, CACFA, and CAOCA in the real-world deployed office environment.

6.3. Performance Results of Redundant Message Elimination and Compression. To evaluate the performance of redundant message elimination and compression, we run the program on three elimination and compression policies on one Wi-Fi sniffer sensor as follows.

(i) Algorithm that filters and compresses data.

(ii) Algorithm that filters but *does not* compress data.

(iii) Algorithm that *does not* filter *or* compress data.

As shown in Figure 9, the difference between bars indicates the performance of data filtering and compression algorithms with the abovementioned policies. The filtering algorithm can filter out about 90% of nonsmartphone Wi-Fi packets, and the compression algorithm can filter out about 70% of redundant filtered data. The big contrast between raw data and filtered and compressed data clearly proved the contribution on data filtering and compression.

To summarize the experiment, the filtering rate is approximately 9.29% and the compression rate is approximately 27.57%. The total contribution of data filtering and compression algorithms is to save about 97.44% storage spaces and provide better performance on further data analysis and mining.

6.4. Performance Results of Human Presence Activity Classification. To analyze human presence activity pattern, we deploy and aggregate the Wi-Fi messages in the office environment for 14 days. In the institute, there are 59 people working in the office monitored area and about 670 people nearby the institute always pass by or inside the monitoring area of Wi-Fi sniffer sensors every day. According to the office environment, we set the time difference values and thresholds as follows.

(i) Visit duration time difference: TDV = 1 hour.

(ii) Capture time value: CTV = 2 min.

(iii) Inside time value: ITV = 5 min.

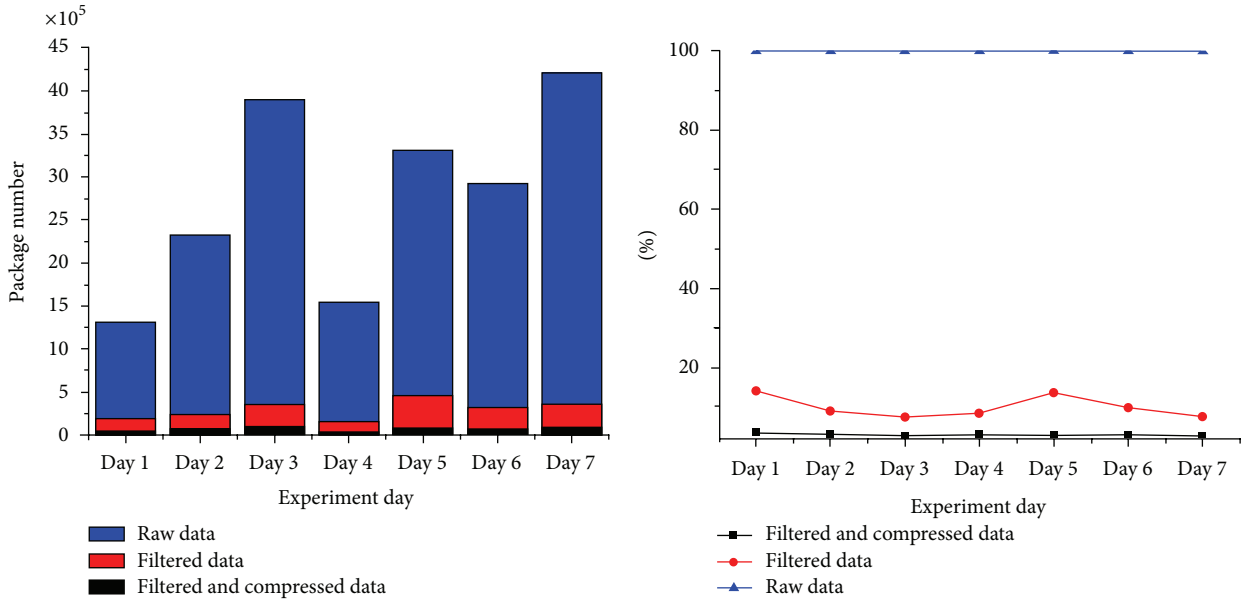
(iv) Engaged time value: ETV = 30 min.

As shown in Figure 10, we calculated the users' traffic in the office environment for 14 days. The average engaged traffic is 55, which is close to the working employees' number, and the capture traffic value 760 is close to to the institute's total employee's number. Considering those people without taking smartphones or taking more than 2 devices (e.g., iPhone and iPad), we conclude that engaged traffic is approaching to long-time presence people number and capture traffic is approaching to the sum of *walkbys* and *inside* people. By calculating, the engagement rate as a significant index of customer engagement in the working days is stable to 5.1%, which is in accordance with the features of office environment. And we found that the engaged traffics of the weekends (e.g., the points day 4 and day 11 in Figure 10) dropped to less than half of the engaged traffics in the working days, which is in accordance with people's weekly work habits.

The experimental results are in line with the real-world environment and as a result the patterns of human presence activity proved that the approach in this paper is feasible and acceptable.

7. Related Works

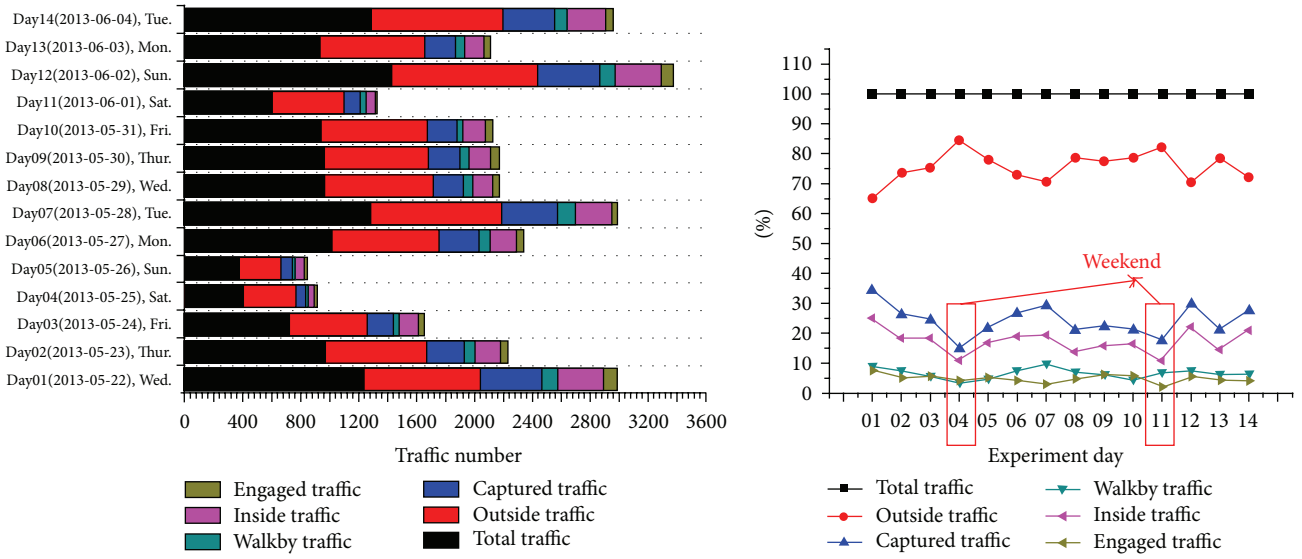
Activity recognition has become an active research area in recent years due to the pervasiveness of sensor assisted



(a) Package numbers of row data, filtered and compressed data

(b) Percentage of filtered data, filtered, and compressed data compared with row data

Figure 9: Performance results of redundant message elimination and compression.



(a) Statistical result of traffics in 14 days

(b) Percentage of outside, captured, walkby, inside, and engaged traffics compared with total traffic

Figure 10: Results of average user traffic for one month.

phones [12–16]. Researchers study on human activity recognition on varied approaches. In [17, 18], a system for sensing complex social systems with data collected from one hundred mobile phones for six months was designed to use standard Bluetooth-enabled mobile telephones to measure information access and use in different contexts, recognize social patterns in daily user activity, infer relationships, identify socially significant locations, and model organizational rhythms. In [19], a novel method was proposed to assess daily living patterns using a smartphone equipped with microphones and inertial sensors. And a feature-space combination approach for fusion of information from sensors sampled at different rates was proposed to identify various high level activities. In [20], a general technique was proposed to exploit this "multidimensional" contextual variable for human mobility prediction and extract different mobility patterns with multiple models under a probabilistic framework. To summarize, the above approaches focus on the complex activity recognition

techniques by rich sensor assisted functions on the smartphones, which assume that large numbers of smartphones have installed the elaborated Apps or software to aggregate multilevel contextual information. Actually, the assumption is a little bit ideal to the daily, practical human use.

In [2], the experiment results on Wi-Fi message transmission of smartphones give a new way to recognize human activities. By deploying Wi-Fi monitoring equipment in an area of interest, it is possible to detect these transmissions, providing a coarse-grained location trace for each phone that passes through the area without modifying the phones. Every Wi-Fi transmission contains a unique device identifier (MAC address). In [9], a system for predicting length of stay at Wi-Fi hotspots was proposed to predict dwell time with and without the aid of client sensor data using machine learning algorithm at hotspot APs. Based on the above-mentioned contributions, we investigate the Wi-Fi channel detection and selection issue to improve the efficiency of Wi-Fi package sniffing and human presence activity classification issue to dig out presence patterns.

8. Conclusion

This paper proposes *Mo-Fi*, a non-intrusive solution for discovering human presence activity, by aggregating large amounts of Wi-Fi messages from smartphones using Wi-Fi sniffer sensors without installing any Apps in smartphones and without disturbing the normal daily use of smartphones. We proposed an optimized Wi-Fi channel detection and selection approach to sniffing Wi-Fi packages as much as possible and activity classification based K-means on dwell duration sequences to identify the patterns of human presence. By deploying the *Mo-Fi* in the real-world office environment for one month, we found that the performance of Wi-Fi messages aggregation of CAOCA and CACFA algorithms is over 3.8 times higher than the worst channel of FCA algorithms and about 76% of the best channel of FCA algorithms, and the human presence detection rate reached 87.4%. We dug out four types of human presence patterns in the office environment, for example, *outside*, *walkbys*, *bounced*, and *engaged* pattern. Finally, we declaimed that our non-intrusive approach is acceptable to discover the human presence in the condition of restricted and practical deployment requirements.

Acknowledgments

The work has been funded by the State Key Development Program for Basic Research of China (2011CB302902) and Program of National Natural Science Foundation of China (61202066).

References

[1] C. Lynch, "Big data: how do your data grow?" *Nature*, vol. 455, no. 7209, pp. 28–29, 2008.

[2] J. K. Laurila, D. Gatica-Perez, I. Aad et al., "The mobile data challenge: big data for mobile computing research," in *Proceedings of the Workshop on the Nokia Mobile Data Challenge, in Conjunction with the 10th International Conference on Pervasive Computing*, pp. 1–8, 2012.

[3] P. Rose, "Top 11 technologies of the decade," *IEEE Spectrum*, vol. 48, no. 1, pp. 27–63, 2011.

[4] A. B. M. Musa and J. Eriksson, "Tracking unmodified smartphones using Wi-Fi monitors," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys '12)*, pp. 281–294, 2012.

[5] Customer Engagement, http://en.wikipedia.org/wiki/Customer_engagement.

[6] IEEE OUI Registry, http://standards.ieee.org/develop/regauth/oui/oui.txt.

[7] J. Krockel and F. Bodendorf, "Visual customer behavior analysis at the point of sale," *International Journal on Advances in Systems and Measurements*, vol. 5, no. 3-4, pp. 178–187, 2012.

[8] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[9] J. Manweiler, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Predicting length of stay at WiFi hotspots," in *Proceedings of the 32nd IEEE International Conference on Computer Communications (Infocom '13)*, 2013.

[10] W. Zhao, H. Ma, and Q. He, "Parallel K-means clustering based on MapReduce," *Cloud Computing*, vol. 5931, pp. 674–679, 2009.

[11] Raspberry PI, http://www.raspberrypi.org/.

[12] J. K. Laurila, D. Gatica-Perez, and I. Ad, "The mobile data challenge: big data for mobile computing research," in *Proceedings of Mobile Data Challenge by Nokia Workshop*, Newcastle, UK, 2012.

[13] J. Gómez-Romero, M. A. Serrano, M. A. Patricio, J. García, and J. M. Molina, "Context-based scene recognition from visual data in smart homes: an information fusion approach," *Personal and Ubiquitous Computing*, vol. 16, no. 7, pp. 835–857, 2012.

[14] E. Miluzzo, N. Lane, K. Fodor et al., "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pp. 337–350, 2008.

[15] P. Barralon, N. Noury, and N. Vuillerme, "Classification of daily physical activities from a single kinematic sensor," in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS '05)*, pp. 2447–2450, September 2005.

[16] J. Chon and H. Cha, "LifeMap: a smartphone-based context provider for location-based services," *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 58–67, 2011.

[17] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.

[18] N. Eagle, A. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 36, pp. 15274–15278, 2009.

[19] R. K. Ganti, S. Srinivasan, and A. Gacic, "Multisensor fusion in smartphones for lifestyle monitoring," in *Proceedings of the International Conference on Body Sensor Networks (BSN '10)*, pp. 36–43, June 2010.

[20] T. Do and D. Gatica-Perez, "Contextual conditional models for smartphone based human mobility prediction," in *Proceedings of ACM International Conference on Ubiquitous Computing*, Pittsburgh, Pa, USA, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 839586, 6 pages
http://dx.doi.org/10.1155/2013/839586

Research Article

Research and Design of an Airfield Runway FOD Detection System Based on WSN

Li Ang

Nanjing University of Science and Technology, Zijin College, Nanjing, Jiangsu 210046, China

Correspondence should be addressed to Li Ang; lyonbox@126.com

Received 18 July 2013; Accepted 23 October 2013

Academic Editor: Zhijie Han

Foreign object debris (FOD) would potentially cause huge damage when it appears on the airport runway, so the FOD surveillance system is one of the essential protectors for airplane's safety now. This paper introduces a designed method of airport runway FOD detection system based on WSN, analyzes function of various image acquisition sensors, and introduces FOD image analysis algorithm. Finally WSN data fusion technology is used to analyze the FOD image.

1. Foreword

What is FOD? FOD is the abbreviation of foreign object debris, which refers to a foreign substance that may damage the aircraft system. There are many types of FOD, such as connecting pieces between aircraft and engine, flying objects, wild animal, leaves, sand, and stone. FOD is very harmful to the aircraft. For example, Air France Concorde crash in July 25, 2000 caused by FOD killed 113 people. The crash created by FOD took research of FOD automatic, fast, and accurate monitoring and alarm system on the agenda. This further proved that an effective safety airport runway intelligent detection system is an important part of the airport flight safety system. Nowadays, there are four typical systems in the world, the British Tarsier system (Figure 1(a)), FOD Detect detection system of Israel (Figure 1(b)), FOD Finder monitoring system (Figure 1(c)), and iFerret monitoring system (Figure 1(d)) [1–4].

At present, most of the domestic airports are still using manual method which is neither efficient nor safe. China has not yet found the radar system on FOD monitoring. Current research methods for detecting runway FOD are mainly for image processing on the runway image to realize the detection and localization of FOD.

This paper introduces the design of a runway detection system based on WSN. The main purpose of this design is to conduct real-time detection, recognition, and tracking of track target, for the general sensors are unable to achieve this goal. In this design, the radar sensor, visible light cameras, infrared cameras, and laser ranging technology are used to further analyze the grade of target threat and the position of the FOD. In addition, the system uses multiple sensors to detect, so we use the sensor data fusion technology. This design is both innovative and practical.

2. Construction of System Model

2.1. The Basic Function of the System. According to the needs of real-time detection of airport runway FOD, FOD detection system should have the following basic functions.

(1) The runway should be detected 24 hours in all cases, regardless of the weather and the limitation of light.

(2) It can detect the damage level and the specific location of FOD and sends out alarm signals correspondingly.

(3) Information detected can be sent to the airport runway information database and recorded for query.

2.2. The System Structure Diagram. As the showing of Figure 2, the overall framework of airport runway detection system designed in this paper is to use the data server, data monitoring device, FOD processing information station and sensing data terminal and communication network to exchange information. First, the communication network receives information from each module and processes

Figure 1: (a) Tarsier system. (b) FOD Detect detection system. (c) FOD Finder monitoring system. (d) iFerret monitoring system.

the received information. And then, it feeds back to each module and completes the specific process of FOD detection. At last, the content of the event is sent to the database.

2.3. The Work Flow of the System

(1) Turn on the system; staffs take their place; start the hardware and software system self-check.

(2) Millimeter wave radar began to carry on the omnidirectional real-time scanning of the runway.

(3) Radar scans to find the FOD and alarm. Optical camera began to take pictures of FOD, judge the harm degree, and send the location information of the FOD.

(4) Monitor center receives information and judge whether to remove. If the optical camera failed to identify, the manual inspection should be adapted.

(5) After receiving removal instructions, the staff began to clean foreign body, upload information to the monitor data center, and record the information.

3. The Sensor System

3.1. Millimeter Wave Radar. Millimeter wave radar usually works in the frequency range of 30 to 300 GHz. The main function of millimeter wave radar is to scan the runway in the automatic inspection uninterrupted state all day and sound the alarm in the detection of FOD [5]. Then send the detected information to optical camera.

3.2. Optical Camera. Optical camera system is composed of visible light camera and an infrared thermal imager. Its main function is to shoot picture of the target after receiving distance coordinate signal of airport runway FOD given from the millimeter wave radar system [6].

Visible light camera is mainly used for photographing in illumination and better weather conditions. FOD target environment image and FOD accurate target image are very important for the detection and identification of targets. Therefore visible light cameras can take both the whole scene image and local fine image features by selecting the zoom capability lens.

3.3. Laser Range Finder. Airport FOD detection uses laser range finder to fix the target location. When visible light camera and infrared camera scan the object and send out the alarm signal, laser range finder measures the distance of the object and determines the exact location of FOD at the same time.

3.4. Photoelectric Encoder. Photoelectric encoder is a binary photoelectric position indicator. It is composed of light source, the encoder, and a photoelectric receiver. Airport runway detection uses photoelectric encoder to determine the angular position for more accurate position of FOD.

3.5. Data Fusion Device. FOD detection uses data fusion for information processing. The data fusion center fuses the information from multiple sensors and can also fuse

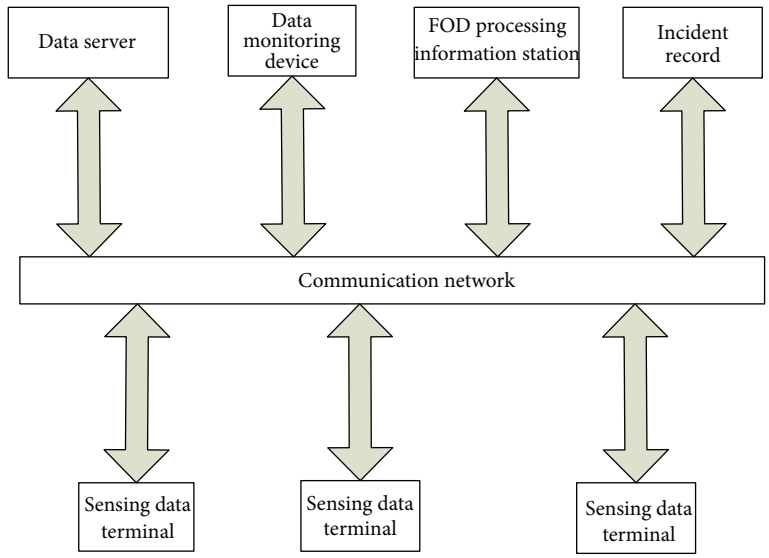


Figure 2: The system structure diagram.

the observed facts from multiple sensor information and the man-machine interface. If finds foreign objects and extract information, makes an alarm signal and send it to the monitoring center or runway staff, then process and record information.

4. Airport Runway Detection Algorithm

According to the analysis of airport runway detection system, FOD detection is very important. Detection algorithm of image objects is the priority among priorities in the algorithm design process. Therefore, in order to accurately locate and recognize the position of FOD, we will use several kinds of algorithms to analyze this problem.

4.1. Analysis and Comparison of Image Segmentation Algorithms

4.1.1. Image Edge Detection Algorithm Based on the Canny Operator. (1) The basic principle of Canny edge detection: the first-order differential filter can filter out the noise and keep the edge characteristic. Noise filter is a derivative of arbitrary direction of 2D Gauss function [7, 8]. Filtering method is carried out through the convolution with the image; image edge is determined by the local maximum image gradient on the filtered image.

(2) Theoretical basis: Canny gives 3 indexes of edge detection performance evaluation [9]. High signal to noise ratio. The mathematical expression of the signal-to-noise ratio (SNR) is

$$\mathrm{SNR} = \frac{\left|\int_{-w}^{w} G(-x)\, f(x)\, dx\right|}{\left[a\sqrt{\int_{-w}^{w} f^2(x)\, dx}\right]}. \tag{1}$$

The boundary filter impulse response is $f(x)$; the edge function is $G(-x)$; the RMS of Gauss noise is a. The false detection rate is inversely correlated to SNR.

High positioning accuracy: in order to increase the positioning accuracy, edge detection is less than the actual distance of edge point. Mathematical expressions of positioning accuracy are

$$\text{localization}\ (f) = \frac{\int_{-w}^{w} G'(x)\, f'(x)\, dx}{\left[a\sqrt{\int_{-w}^{w} f'^2(x)\, dx}\right]}. \tag{2}$$

The first-order derivatives of $G(-x)$ and $f(x)$ are $G'(-x)$ and $f'(x)$. The positioning accuracy is proportional to the localization value.

Single edge response means that one response corresponds to only one edge points, which reduces the probability of false edges appearing. Detection of average distance $D(f)$ of zero crossing pulse response derivative operator is to ensure a single edge responses. Among them, the second derivative of $f(x)$ is $f''(x)$. That should meet

$$D(f) = \pi\left[\frac{\int_{-\infty}^{\infty} f'^2(x)\, dx}{\int_{-\infty}^{\infty} f''^2(x)\, dx}\right]^{1/2}. \tag{3}$$

Set 2D Gauss function

$$G(x, y) = \left(\frac{1}{2\pi a^2}\right)\exp\left[-\frac{x^2 + y^2}{2a^2}\right]. \tag{4}$$

(3) The advantages of Canny operator detection method [10] are as follow:

(1) the low bit error rate;

(2) the high positioning accuracy;

(3) the high false edge inhibition rate.

(4) There are some possible defects in the following aspects when the Canny operator detects the edge of the runway.

(1) The Canny operator uses the Gauss filter for image smoothing, airport runway background image illumination changes, and Gauss filter coefficient is difficult to determine. So detection of airport runways is instable.

(2) Localization of Canny operators on the edge is more accurate, but it is too sensitive to noise, the brake marks and runway centerline at the airport runway background image are likely to be mistaken for the wrong edge information.

(3) The high and low threshold parameters of Canny algorithm is set artificially, so it is not adaptive. High and low threshold setting have a great influence on the result of edge detection.

The existence of these problems may affect the effect of the application of Canny algorithm in detecting FOD in the airport runway, but for many years, the image in different areas, different characteristics, and many scholars put forward many improved algorithms about Canny operator. According to the characteristics of the airport runway background image, using adaptive double threshold method will get a good effect on the airport runway edge detection by improved Canny operator.

4.1.2. Runway Edge Detection Algorithm Based on Hough Transform

(1) Runway Edge Detection Algorithm. Detection of airport target in remote sensing images often needs to extract straight lines corresponding to the runway; therefore we use Hough transform. Hough transform is a commonly used line detection method. It has the advantages of detection and effective recognition on the target, being not sensitive to noise, and so forth. Because it has these advantages, it has been fully utilized in the detection of airfield runway. The disadvantage of the Hough transform is that the computation is very large, so it is limited in application. In order to solve this problem, people have improved algorithm. In this paper, using the phase information of straight to reduce the total computation of Hough transform greatly has achieved remarkable effect.

(2) The Principle of Hough Transform Line Detection. Hough transform is an effective treatment on linking and finding the line segments in the image. The basic idea is to establish a correspondence of "line-point" between image space and parameter space and turn the straight detection problem in image space into the parameter space point detection problem. There are two kinds of parameter spaces commonly used in the Hough transform. One kind is slope intercept parameter space; the other kind is the polar parameter space. In the establishment of parameter space with the slope a and intercepts b of the line, every point on the line $L0 : y = a0x + b0$ in the image space represents a straight line in the parameter space.

The commonly used method of Hough transform line detection is the use of the following formula to establish duality transformation between image space and the polar coordinate space:

$$x \cos\theta + y \sin\theta = \rho, \tag{5}$$

where ρ is polar radius; θ is the polar angle, 0~180°.

In order to detect the line composed of nonzero point in rectangular coordinate system, the ρ should be discretized into $N\theta$ parameter spaces, and the ρ is discretized into $N\rho$ parameter spaces, according to the requirements of resolution of detection.

This method is called the standard Hough transform method. It has the advantage that range θ and ρ in the parameter space is limited, no matter how changes in line. So the line detection methods are basically based on this method at present. But this method has the disadvantages of needing large amount of calculation and a large storage space in case of larger values of N.

The runway detection algorithm does not require the straight line extraction of high resolution in the airport runway region segmentation. But the white airport runway central line and the black tire skid marks must be removed effectively. Therefore, the improved algorithm should have better judgment on collinear line segments belonging to the airport runway boundary or interference line. Moreover, because of real-time requirement, the computational complexity of the algorithm cannot be too high.

5. Application of Data Fusion in the Airport Runway Detection System

5.1. Data Fusion Technology in This System. Multisensor data fusion is a new research field. Practice has proved that compared with the single sensor system, multi sensor data fusion technology can enhance the survival ability, improve the reliability and robustness of the whole system, strengthen confidence, improve the reliability and robustness of the whole system, strengthen confidence, and improve the precision of the whole system, expand the system time, space coverage ratio, spatial coverage in solving detection, tracking, target recognition. Because of this, data fusion technology is widely applied in the field of airport runway detection [11].

People put the information from each sensor for the detection of FOD on the runway together by using data fusion technology and detect the real-time information from surrounding environment of the airport runway.

In the data fusion system of airfield runway FOD detection, the information from sensors has different characteristics: real-time or non-real-time, fast or slow, fuzzy or clear, mutual support or complementary, maybe conflicting or competitive. The basic principle of multisensor data fusion is making full use of the different functions of multiple sensors and division of labor and cooperation between different sensors. In order to explain or describe the object measured congruously, various pieces of information of sensor detection are optimized. Therefore, the data is no longer a single

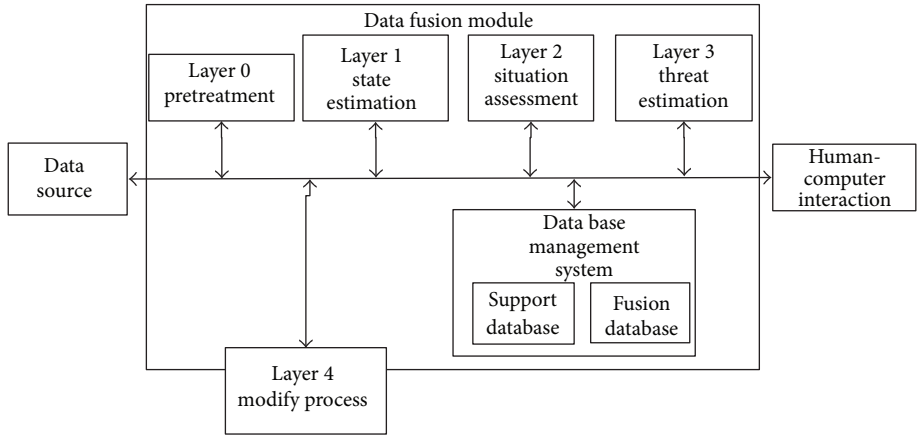


Figure 3: Data fusion module.

input information but more information and better information from multi sensor. This improves the effectiveness of the entire sensor system [12, 13].

5.2. The Function Model of Data Fusion. Application of data fusion model in different area is different. The ultimate goal of data fusion is realized by four different processing layer levels. Figure 3 shows the signal processing in different fusion level.

Layer 1. This layer is the level of processing layer. It fuses the information of position and identity from single sensor, so as to obtain the FOD information of position and identity.

Layer 2. Layer 2 is to describe and explain the current situation, static state and dynamic situation, and the link between events through processing the observation data and a series of events. The data from layer 1 is obtained by layer 2, so we can have more indepth understanding about the FOD and surrounding environment of the airport.

Layer 3. This layer judges the surrounding environment and threatening grade of airport runway FOD and then makes a decision. Threat assessment is different from the situation, because threat assessment will analyze FOD quantitatively in many aspects, so as to determine the position of the FOD.

Layer 4. This layer can be called optimal fusion layer. By modifying the whole process and enabling the efficient use of resources, the sensor can be effectively managed. The main function of this layer is to monitor and evaluate the fusion process and to guide on how to get the data, which can achieve a better fusion result. The processing layer and layers, external system, and system operators have contact.

In the high-level fusion process, we need to use a large database. Quickly adding data and retrieval of this large databases must be realized [14, 15].

The hierarchical model emphasizes the various steps of data fusion processing, rather than computer structure. When processing layer moves from one layer to the other three layers, the model emphasizes the level. Layers of processing, fusion data generated.

6. Conclusion and Future Work

The present study aimed at the airport runway detection system, and we have two directions [5]. On the one hand, FOD single sensor targets detection and recognition in millimeter wave radar system or optical camera system based; On the other hand, combined with the multi-sensor detection system has become the mainstream mode of airport FOD detection, rapid response to FOD target and alarm, reduce the FOD threat to flight safety. This paper designs the airfield runway FOD detection system based on sensor and information fusion technology.

On the basis of this study, there are many aspects worthy of further study on the runway foreign object detection, recognition, and alarm problems.

(1) In this paper, the airport runway detection in SAR complex scenes was done by the image fusion method. But how to better detect the single SAR image in the complex structure of the airport runway still needs further research and exploration.

(2) How to detect and recognize airport runway accurately in real time at the wide range of airport remote sensing image still needs a further study on detection and extraction methods of remote sensing images in the long straight road.

References

[1] F. Manjie, *Airfield Runway Foreign Body (FOD) Detection*, Beijing Jiaotong University, 2011.

[2] J. Patterson, *Foreign Object Debris (FOD) Detection Research*, Airport Safety Specialist, FAA, 2008.

[3] P. Feil, W. Menzel, T. P. Nguyen, C. Pichot, and C. Migliaccio, "Foreign objects debris detection (FOD) on airport runways using a broadband 78 GHz sensor," in *Proceedings of the 38th European Microwave Conference (EuMC '08)*, pp. 1608–1611, October 2008.

[4] Q. Xu, H. Ning, and W. Chen, "Video-based foreign object debris detection," in *Proceedings of the IEEE International Workshop on Imaging Systems and Techniques (IST '09)*, pp. 123–126, Shenzhen, China, May 2009.

[5] P. D. L. Beasley, G. Binns, R. D. Hodges, and R. J. Badley, "Tarsier. A millimetre wave radar for airport runway debris detection," in *Proceedings of the 1st European Radar Conference (EuRAD '04)*, pp. 261–264, Amsterdam, The Netherlands, October 2004.

[6] L. Song, *Research of the Unmanned Turret Sensor Intelligent System*, Shenyang Ligong University, 2010.

[7] S. Jinping, "Research of image segmentation method," *Modern Electronics Technique*, vol. 45, no. 4, pp. 34–35, 2006.

[8] Y. Xianhua, "Measured algorithm of image edge based on Matlab," *Research and Development of Electromechanical Products*, vol. 27, no. 7, pp. 6–8, 2010.

[9] Y. Jun, "Algorithm of image edge detection and extraction based on Canny operator," *Science and Technology Information*, vol. 23, no. 5, pp. 15–17, 2007.

[10] C. Yanlong, "Canny edge detection method based on particle swarm optimization," *Computer and Telecommunications*, vol. 7, no. 4, pp. 8–10, 2011.

[11] A. B. Poore, "Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking," *Computational Optimization and Applications*, vol. 3, no. 1, pp. 27–57, 1994.

[12] A. N. Steinberg and C. Bowman, "Revisions to the JDL data fusion process model," in *Proceedings of the National Symposium Sensor and Data Fusion (NSSDF '98)*, vol. 12, no. 4, pp. 23–25, Lexington, Mass, USA, 1998.

[13] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links—part I: distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, 2008.

[14] F. Xi, J. He, and Z. Liu, "Adaptive fast consensus algorithm for distributed sensor fusion," *Signal Processing*, vol. 90, no. 5, pp. 1693–1699, 2010.

[15] W. Yu, G. Chen, Z. Wang, and W. Yang, "Distributed consensus filtering in sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 39, no. 6, pp. 1568–1577, 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 153034, 7 pages
http://dx.doi.org/10.1155/2013/153034

Research Article

Data Deduplication in Wireless Multimedia Monitoring Network

Yitao Yang,[1,2] Xiaolin Qin,[1] Guozi Sun,[2] Yong Xu,[1,3] Zhongxue Yang,[1] and Zhiyue Zu[2]

[1] *College of Computer Science & Technology, Nanjing University of Aeronautics & Astronautics, Nanjing, Jiangsu 210016, China*
[2] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China*
[3] *Department of Computer Science & Technology, Anhui University of Finance & Economics, Bengbu, Anhui 233030, China*

Correspondence should be addressed to Xiaolin Qin; qinxcs@nuaa.edu.cn

Received 17 July 2013; Accepted 28 October 2013

Academic Editor: Zhijie Han

Wireless sensor network has been applied to many areas for a long time. A new kind of wireless sensors equipped with a camera and a microphone has been emerging recently. This kind of sensor is called wireless multimedia sensor (WMS) because it can capture and process multimedia data such as image, sound, and video. The visual monitoring network is a typical scenario of WMS application. Massive data would be produced in a short time because of the intensive WMS deployment. Many data aggregation and compression technologies have been proposed for addressing how to transfer data efficiently. However, data aggregation technologies need highly efficient router algorithm, and compression algorithms might consume more computation time and memory because of the high complexity. This paper applies data deduplication technology to this scenario. It can eliminate the redundant data from raw data to exploit the network bandwidth efficiently. Moreover, a chunking algorithm with low computation complexity is presented in this paper, and its efficiency has been proved through the experiments.

1. Introduction

Wireless sensor network (WSN) consists of a large number of spatially distributed autonomous sensors which are connected by wireless communication. It is mainly used for dynamic acquisition of physical information within the network coverage which will be delivered to users later. Currently, WSN is widely applied to the capture, processing, and transmission of the data such as temperature, light intensity, humidity, and gas concentration [1, 2]. In the recent years, as the production level of sensors improves, additional camera, microphone, and other functional devices are installed on traditional wireless sensors and enable them to capture, process, and transfer multimedia information such as image, sound, and video, so that users can obtain improved physical information which is more vivid and more accurate. This new kind of sensor can be called wireless multimedia sensor (WMS). Wireless multimedia sensor network (WMSN) consists of WMS nodes, gateway nodes with storage, sink nodes, and so on. Then, a typical WMSN architecture is proposed in [3], as shown in Figure 1. WMSN is widely used in many areas including visual monitoring, individual positioning, industrial control, intelligent transportation, environmental monitoring, smart home, and telemedicine [3–5].

Multimedia sensor nodes have limited resources such as energy capacity of battery, storage space, and computing power, while the information they acquire features large data size and high requirement for computing. The data in-network processing technology [6, 7], as one of the key technologies used to save bandwidth and network energy in data-centered WMSN, can eliminate the redundancy of source data and minimize the data traffic between nodes by applying data fusion technology or data compression. The data fusion technology uses different routing methods to combine data packages and eliminate redundant data from them. This technology relies on three basic modules:routing algorithm, data fusion, and data presentation [6]. However, in addition to the advantage of high data aggregation, the application of data fusion technology may also lose raw data structure, while data compression can prevent this problem. Data compression has been applied to WMSN, and the compression algorithm can be divided into two categories:distributed data compression scheme and local data compression scheme. The proposed data compression

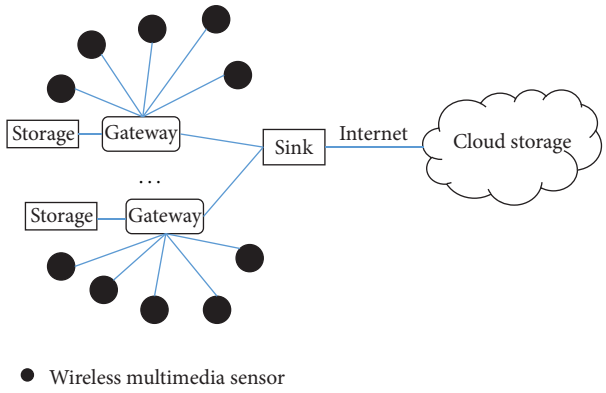


Figure 1: A typical system architecture of WMSN.

algorithm is the extension of data fusion technology in multiple-hop network topology, and compression algorithms are applied to the entire network [8, 9]. Other compression algorithms independently perform data compression on local node, so they do not depend on intensive network or routing algorithm and are more suitable for sparse sensor network [10, 11]. The data compression scheme can reduce the amount of data transmission in sensor networks to save bandwidth and energy consumption; however, as compression algorithm is relatively complicated and needs more computation time and memory, the energy cost of nodes that apply compression algorithm is much higher than that of nodes without using compression algorithm [12].

Event-based monitoring scenario is a typical application of WMSN. When the triggering condition of an event is met, the wireless multimedia sensor node will capture the image data at that moment and upload the real-time data to the storage device of gateway node. Then, the gateway node collects data from different sensor nodes and transmits the data to the cloud for storage and backup through sink node based on corresponding strategies [13]. In this scenario, the WMS node deployment in WMSN is very intensive and will produce a large amount of data in a very short period. These data have distinctive characteristics so it is suitable for applying data deduplication technology to this scenario. Data deduplication can effectively eliminate the redundant data to reduce the multimedia data traffic between nodes and improve the bandwidth utilization of WMSN. At the same time, its computation complexity is less than that of compression, which means less computation energy consumption. The chunking algorithm is the key factor that impacts the effect of deduplication. Basic sliding window (BSW) algorithm is one of the classical chunking algorithms. This paper proposes a specific chunking algorithm based on BSW. The algorithm can decrease system overhead and increase deduplication efficiency by merging continuous redundant chunks. In contrast with BSW, the experiment results will show how efficient the new algorithm is. The rest of this paper is organized as follows. Section 2 firstly describes the background of data deduplication and introduces BSW algorithm. A continuous redundant chunk-merging algorithm is proposed in Section 2.4; Section 3 performs a simulation experiment and takes the basic sliding window algorithm for comparison to analyze the advantages and disadvantages of both; Section 4 will draw a conclusion.

2. Data Deduplication

Data deduplication [14] is firstly applied in the field of storage backup to reduce storage costs and improve storage space utilization. The technology segments the source data S into a continuous chunk set $C = \{c_1, c_2, \ldots, c_n\}$ according to chunking algorithm. If small changes are made on S, S should be denoted as S' and should be segmented into a new continuous chunk set $C' = \{c'_1, c'_2, \ldots, c'_2\}$ by using the same algorithm. Suppose that S is already in memory; only the chunks in C' but not in C should be stored when saving S'. If $|C \cap C'| \to n$ $(n \leq m)$, that is, most of elements in C' are the same as the elements in C, and then the chunk sequences are regarded as stable. The more stable the sequence is, the smaller the storage cost is. Data deduplication can be applied in WMSN to reduce the amount of data transmission between nodes. If the WMS node is denoted as *sender* and the gateway node is denoted as *receiver*, the *sender* is ready to send the data S to the *receiver*. Assume that the *receiver* already has the data R and the *sender* knows what chunk in S is exactly the same as the chunk in R, so the content of these chunks needs not be transmitted, but only their index in R should be transferred. The *receiver* will find these chunks locally to restructure S according to the index to achieve the goal of saving network bandwidth. It is obvious that the chunking algorithm is important to ensure the stability of chunking sequence.

Data deduplication can be divided into file level, block level, and byte level according to operating granularity. The block level can be divided into fixed-size chunk and variable-size chunk. The smaller the granularity is, the more redundant data are removed, but the implementation complexity and system overhead also increase accordingly. The redundant data that file level can eliminate is the least, but it is easy to achieve. The byte level has the best deduplication effect, but its overhead is extremely expensive and difficult to implement. If fixed-size chunks are used, their matching probability will be greatly reduced due to fixed boundary. In monitoring networks, the position of multimedia sensor is generally fixed, so the images have a small amount of information changes. Good variable-size chunking algorithm can separate duplicate and nonduplicate information as much as possible to reduce the transmission amount of duplicate information. The content-based chunking algorithm applies sliding window technology to determine the boundary of the chunk based on file content. It can control the impact of update on data partitioning within a small range, so that only several chunks near the update location will be affected while other chunks remain unchanged. Therefore, this chunking algorithm is very suitable for the scenario above.

This section will first introduce basic sliding window algorithm and will propose an improved continuous redundant chunk-merging algorithm. Then, the advantages and disadvantages of both methods are analyzed in terms of system overhead.

2.1. Basic Sliding Window Algorithm. The process of basic sliding window algorithm [15] is as follows: a window with fixed size is moved cross the data, and, at every position in the data, a Rabin fingerprint [16] of data in the window is calculated. If an rf-*Match* appears in position k, then k is declared as a chunk boundary. Repeat these steps until it reaches the end of data. Finally, a sequence of chunks $c_1, c_2, \ldots, c_n$ is generated.

Definition 1. A Rabin fingerprint for a sequence of bytes $t_1, t_2, \ldots, t_\beta$ of length β is given by the following expression, where p and M are constant integers:

$$\mathrm{rf}\left(t_1, t_2, \ldots, t_\beta\right) = \left(t_1 p^{\beta-1} + t_2 p^{\beta-2} \cdots + t_{\beta-1} p + t_\beta\right) \mathrm{mod}\ M. \quad (1)$$

Definition 2 (rf-*Match*). For a given sequence $S = s_1, s_2, \ldots, s_n$, rf represents Rabin fingerprint function, l is the window size, and r is a constant integer; then, the rf-*Match* at position k is defined as follows: the last r bits of rf(W) value are zero, in which $W = s_{k-l+1}, s_{k-l+2}, \ldots, s_k$ is the subsequence of length l preceding the position k in S.

2.2. Analysis of Duplicate Data. In order to analyze the redundancy degree of the image data captured from the monitoring network, we simulated a scenario in the laboratory, where a path was left in the middle and fixed objects were placed at both sides. Excluding the impact of light change, a CMOS camera was fixed to monitor the scenario. If any objects appeared on the path, the camera would capture the images at that moment and store them in external storage immediately. We asked different groups of testers to move on the path, and a total of 300 monitored images were collected. All image data were stored in nondestructive bitmap format (BMP; resolution 640∗480; bit depth 16 bits). The file name was numbered in ascending order: $0, 1, \ldots, 299$. The 0th image was the initial scenario.

We made two groups of experiments. Group 1 applied byte comparison tool and took the 0th image as the benchmark to compare it with the remaining 299 images and record the number of bytes at identical parts. The experimental results are shown in Figure 2. It can be seen that, compared to the initial scenario, only the image value on the path is changing in other images, and nearly 78% of pixels in average remain unchanged. We call this part of data as redundant data that can be eliminated, namely, duplicate data.

Group 2 was used to verify the validity of BSW algorithm. In Group 2 experiment, BSW algorithm was used to segment all images and count the frequency that each chunk appeared. Theoretically, if the frequency is greater than 1, this means that the content of this chunk is only required to transmit one time. Before the BSW algorithm is implemented, three parameter values M, r, and l should be determined firstly. M determines the possibility of collision for Rabin fingerprint value, r directly affects the final size of chunk, and l determines the minimum chunk size. We built a hash table in memory to store the content of chunk, the 512 bytes SHA1 of chunk, and the frequency chunk appeared. The BSW algorithm was then implemented. Each time a new chunk was generated, the hash table was queried one time. If there were identical chunks, only one chunk was stored and its frequency was added as 1. If BSW algorithm was applied to segment the byte sequence set $S = \{S_1, S_2, \ldots, S_n\}$, then finally a chunking sequence set $C = \{\{c_{11}, c_{12}, \ldots, c_{1n}\}, \{c_{21}, c_{22}, \ldots, c_{2n}\}, \ldots, \{c_{n1}, c_{n2}, \ldots, c_{nn}\}\}$ was generated. If the frequency of each chunk was $k_{11}, k_{12}, \ldots, k_{1n}, \ldots, k_{n1}, k_{n2}, \ldots, k_{nn}$, respectively, then the duplicate chunk set that could be eliminated in C was represented by $C_{\mathrm{dup}} = \{c_{ij} \mid k_{ij} > 1\}$, and the ratio of duplicate chunk in S was represented by $\delta = |C'_{\mathrm{dup}}|/|C|$.

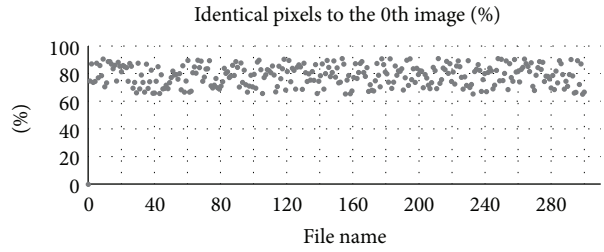


Figure 2: Percentage of identical pixels to the 0th image.

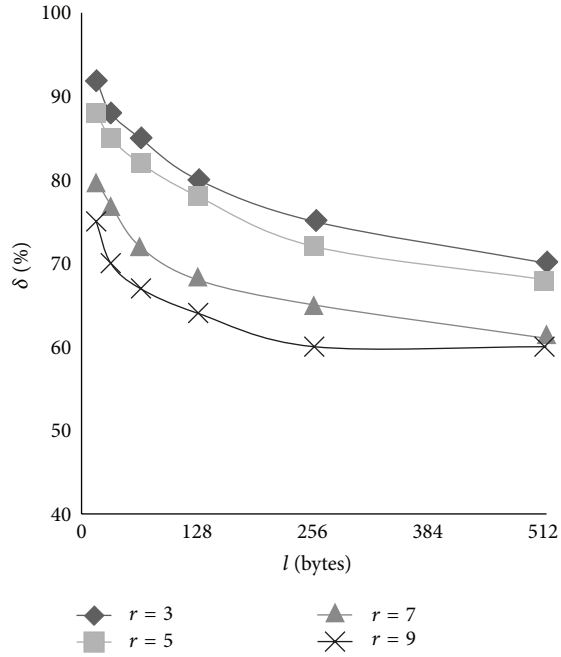


Figure 3: Impact of r and l on the ratio of duplicate chunk in S.

Considering that the sensor memory capacity was limited, we chose $M = 2^{30}$, and the collision probability of fingerprint at this length was small. Figure 3 shows different impact of r and l on δ, and, the same as in theoretical analysis, larger r and longer l led to the increase of chunk size and the decrease of chunk number, finally resulting in the decline of δ. However, the excess reduction of r and l would cause extra system overhead. In the experiment, we allocated 20-megabyte memory to hash table. The impact of r and l on the elimination rate of hash table is shown as in Figure 4.

The results show that, because the capacity of hash table was limited, excessive chunks would lead to the frequent elimination of hash table items, thus increasing the system overhead. Therefore, a more balanced choice was $r = 5$ and $l = 64$ bytes. The chunk statistic results with these parameter values are shown in Table 1. In a follow-up experiment, the two parameters selected these two values.

2.3. Evaluation Indicators. In this section, we introduce an evaluation indicator mentioned in [17] for evaluating our works.

For two byte sequences P and Q, use PQ to represent their concatenation. If $PQ = S$, then P is the prefix of S and Q is the postfix of S. $|S|$ represents the length of S in bytes.

For a byte sequence S, $H(S)$ denotes the chunk sequence $c_1, c_2, \ldots, c_n$ generated by a chunking algorithm on S. $\mu(S)$ = (total size of all chunks/the number of chunks) denotes the average chunk size in $H(S)$.

For the given threshold t, if (shared $(S, S')/|S| > t$), then S' is a local modification sequence of S, in which shared (S, S') refers to the sum of bytes with the same prefix and postfix of S and S', and it is defined that new $(S, S') = |S'| - \text{shared}(S, S')$.

$\Delta(S, S')$ is defined as the total size of chunks that appear in $H(S')$ but not in $H(S)$.

Definition 3. Assume that S is a byte sequence; S' is the local modification sequence of S. $V(S)$ is defined as follows: $\Delta(S, S') - \text{new}(S, S')$.

Definition 4. For a defined $V(S)$, the algorithm overhead index α is defined as follows:

$$\alpha = \frac{V}{\mu}. \tag{2}$$

With chunking algorithm, if duplicate data and nonduplicate data are divided into one chunk, the duplicate data in this chunk is an extra cost because it should not be stored and transmitted. V is the average number of bytes used to evaluate the overhead. The parameters r and l in Definition 2 can be adjusted to make V smaller, but this will also increase the metadata overhead, so μ is introduced to balance the overall cost. μ is the average chunk size which is inversely proportional to the metadata overhead and proportional to V. By calculating the ratio of V and μ, a comprehensive evaluation on chunking algorithm and cost of metadata can be achieved.

2.4. Continuous Redundant Chunk-Merging Algorithm. The chunks created by the BSW algorithm in Section 2.2 are not equal in size. The chunk size has very large impact on the algorithm effect, so we try to improve BSW algorithm by adjusting the chunk size. Group 2 experiment in Section 2.2 has recorded all chunks sizes. 79% chunk size is less than the average chunk size. If smaller chunks can be merged to reduce the total number of chunks, the average chunk length μ will increase; thus the overhead index α will decrease. For our simulation scenario, most of the pixels are basically unchanged. BSW algorithm is a content-based chunking algorithm, so the boundary between chunks is stable. For example, Figures 5(a) and 5(b) shows two continuous image files. The nonduplicate bytes part concentrates in one area, and the chunk boundary does not change in the first and the last duplicate bytes parts, so we declare that the chunk boundary is stable. If we merge the first three chunks and the last three chunks in Figures 5(a) and 5(b), respectively, total number of chunks will drop from 20 to 12, average chunk size μ will rise by 66%, and overhead index α will reduce by 39.7%.

We propose a continuous redundant chunk-merging (CRCM) algorithm. The algorithm works as follows: first, to determine the chunk boundary by applying BSW algorithm on files. Each time a new chunk is generated, the hash table is queried to determine whether the chunk has been stored before. If stored, the chunk will be sent to a data stack. If not stored, the algorithm continues to look for the next chunk and then repeat the above steps until meeting the end of

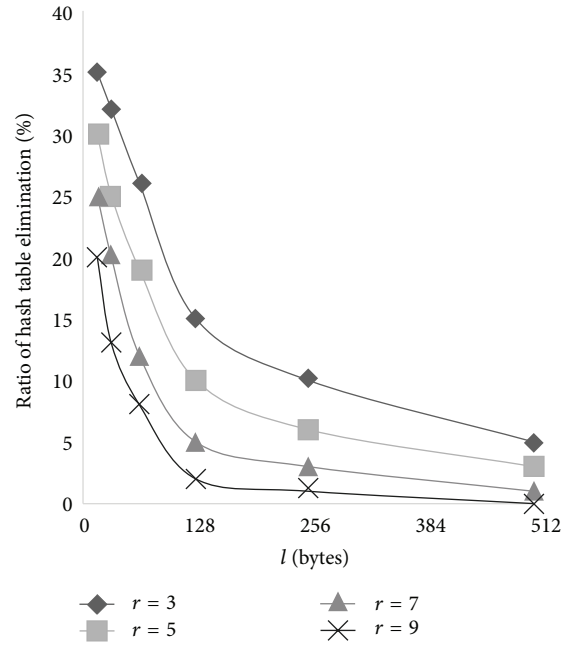


Figure 4: Impact of r and l on elimination rate of hash table.

Table 1: Chunk statistics of Group 2 experiment ($r = 5$ and $l = 64$ bytes).

Stat. item	Result
Total number of chunks	10068
Number of stored chunks	2395
Number of redundant chunks	8273
Percentage of redundancy	82.17%
Average chunk size (ACS)	19 K Bytes
Number of chunks whose size > ACS	2115
Number of chunks whose size < ACS	7953
Maximum chunk size	25 K Bytes
Minimum chunk size	1 K Bytes

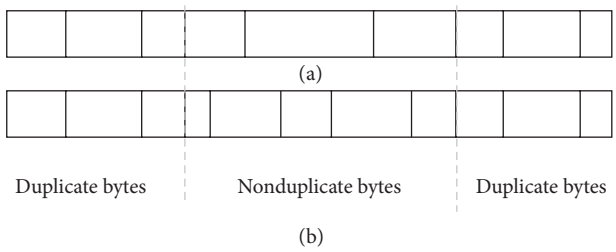


Figure 5: Example of stable chunk boundary.

file. The algorithm merges chunks according to the following rules: if the total length of chunk in the stack is greater than the threshold L, then merge all chunks in the stack to form a new chunk and insert it into the hash table. The detection of chunk merging is performed when a new chunk is generated.

Because the chunk boundary is stable, the probability of merging the same continuous redundant chunks is reasonably great when segmenting different files by applying CRCM algorithm. The algorithm effect is related to the threshold L and metadata costs. These will be discussed through the experiment in Section 3.

3. Result and Discussion

When wireless multimedia sensor is used for experiments in the simulation scenario, the following questions should be considered:

(i) hardware specification of WMS,

(ii) construction of wireless multimedia monitoring network,

(iii) data synchronization between nodes.

Computing power, storage capacity, and battery life of WMS have been improved greatly. The energy consumption of WMS is mainly in data processing, storage access, image acquisition, network communication, and so forth. Stargate [18] is a product of MEMSIC, and its hardware specification is Intel PXA-255Xscale 400 MHz CPU, 64 MB SDRAM, and 32 MB flash storage. Meantime, it also supports embedded Linux operating system and IEEE802.11 wireless communication protocol. Therefore, it is enough to complete some complicated computing tasks.

The wireless multimedia monitoring network composed of multiple WMS independently collects data and transmits information to sink node through gateway node. The computing power and storage capacity of gateway node are both far greater than those of WMS nodes. Multiple WMS can eliminate duplicate data together with gateway node at the same time, and the effect will be better, but this brings about a new problem of data synchronization between nodes. Some proposed synchronization methods such as distributed data deduplication [19] can address this problem.

This section will continue the experiment in the simulated scenario in Section 2.2. The chunking algorithm was replaced with CRCM algorithm, and then the evaluation was performed in terms of overhead index and PDR.

The WMS nodes in the experiment adopted the engineering board based on ARM9 architecture, equipped with 200 MHZ CPU, 64 M SDRAM, 1 GB flash storage, and onboard *QuickCam* camera. The maximum image resolution was 640 × 480, and the operating system was embedded Linux (2.4.19). Gateway node used DELL VOSTRO 3560 laptop, specification was the following: Intel i7-3632QM 2.2 GHz, 8 G RAM, and 250 GB hard of which drive. WMS node and gateway node used the IEEE802.11 wireless protocol to communication.

In the experiment, WMS node was used to monitor the simulation scenario. Any changes in the scenario would trigger sensors to capture and transmit real-time information to gateway node. CRCM algorithm was used for image chunking before the transmission of WMS node. Only those nonduplicate data chunks were transmitted. For duplicate chunks, only their index values were transmitted. The details of data synchronization between nodes would not be mentioned here.

Firstly, we observe the impact of threshold L on overhead index by running CRCM algorithm (see Figure 6). L is the multiple of the average chunk size of current file, that is, $L = x \cdot \mu$. When $L = 0$, CRCM algorithm was degenerated into the BSW algorithm and $\alpha \approx 2.5$. Greater L resulted in more merged chunks. When $L/\mu = 1.4$, the overhead index was the best, $\alpha \approx 1.3$, and the cost reduced by about 23.8%. As L continued to increase, the overhead was not reduced any longer. The reason was that if excessive chunks were merged, chunk size would be large, thus increasing the V value in α.

Metadata cost is another system overhead, so it must be involved in our evaluation. We focused on the metadata storage cost. WMS node's storage capacity is limited, so duplicate chunk data cannot be stored locally forever. Therefore, local metadata storage will encounter the problem of hash table elimination. We conducted four groups tests on different memory size allocated for hash table. The results were shown in Figure 7, and we can find that the ratio of hash table elimination by running CRCM was less than that of BSW algorithm, because BSW algorithm generated more chunks which needed more hash table items. According to our WMS hardware capacity, we considered 20 M would be a better choice in this scenario.

The computation complexity of both BSW algorithm and CRCM algorithm is much less than that of compression. The most time-consuming calculation is Rabin fingerprint calculation. Recalling rf in Definition 1, for fast execution, we can define an iterative formula as follows:

$$\mathrm{rf}\left(t_{i+1} \cdots t_{\beta+i}\right) = \left(\mathrm{rf}\left(t_i \cdots t_{\beta+i-1}\right) - t_i \cdot p^{\beta}\right) \cdot p + t_{\beta+i} \mod M, \tag{3}$$

where $t_i \cdot p^{\beta}$ can be precomputed and stored in a table. Rather than generating a new fingerprint from scratch, advancing rf calculation only requires three operations: a subtraction, a multiplication, and a mask mod.

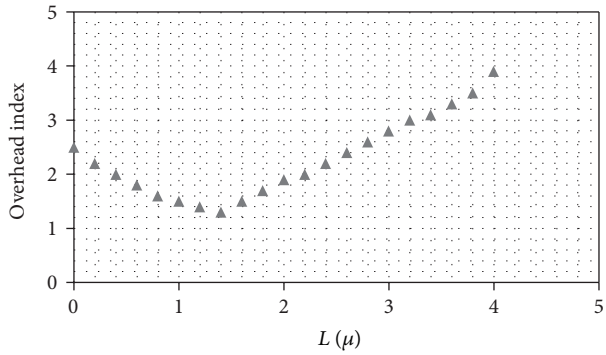


Figure 6: Impact of L on overhead index.

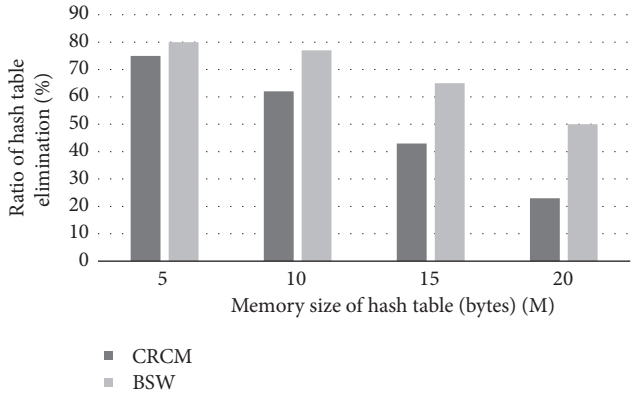


Figure 7: Ratio of hash table elimination by running CRCM versus BSW on different memory size.

4. Conclusion

In wireless multimedia sensor networks, massive data needs to be transmitted to the cloud storage. This paper applies data deduplication technology to WMSN. In the simulated monitoring scenario, this paper puts forward a continuous redundant chunk-merging algorithm by taking the characteristics of scenario data into full consideration. The algorithm considers the computing power and storage capacity of sensor hardware. Compared with BSW algorithm, it effectively reduces system overhead through experiments.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61373015 and 61272422), The Research Fund for the Doctoral Program of High Education of China (Grant no. 20103218110017), the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and the Fundamental Research Funds for the Central Universities, NUAA (Grant nos. NP2013307 and NZ2013306).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] Tubaishat, Malik, and S. Madria, "Sensor networks: an overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20–23, 2003.

[3] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.

[4] F. Hu and S. Kumar, "QoS considerations in wireless sensor networks for telemedicine," in *Internet Multimedia Management Systems IV*, vol. 5242 of *Proceedings of SPIE*, pp. 217–227, November 2003.

[5] J. Campbell, P. B. Gibbons, S. Nath, P. Pillai, S. Seshan, and R. Sukthankar, "IrisNet: an internet-scale architecture for multimedia sensors," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pp. 81–88, New York, NY, USA, 2005.

[6] J. N. Al-Karaki, R. Ul-Mustafa, and A. E. Kamal, "Data aggregation and routing in wireless sensor networks: optimal and heuristic algorithms," *Computer Networks*, vol. 53, no. 7, pp. 945–960, 2009.

[7] Z. Lu, Y. Wen, R. Fan, S. Tan, and J. Biswas, "Toward efficient distributed algorithms for in-network binary operator tree placement in wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 743–755, 2013.

[8] A. Ciancio and A. Ortega, "A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting," in *Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, pp. 825–828, March 2005.

[9] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Proceeding of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 1–10, April 2004.

[10] F. Marcelloni and M. Vecchio, "An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks," *Computer Journal*, vol. 52, no. 8, pp. 969–987, 2009.

[11] Liang, Yao, and Wei Peng, "Minimizing energy consumptions in wireless sensor networks via two-modal transmission," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 12–18, 2010.

[12] K. C. Barr and K. Asanović, "Energy-aware lossless data compression," *ACM Transactions on Computer Systems*, vol. 24, no. 3, pp. 250–291, 2006.

[13] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 4, pp. 18–39, 2008.

[14] Q. He, Z. Li, and X. Zhang, "Data deduplication techniques," in *Proceeding of the International Conference on Future Information Technology and Management Engineering (FITME '10)*, vol. 1, pp. 430–433, Changzhou, China, October 2010.

[15] A. Muthitacharoen, B. Chen, and D. Mazieres, "A low-bandwidth network file system," *ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, 2001.

[16] Rabin and O. Michael, "Fingerprinting by random polynomials," Center for Research in Computing Technology, Aiken Computation Laboratory, 1981.

[17] K. Eshghi and K. T. Hsiu, "A framework for analyzing and improving content-based chunking algorithms," Hewlett-Packard Labs Technical Report TR 30, 2005.

[18] M. O. Farooq and T. Kunz, "Wireless multimedia sensor networks testbeds and state-of-the-art hardware: a survey," *Communications in Computer and Information Science*, vol. 265, no. 1, pp. 1–14, 2011.

[19] Y. Zhang, Y. Wu, and G. Yang, "Droplet: a distributed solution of data deduplication," in *Proceeding of the ACM/IEEE 13th International Conference on Grid Computing (ICGC '12)*, pp. 114–121, Beijing, China, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 352062, 10 pages
http://dx.doi.org/10.1155/2013/352062

Research Article

Resilient Multipath Routing Mechanism Based on Clustering with Intrusion Tolerance

Shukui Zhang,[1,2] Hongyan Zuo,[1] Jianxi Fan,[1] and Juncheng Jia[1]

[1] *School of of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjin 210093, China*

Correspondence should be addressed to Shukui Zhang; zhangsk2000@163.com

Received 12 July 2013; Accepted 11 September 2013

Academic Editor: Chongsheng Zhang

Wireless sensor networks (WSNs) integrate sensor technology, communication technology and information processing technology; it is a synthetic discipline. WSNs have been applied into almost all walks of life. However, in many special fields, network security is a basic and important factor which we must concern for. This paper introduces the key management scheme after analyzing the security threats of LEACH protocol. At the same time, we adopt several related mechanisms to protect the communication among the sensor nodes and the confidentiality of the data transmission. Also, we use the mixed multipath mechanism among the clusters to improve the security of the network and strengthen the intrusion tolerance.

1. Introduction

The single path in WSNs just created the optimal path which is from the source node to the convergent node to respond timely to the specific mission requirements, which can save space for storage resources, reduce each communication traffic, and simplify algorithm, but the limited bandwidth will lead to the fact that network reliability is poor and fault tolerance is weak, and a node failure could lead to the fact that the whole link is not available, so designing reliable routing protocol is needed to improve the reliability of data transmission [1]. Multi-path routing is that among the source node and the destination node multiple paths are established, which use the primary path for data transmission; if the primary path fails, suboptimal path continuing for delivery will be selected from the backup paths. Disjoint multi-path is the establishment of a number of no convergent node paths among source node and convergent node, while, if the primary path of the way of a node failses, it will cause the entire link paralysis. In order to solve the problems that winding path introduced, this mechanism allows multiple intersecting nodes among the backup paths and the winding path. Each of the nodes on the primary path has a winding path, multiple winding path as an alternate path to constitute the primary path winding multipath. If a node is not available on the primary path, the routing protocol will select automatically the winding path that bypasses the fault nodes to transmit data to the node, so through the multi-path mode the reliability of the data transmission can be improved [2].

Routing protocol influences the performance of the network importantly, which can be divided into flat routing protocol and cluster routing protocol in the network topology view. In flat routing, each sensor node's status is equal and has the same routing function, which can transmit data through other nodes in the middle, so such structure is simple and easy to maintain and has good robustness. But when the number of nodes is large and the network scale is huge, the management and maintenance of routing will consume more energy, scalability relatively will be poor, as a transit node, which will transmit data more times, and energy consumption will be faster, so the structure is only suitable for small and medium-sized network [3].

Directed diffusion protocol [4] is a query-based flat routing protocol, and routing takes periodic updates and maintenance to adapt to changes in network topology. Convergent node sends query tasks, using the flooding way to spread the interested message to all sensor nodes in the monitoring area. In the process of information dissemination, protocol creates

a gradient which reversely transfers data from the source node to the convergent node. Then the convergent node finds the strengthening path which is the earliest path from the source node to the convergent node path and informs the source node. Eventually, the source node transfers the data collected to the convergent node by this route. The algorithm has good scalability, but both the energy consumption and routing overheads are larger.

The sensor node itself has a small size, it is easy to replace the battery, and so on, so that we must first consider is the energy consumption problem. Now clustering topologies controlling the network are commonly used. In order to balance the network node energy consumption, periodically cluster head selection is usually carried out. Clustering routing is suitable for networks which deploy numbers of nodes and have a large scale. LEACH (Low Energy Adaptive Clustering Hierarchy) protocol [5] is an adaptive clustering topology mechanism, which uses periodic execution process, randomly elects cyclic cluster head, and balances energy distribution networks to each sensor node. The agreement introduces a "round" concept, which stages in the establishment of the cluster head. The node generates a random number between 0 and 1, and if this number is less than $T(n)$, the node will become a cluster head and the elected node will message to inform other nodes that it is a cluster head. Other nodes are usually based on their received signal strength to select a member of clusters and send the message notification to add the cluster head in the cluster. After cluster head receives the request which all nodes join it, the cluster head gives them the communication slot allocation. In stable transmission phase, member nodes of the cluster head node receive the broadcast query request, which is monitored area of data collection, and then send data at one hop to the corresponding slot cluster head. After a period of time, cluster head collects data of which all member nodes were the data sent, and fusion finally sends the result to the base station. As in formula (1), random number $T(n)$ is expressed as follows, where p is the head node of all the nodes in the cluster percentage, r is the current round number, $r \bmod (1/p)$ represent elected cluster heads, the G is nonelected cluster heads:

$$T(n) = \begin{cases} \dfrac{p}{1 - p\left(r \bmod (1/p)\right)}, & n \in G, \\ 0, & \text{other}. \end{cases} \tag{1}$$

However, clustering mechanism will have problems such as the uneven distribution of cluster heads, heavy tasks or fast energy consumption issue that makes them stop working soon, affecting the data transmission or even the entire network topology. To solve this problem, a lot of articles used double cluster head algorithm [6] and join assistant cluster head clustering algorithm [7] to a certain extent, reducing the burden of cluster heads and the total energy consumption of the node. This paper summarizes some of the existing clustering algorithms; we propose a new mechanism to enhance the network fault tolerance.

However, WSNs and the rapid development of related technologies have strong practicality, which makes network security vital and relates people to the hot issue. The network is generally used to collect and process the data which is difficult to reach in harsh environments. Most of the sensor nodes are not tamper-resistant or anticapturing ability; once a node is captured, an attacker can get through information from this node to other nodes in the network. It makes the attack nodes send some error message or do not communicate with other nodes, which even leads to communication link failure resulting in great loss of data. Sensor network security risks are often precise because of the bad node deployment area or the communication methods used. Therefore, it should solve the transmission of information confidentiality, integrity, authenticity, and intrusion detection problem, or even if the network is attacked by malicious ones, and how to maintain their basic communication function is very important [8]. In this paper, we proposed a new scheme which is aimed at analyzing security issues which LEACH protocol security causes, and then we combines the inherent characteristics of the network, whose purpose is to reduce the chance in which the data transmission process information is stolen; nodes are posing or malicious nodes tamper with the information; this makes the node in the process of transmitting data even if the encountered attack will also be able to normalize collaboration work to complete basic tasks.

2. Description of the Problem

Cluster head bears most of the LEACH protocol tasks such as data fusion, forwarding, and communication with the base station. Cluster head task is heavy, which makes of fast energy consumption, but also in the data transfer process, it is more vulnerable to malicious attacks and exposure the transmitted information. It leads to poor network security, and thus the node does not complete the task. Here is a summary of LEACH vulnerable to attack in the following three items [9].

(1) Sybil attack: it refers to a single node emerging with multiple identity. In the cluster head election phase, attacker's multiple identities will lead to many nodes are easy to selects as the cluster heads. Elected malicious node controls most of the data node and further damages of the network.

(2) Hello flooding attacks: according to the means which cluster head send the signal strength, member nodes choose to join a cluster. If a malicious attacker has more energy flooding the entire network broadcast a manner notice strength, many nodes will choose to join this cluster where the malicious node exits, then the network will be vulnerable to the impact of these malicious nodes, which even will cause the network to be unavailable.

(3) Selective forwarding attack: such attacks mean that if a malicious node receives a packet, it will not be forwarded directly to the data integrity, but part of it will be lost or tampered before sending, so that the data cannot be accurately inevitably lead to reach the destination, and this is the loss of capture authenticity and integrity of the data.

The trust evaluation mechanism [10] can discover uncooperative nodes in the network, which can be used to reduce

the probability of being attacked, thus improving data integrity, security, and confidentiality. If the node trust value is introduced, it will be removed from the network after being determined in some way after which we can improve the probability of internal network attacks. But often there are many external malicious network attacks, such as theft, tampering and spoofing. How to take measures to simultaneously solve both internal and external attacks is a problem worthy of study. The literature [9] combines cryptography and proposes a secret sharing technology with intrusion tolerance capabilities clustering routing scheme to improve the data transmission network intrusion tolerance capabilities. In order to reduce energy consumption and improve the route network security, the literature [11] introduced a secure boot program and node two-way mechanism evaluation to detect malicious node is directly removed from the network to enhance the security of the entire network.

In order to reduce the burden of cluster head and the energy consumption of nodes, this paper presents a special feature called "Daemon Node (DN)" whose job is head node selection, path creation, and so on. This node can also be called a "backstage node" or "service node" because even if other nodes are in sleep, they have been in working condition, unless their energy is depleted or there is collapse of the entire network, and the node will be extinct. Then we adopted and improved the literature [9, 11] studied the program, and proposed an Intrusion tolerance based on clustering multipath routing mechanism (ICRM), which uses secret sharing between the nodes; in some way the key is split and assigned to the sensor nodes in the network, and cipher text recovery must be carried out with a threshold common trusted node collaboration. Among neighbors, daemon nodes and cluster head is established the pairs of random key introduced evaluation mechanisms. Data transmission between clusters will use hybrid multi-path mode, hoping to improve the data on the network communication traffic and reduce the probability of the node to be captured, thus reducing the possibility of attacks on the entire network.

3. Network Model

n sensor nodes are randomly deployed in a square of side length L of the monitor area A. Intercluster communication distance at least is greater than the distance between the two cluster heads, and the network collected by cluster head set is connected.

Assume that the daemon node has the following properties: (1) not for data acquisition and integration but to receive the base station it sends query request tasks and broadcast to the member nodes; (2) it maintains the energy of each node, location and other information, and real-time update node status information; (3) it makes dynamic perception node in other cases, such as the new node adding or failure of the node exit; (4) among the cluster nodes and other daemon nodes, virtual path in established for data transmission, maintenance established routes; (5) it is used for controlling cluster head work, such as cluster head direction of data transfer. WSNs model [12] is as follows.

(1) Each sensor node for data transmission contains the same initial energy and each of which has a unique ID number.

(2) After the deployment of sensor nodes which cannot move freely, the base station position is fixed in one place outside of the monitoring area, the daemon nodes and the selected cluster heads position is connected and unified tag.

(3) According to the received signal strength to nodes positioned approximate distance to base station, but which do not know the specific location, or there is no GPS function to obtain accurate position information.

In clustering algorithm, the cluster size is a factor to be considered, for ICRM algorithm which is described as follows.

(1) Taking into account the time required to select a cluster head consumption problem, here is the introduction of the cluster head energy factor and in certain conditions randomly selected T_{CH} whose value is less than the threshold. T_{CH} reference LEACH protocol definition is

$$T_{\mathrm{CH}} = \begin{cases} \dfrac{p}{1 - p\left(r \bmod (1/p)\right)} \dfrac{1}{\sqrt{E_{\mathrm{init}}/E_{\mathrm{current}}}}, & n \in G, \\ 0, & \text{other}, \end{cases} \tag{2}$$

where E_{init} is the initial energy which the remaining nodes in addition to the daemon node have and E_{current} is the node's current energy.

(2) The area covered by each cluster radius is fix (R), which is the same size of each cluster. According to literature [13], R derived optimum radius of the cluster is defined as follows:

$$R = 2\sqrt[4]{\frac{2\pi \times S \times E_{\mathrm{DA}}}{27 \times n \times \varepsilon_{\mathrm{pa}}}}, \tag{3}$$

where S is the total area of the monitoring area, E_{DA} is the integration of energy consumption data 1 bit, and $\varepsilon_{\mathrm{pa}}$ is the energy for power amplification.

(3) For a more balanced distribution, the distance between adjacent randomly selected nodes requires more than D; D is defined as

$$D = 2R + \frac{q}{\sqrt{Lk\pi}}, \tag{4}$$

where q is a constant aimed at area A which can be adequately covered and is almost uniformly distributed.

(4) In order to ensure inter-cluster does not cover and all nodes must belong to a cluster, each cluster daemon node using radius fix (R) sends a message, the node which received information is added to the cluster where the daemon nodes is. If a node receives many messages, select the first notification to join and become a member of the cluster nodes, where each member node is only one hop from the cluster head node, and then the daemon node sends a

confirmation message to indicate that the node can join the cluster.

(5) The daemon node obtains the information or each node in the cluster, denoted as E_{res} for the cluster head residual energy, as E_{avg} for the energy average of all the nodes in the cluster. After the node works for some time, if E_{res} is less than E_{avg}, daemon node will reselect again the cluster head which dynamically updated the node with maximum residual energy.

(6) In this paper, the literature [5] uses the same first-order model of wireless communication.

The sender sends l bit data to the distance d of the energy consumption of the receiver which is

$$E_{tx}(l,d) = \begin{cases} l \times E_{\text{elec}} + l \times \varepsilon_{\text{mp}} \times d^4, & d \geq d_0, \\ l \times E_{\text{elec}} + l \times \varepsilon_{\text{fs}} \times d^2, & d < d_0, \end{cases} \quad (5)$$

L bit data received energy consumed is

$$E_{rx}(l) = l \times E_{\text{elec}}, \quad (6)$$

where E_{elec} is the sending or receiving circuit that consumes energy. d_0 is the distance threshold value:

$$d_0 = \frac{\sqrt{\varepsilon_{\text{fs}}}}{\sqrt{\varepsilon_{\text{mp}}}}, \quad (7)$$

where ε_{mp} is a multichannel attenuation model power amplification factor; when sending the distance is greater than or equal to d_0, multiple attenuation model will be used; ε_{fs} is the power amplification coefficient in the free space model; when the transmission distance is less than d_0, the free-space model will be used.

4. ICRM Working Process

Daemon node contains the encryption key K and the hash function $F()$, by which the data processing function is very difficult to launch a value. Outside the area the base station is in the monitoring of a security zone; all the nodes of the network ID and the key pool are loaded onto the base station, but also each daemon node of the cluster storing the key information which nodes in the cluster and nodes in other cluster nodes and maintaining a key ring, which this article assumes that daemon node has more energy and ability.

4.1. Authentication. Each daemon node is managed and maintained in its cluster, real-time monitors of each cluster node, in which the sole is representative of the cluster where the nodes is, so need to be authenticated to the base station. Here BS is the base station, DN is the daemon node, DN_i is the ith ($i = 1, 2, \ldots, m$) node, A $\rightarrow$ B{C} means that A sends the message C to B, and ACK is the identification.

This phase is as detailed below:

(1) DN_i sends status request message REQ to BS:

$$\text{DN}_i \longrightarrow \text{BS}\{i, \text{REQ}\}; \quad (8)$$

(2) after BS received, find DN_i corresponding ID_i (daemon node identifier) which is calculated $F(\text{ID}_i)$ by $F()$, then calculate $V_i = R_i + F(\text{ID}_i)$, which R_i is selected random value according to different daemon node. Then it is cryptographically certified $\text{MAC}_{R_i}(\text{ACK})$ by R_i, finally BS sends a message to DN_i:

$$\text{BS} \longrightarrow \text{DN}_i \left\{V_i, \text{MAC}_{R_i}(\text{ACK})\right\}; \quad (9)$$

(3) after DN_i, receipt of the message, calculate $F(\text{ID}_i)$ by $F()$, draw $R_i = V_i - F(\text{ID}_i)$, and decrypt $\text{MAC}_{R_i}(\text{ACK})$ by R_i to get ACK. Then DN_i is also a randomly generated number, R_i^*, calculated as $V_i^* = R_i^* + F(\text{ID}_i)$, send a message to BS:

$$\text{DN}_i \longrightarrow \text{BS}\left\{V_i^*, \text{MAC}_{R_i^*}(\text{ACK})\right\}; \quad (10)$$

(4) after BS receipt, calculate $R_i^* = V_i^* - F(\text{ID}_i)$, then decrypt whether the ACK by R_i^*, and if so, DN_i authentication is successful, or need to use retransmission mechanism.

4.2. The Key Distribution and Establishment. Daemon nodes communicate with the base station to inform the number and location of the cluster head. Each cluster head CH_i ($i = 1, 2, \ldots, m$) stores allocation key share. Key segmentation process [9] summarized as follows. The base station first use of encryption function ENC and E will be encrypted secret S, $E = \text{ENC}_k(S)$. Reuse some decomposition algorithm divide E into m obtaining $E_1, E_2, \ldots, E_m$, and assign to each cluster head. Then divide the entire key K into m obtaining $K_1, K_2, \ldots, K_m$. Finally put two tuples (E_i, K_i) ($i = 1, 2, \ldots, m$) as key shadow S_i and send to the cluster head CH_i.

To implement a secure connection between the nodes this requires the establishment the key, because of it includes different types of nodes, in order to reduce the degree of interaction between them; this section of different nodes uses different keys. This reference literature [14] is the key distribution of thoughts to solve the clustering model with daemon nodes that is the setup of all kinds of key nodes. Daemon node and base station are stored as preloaded master key, whose mainfunction is to prevent the network attack during initialization which causes information to be stolen, and K_m generate the key pair which will be automatically deleted. This phase is divided into three parts as follows.

The Establishment of a Key among Nodes and Base Stations. Base station and each daemon node according to K_m generate a session key is as follows:

$$\text{PK}(r) = E\left(K_m, R\right), \quad (11)$$

where R is random number for a base, with the daemon node's private key K_i ($i = 1, 2, \ldots, m$) encrypted obtaining R_{K_i} and sent to each daemon node. Communication between daemon nodes and base station in order to guarantee the authentication should be deposited in the key.

The Key Establishment between Daemon Nodes and Member Nodes. Member nodes distribute key share with reference

to the above process of cluster heads share the number for $n - 2m$ (where n is all the nodes numbers, $2m$ is the total number of the cluster heads and daemon nodes), and eventually every member node gets a key from the base station and then directly sends its ID and key to daemon node. The daemon nodes after receiving information query their own key ring to find if there is information matching; if so, they will send a confirmation the members of the cluster head ID and control the establishment of the connection among the member nodes. Use the encryption key pair $K_{\mathrm{DN},i}$ daemon nodes and the communication link between the ith ($i = 1, 2, \ldots, n - 2m$) member nodes, cluster heads, and member nodes communication link between key encryptions for $K_{\mathrm{CH},i}$.

The Key Establishment among the Daemon Nodes. According to K_m and $F()$, generate a unique key and key pair, such as daemon node a, ID for a identifier, and L_a as its position information [15], based on the two having the only key K_a as follows:

$$K_a = F_{K_m}\left(\mathrm{ID}_a, L_a\right). \tag{12}$$

Then each daemon node builds key pair with the neighbor daemon nodes in the cluster. Daemon nodes a and b in neighbor cluster, for example, a broadcasts its ID to its neighbor cluster after receipt of b and compares its stored information in the key ring, if there is corresponding information, to establish a connection between two nodes [14]. They set up the shared secret pair K_{ab} as follows:

$$K_{ab} = F_{K_m}\left(\mathrm{ID}_a, L_a, \mathrm{ID}_b, L_b\right). \tag{13}$$

If b did not correspond to a key, each of them will, respectively, send the message which is the request to establish a shared secret pair to the base station, including their ID. With the receipt of base station, assigne key for them and reply from the key pool, after a and b are received using the decryption key K and gain K_{ab}, finally to establish a connection.

Shared key to establish the schematic is shown in Figure 1.

4.3. Measurement of the Node. In order to improve the legitimacy of node identity in communication, we also need to test the credibility of cluster heads and member nodes; in this section by reference to the literature [11] the ideas of nodes for assessment, when their trust value reaches a certain threshold, are allowed to participate in data transfer; otherwise it is considered entrusted node and deleted from the web directly. This phase is divided into two parts as follows.

Cluster Heads to Member Nodes. Each cluster head is set to a value of moderate threshold T ($0 < T < 1$), using formula (14) calculated member nodes reputation value Cread_i ($i = 1, 2, \ldots, n - 2m$):

$$\mathrm{Cread}_i = \frac{\lambda_1 C + \lambda_2 B}{\left(1 - \lambda_1 - \lambda_2\right) S}, \tag{14}$$

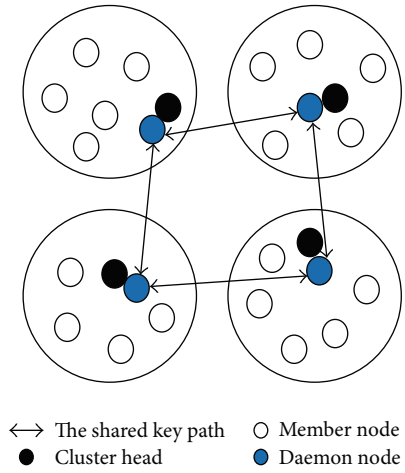


Figure 1: The establishment of a shared secret key.

where C is a member node sending data current, B is the sum of the member nodes to send data before, S is the total number which the member nodes send, λ_1 and λ_2 ($0 < \lambda_1, \lambda_2 < 1$) being both the corresponding weights.

If $\mathrm{Cread}_i \geq T$, members of the node are credible; if $\mathrm{Cread}_i < T$, mark them as malicious nodes.

If member nodes are malicious nodes, after daemon node receives cluster head signal board case, delete the message $\mathrm{mes}_{\mathrm{del}}$. Daemon nodes will store their ID but delete their carrying key. The evaluation mechanism shown in (15) by the way has already been leaked; keys can be deleted from the network, which increase the security of network.

Cluster Head in Member Nodes of the Assessment. Consider

$$\begin{aligned} &\mathrm{CM}_i \longrightarrow \mathrm{CH}\left\{\mathrm{ID}_{\mathrm{CM}_i} \parallel \mathrm{ID}_{\mathrm{CH}} \parallel E\left\{K_{\mathrm{CH},i}, \mathrm{Random}_i, \mathrm{mes}\right\}\right\}, \\ &\mathrm{CH} \longrightarrow \mathrm{CM}_i\left\{\mathrm{ID}_{\mathrm{CM}_i} \parallel \mathrm{ID}_{\mathrm{CH}} \parallel E\left\{K_{\mathrm{CH},i}, \mathrm{Random}_i, \mathrm{mes}_{\mathrm{del}}\right\}\right\}, \end{aligned} \tag{15}$$

where CM_i is the ith member nodes, CH is cluster head, $\mathrm{ID}_{\mathrm{CH}}$ is the cluster identifier, $E\{K\}$ is encryption using the secret key K, Random_i is the ith node which generated random number, and mes is sending information for members of the cluster head.

Member Nodes to Cluster Heads. Evaluation of the above (1) can prevent malicious nodes into the network, but in order to bring more security, it also needs member nodes evaluating cluster head to determine the current cluster heads which can be trusted. This mechanism is shown in the following.

Member Nodes in Cluster Head of the Assessment. Consider

$$\begin{aligned} &\mathrm{CM}_i \longrightarrow \mathrm{DN}\left\{\mathrm{ID}_{\mathrm{CM}_i} \parallel \mathrm{DN} \parallel E\left\{K_{\mathrm{DN},i}, \mathrm{Random}_i, \mathrm{mes}\right\}\right\}, \\ &\mathrm{DN} \longrightarrow \mathrm{CM}_i\left\{\mathrm{ID}_{\mathrm{CM}_i} \parallel \mathrm{DN} \parallel E\left\{K_{\mathrm{DN},i}, \mathrm{Random}_i, \mathrm{mes}_{\mathrm{del}}\right\}\right\}. \end{aligned} \tag{16}$$

What is mentioned above only represents single member nodes in cluster head of assessment; in order to improve the reliability of the results, all the member nodes are involved in the mechanism and then send the result to directly place daemon node in cluster, after daemon nodes received all member nodes in this cluster node test information, and then cluster head judges the credibility.

4.4. Multipath Routing. The literature [16] proposes an improved bulk density of the intrusion-tolerance ability of the routing protocol, to reduce the influence degree of the failure nodes on the network; the experiments show that the new routing protocol can effectively achieve data branching and improving the network capacity intrusion-tolerance ability. But in the literature, a common node in routing is just established, and this section will use the combination of disjoint path and winding path to the multi-path routing established between cluster heads. This phase is in the following three processes.

Build Paths. Because of the time of authentication and key establishing base station already known as daemon node location and distance information, daemon nodes need to send PATHREQ messages (including their own ID) in order to show how to begin the establishment of path. The base station feedbacks the PATHREV information to the daemon nodes after it receives the message, and according to the previously stored information we calculate the distance between them; finally each guard node maintains a routing table.

The Route Choice. First, using Floyd algorithm we calculate the shortest path from source daemon node to the base station and record every daemon node of the path. Then we build short path again, but no longer using the daemon node which is the primary path from source daemon node to base station which it passes by, for example, daemon node A midway through the B and C to base station, when the second shortest path is built; in other daemon node we expect the two nodes again using Floyd algorithm and calculate the second shortest path from A to the base station, as A backup path, and then press the same way to build A next second shortest path to the base station, as the second backup paths, and so on, which constructs the backup paths [17]. Finally using winding path algorithm, we construct the primary and second shortest path winding paths of each daemon node which makes every daemon node in at least one path. Building the multi-path routing diagram is shown in Figure 2.

But this only establishes virtual path among daemon nodes; the daemon node should command and control cluster head communication if daemon nodes and cluster heads position is once established, it will not be changed; even changes in the cluster heads are just changes of the location of the original cluster head and new cluster head. After the success which among cluster heads is the primary path, the second shortest path and winding path are set up and finally step into the stage of data transmission.

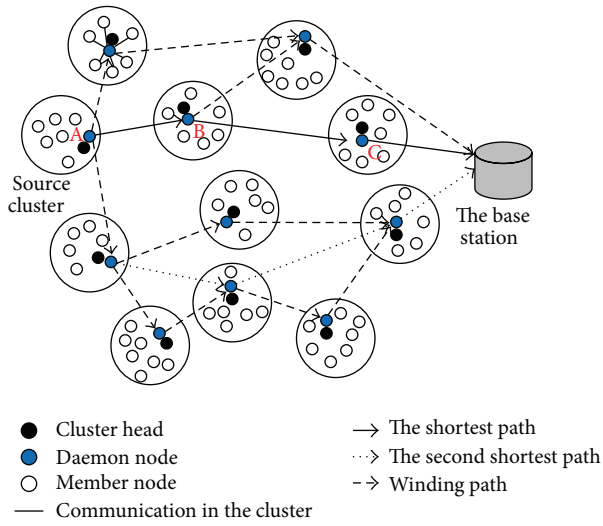


Figure 2: A mix of multi-path.

The Data Transmission Phase. Daemon node preserves the backup in which it and cluster heads have merged data. This section established the shortest path and the second shortest path; member nodes sent the collected information to the cluster heads, primary cluster fusion data and grouped packaging [18–20], a set of data using the shortest path transmission and another group using the second shortest path transmission. The two paths of the next cluster head receive data and fusion of information which it presses and continues to transfer until reaching the base station; base station receive two grouping of the data in the data reorganization, and also needs to get the key information. First we have to collect mS_i of cluster head CH_i ($i = 1, 2, \ldots, m$) on two links, with secret sharing scheme E and the secret key K to recover, finally by using the key K and the DEC decryption E obtaining S, $S = \mathrm{DEC}_K(E)$.

Through the data packet assigned to multiple paths, even if a cluster head is captured within the network, this will not leak the complete information of data. And each cluster head is established by winding path on the primary path; even cluster is attacked, and the data transmission will automatically bypass the failure nodes, and cluster heads will continue information transmission to the base station through the winding path, which can effectively enhance the network capacity invasion ability and improve the efficiency of data transmission.

5. Exception Handling

5.1. The Node Processing. Node failure may be because of its own characteristics or captured, which leads to network topology changes. Daemon nodes have real-timely monitory other nodes, when they find abnormal situation, which will diagnose and do the corresponding processing and store energy of each node information and trust degree, after a period of time if the cluster head energy value is lower than the preset threshold, according to the maintenance of information table selected residually more energy and high

trust value node for the next round of cluster head, cluster head message to broadcast again.

If cluster head is assessed as not credible, the daemon nodes will broadcast cluster head failure message to the member nodes. After the member node is received, it stops sending the collected data to cluster head in the cluster, and daemon nodes will delete the invalid cluster heads and reassign them. If a member node is not credible, daemon node will directly delete that node.

If a new node applies for joining a cluster, it will first send its key to daemon node in the cluster. After daemon nodes receipt, we need to check when a new node is key information and maintain consistent daemon node, the daemon node to perform this node to join operations, but we also need to see cluster head in the evaluation, after assessment can be involved in data collection, transmission, and so forth.

5.2. Routing Maintenance. Multiple paths are to work together, which may be because of a certain path to the cluster head that is attacked or lacks energy, which causes this path fails that then continues and to transmit data through a winding path; the failure path routing maintenance mechanism is adopted to repair in time, elaborated in two different conditions as follows.

(1) If a cluster head in the path fails, daemon nodes automatically sense and from the maintenance of information in the table select the most appropriate member nodes as the cluster head of in the cluster, the source cluster head and the new cluster head exchanged; the new cluster head immediately receives daemon node in the fusion of data backup, established under the control of the daemon nodes into transmission path in waiting for the next round of data transfer.

(2) If this is not suitable for cluster head, the routing failures also show that the cluster heads are unavailable and low energy nodes directly close the communication module of dormancy. Daemon nodes inform base station node, the automatic cluster scatter, the daemon node leaving virtual path; another daemon node is automatically connected to modify maintenance information again.

Using multi-path routing maintenance mechanism, if the path or the cluster heads appear to be problem, they can be restored or replaced in time. If there is not the cluster head which did not meet the conditions, failure of cluster's path directly removed makes some nodes into a dormant state, which can save energy and prolong the service life of them.

6. Performance Evaluation

It puts forward a scheme that is still on the basis of LEACH protocol. The main purpose is to enhance the security of data transmission and improve the reliability of the network; the following is performance of the detailed analysis of the mechanism.

6.1. Process Description. This paper built different keys among the various nodes, and between the two nodes is only key pair. First, the node's join needs verification. The nodes whose validation is not adopted are difficult to enter the network, the node which is successful to join but if the trust value is the lower it will not be able to participate in the transmission of the data. Before communication, each kind of nodes also needs node-to-node authentication, if this succeeded, data transmission can be carried out. Even if a node in the network running process is internal attack, leak is associated with being the attacked node itself, and the content of the other nodes can still work normally. Second, due to the introduction of node evaluation mechanism, cluster head in member node of the assessment can filter out certain malicious nodes, and member nodes of cluster head nodes review will also become a cluster head which has been dropped as a malicious node network, which can be at ease using cluster heads for data fusion and forwarding therefore, to some extent, this mechanism can be resisted. Section 2 summarizes the three kinds of attack.

The establishment of random key pair needs to broadcast each node number, so that we can save the communication traffic. Between source cluster heads and base station the hybrid multi-path mechanism is adopted, concurrent transmission data can be in the path, so that each path will reduce the quantities of data, so as to reduce the path of a single load, which is resolved in a single path on a cluster head energy consumption which is too fast and easily becomes death problem and further can reduce the topology change of path reconstruction or network reconfiguration overhead. In addition, even if building the shortest path, the second shortest path is even more than a certain number of cluster heads as the standby path being attacked, which can also be adopted as timely winding path to transmit data of each cluster head, daemon node using route maintenance mechanism, and the successful replacement of cluster heads to participate in the next round of data transmission, to improve network communication traffic as a whole.

6.2. Evaluation Indicators. This paper compares the ICRM and LEACH; DD agreement, analysis of performance indicators [18] are defined as follows.

Load balancing factor is

$$b_f = \frac{1}{n-m} N_{\text{low}}. \tag{17}$$

This value is used to measure the balance degree of clusters, where N_{low} is the number of energy which is lower than the average of the total number nodes in network, m is the number of daemon nodes, n is the total number of nodes, $n - m$ is the number of participating nodes in the data transmission.

The average end-to-end delay is

$$T_d = \frac{1}{n-m} \sum_{t \in T_s} t_d. \tag{18}$$

It is the average of the sum of the time delays which is transmitted packets from the source cluster heads to the

destination in T_s time, where is the t_d for each participating data transmission delay of cluster heads.

Network throughput is

$$T_p = (n - m)\frac{P_{\text{send}}}{T_s}. \quad (19)$$

It is the number of data packets of successful transmission in T_s time, P_{send} is each cluster head that sends the packet number.

The routing control overhead is

$$E_c = D_c + T_c, \quad (20)$$

where D is the pay expenses which are daemon node controlling according to the built path transmission cluster heads need, T is multiple paths in need of the overhead of data transmission.

6.3. Experimental Verification. Performance in order to verify the proposed algorithm uses c++ programming and corresponding analysis; in this section the ICRM and LEACH and DD protocol defined in the previous section on the evaluation index of were compared, and the running time is a measured parameter. 100 sensor nodes are randomly placed within 100 m × 100 m square area, because nodes are randomly deployed; location is not fixed, and there will be multiple nodes which are stacked in a place where nodes distribution is not uniform, causing no spread to some region; the random node distribution is shown in Figure 3. With base station coordinates of (120, 120), a sensor node range of (0, 0) to (100, 100) area, and node for the initial energy of 1 J, node death occurred when energy is 0, an experiment of assuming no daemon node death.

Figure 4 shows that clustering protocol LEACH in network load balance factor is better than that of flat DD routing protocol, which shows flat routing network load to be better than than clustering routing protocol. ICRM due to daemon node is introduced in the work, in the process of establishing virtual path, cluster head, and member nodes dormancy, in order to reduce energy consumption. In data transmission, the cluster heads packaged data packet transmission in multiple paths, which obviously decrease viral pathways on cluster head load and energy consumption. By comprehensive comparison, this paper presents the mechanism of the minimum balance network load factor and stability.

Figure 5 shows that the flat structure of multi-path routing protocols is used with the DD, compared with LEACH and ICRM clustering mechanism which is significantly smaller average end-to-end delay, which explains clustering structure division of each node clearly and is able to better coordinate the entire network. ICRM generated is the multipath among daemon nodes; cluster heads need to be controlled for data transmission, which is faster than flat routing path of each node speed, whose LEACH from the base station in the remote node increases path establishment and data transmission delay, so the ICRM is average minimum delay here.

Comparison on throughput is shown in Figure 6, because the ICRM introduced invasion mechanism, and it can effectively resist malicious nodes to participate in the network;

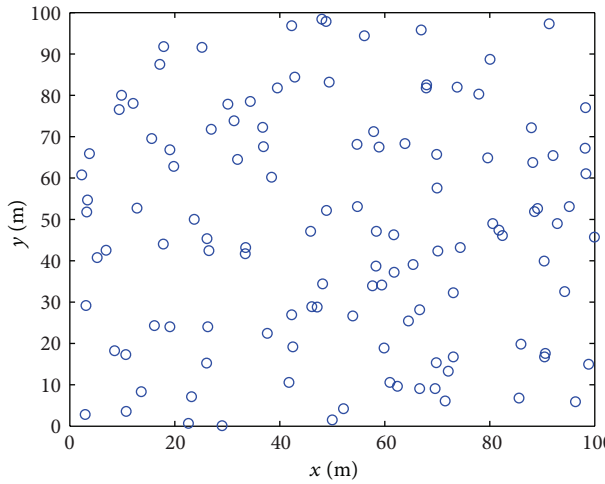


Figure 3: Nodes random distribution.

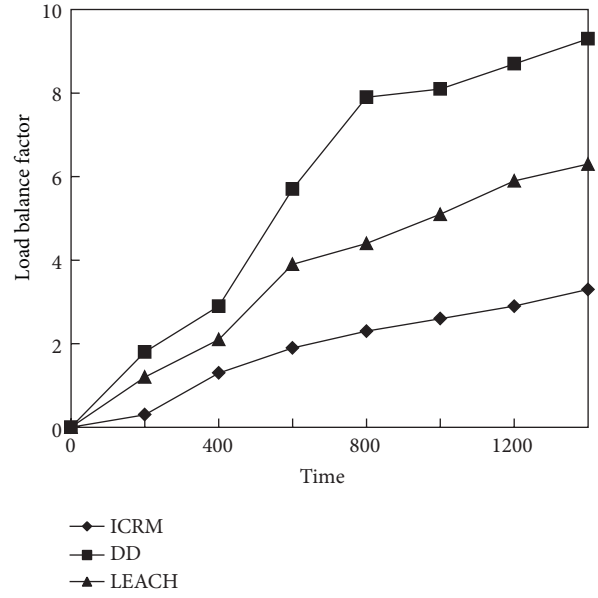


Figure 4: Load balance.

network packets loss rate is smaller, and the cluster head setup between multi-paths makes data transmission more reliable, even if a certain path fails; route maintenance mechanism to take timely measures to make the data transmission is almost unaffected, and routing robustness is better. With daemon nodes splitting the tasks of the cluster head nodes, which can reduce the energy consumption of nodes, each cluster head works longer, with more quantities of data as a whole. While, in DD agreement, most of the nodes energy consumption quickly, easy to death, makes the minimum quantities of data protocol. LEACH protocol cluster head bear there are the mission of the heavier, but member nodes energy consumption less, better than the DD agreement, but uneven distribution of cluster heads, which lead some cluster heads to premature death which affects the data transmission.

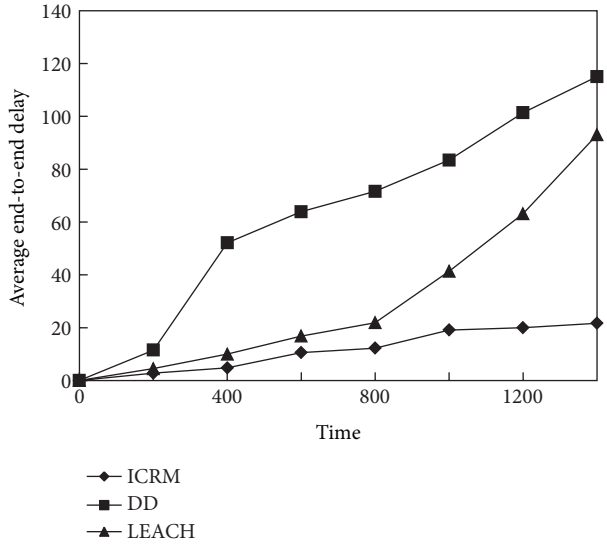


Figure 5: Average end-to-end delay.

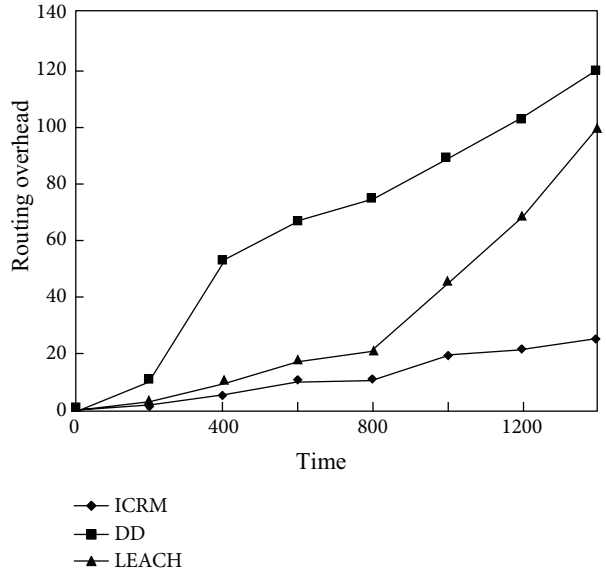


Figure 7: Routing overhead.

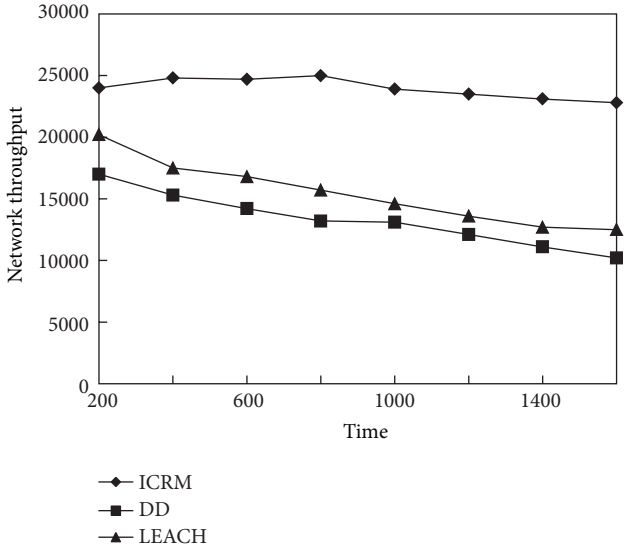


Figure 6: Network throughout.

In Figure 7, because the DD agreement path generation time is long and often interrupted, routing maintenance spends more cost, so the routing control overhead is the largest. Cluster head easy to die in the LEACH protocol and often to be cluster head selection again every time refactoring and controlling the routing overhead are large. ICRM and cluster heads are controlled by daemon node work, and cluster heads are less susceptible to attack and daemon node real-time monitoring work and timely replacement of each failure cluster heads or removal of low trust value of nodes, in each path on the energy balance, almost can be seen from the diagram, the routing mechanism overhead is relatively stable, significantly lower than the other two protocols.

7. Conclusion

This paper to solve the problem was made in the course of the LEACH protocol in data transmission security hidden danger which introduced a simple key management mechanism, the secret sharing technology, and random key to the model of a bunch of key management information into more pieces which were assigned to each sensor node, among daemon nodes, cluster heads, and member nodes which need to be the key of authentication; in a certain moment even network, one or several nodes attack; also will not leak the key information. In order to increase the credibility of the working node we also introduced the node evaluation mechanism, and low trust value node has been dropped as a network cannot be directly involved in data transmission and effectively improve the performance of network security. Data transmission among clusters enhanced multi-path mechanism, which further improves the reliability of data transmission and implements network load balancing.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants nos. 61070169, 61201212, Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, Specialized Research Foundation for the Doctoral Program of Higher Education of China no. 20103201110018 and Application Foundation Research of Suzhou of China nos. SYG201118 and SYG201240.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] Z. Bidai, H. Haffaf, and M. Maimour, "Node disjoint multipath routing for ZigBee cluster-tree wireless sensor networks,"

in *Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS '11)*, pp. 1–6, April 2011.

[3] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[4] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 174–185, 1999.

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[6] B. V. Nadimpalli, P. Mulukutla, R. Garimella, and M. B. Srinivas, "Energy-aware routing in sensor networks using dual membership clusters and data highways," in *Proceedings of the IEEE Region 10th Conference Analog and Digital Techniques in Electrical Engineering (TENCON '04)*, pp. C184–C187, November 2004.

[7] J. Z. Long, Y. T. Chen, D. M. Deng et al., "Assistant cluster head clustering algorithm based on LEACH protocol," *Computer Engineering*, vol. 37, no. 7, pp. 103–105, 2011.

[8] H. Nabizadeh and M. Abbaspour, "IFRP: an intrusion/fault tolerant routing protocol for increasing resiliency and reliability in wireless sensor networks," in *Proceedings of the International Conference on Selected Topics in Mobile and Wireless Networking (iCOST '11)*, pp. 24–29, October 2011.

[9] H. Yu, J. He, T. Zhang, and P. Xiao, "A group key distribution scheme for wireless sensor networks in the Internet of things scenario," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 813594, 12 pages, 2012.

[10] N. A. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks: a review," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 167575, 7 pages, 2013.

[11] J. Lopez, R. Roman, I. Agudo, and C. Fernandez-Gago, "Trust management systems for wireless sensor networks: best practices," *Computer Communications*, vol. 33, no. 9, pp. 1086–1093, 2010.

[12] T. Zhang, L. Li, and C. Yan, "A secure cluster-based router protocol for WSNs," *Chinese Journal of Sensors and Actuators*, vol. 22, no. 11, pp. 1612–1616, 2009.

[13] P. C. Yu, H. Z. Zhang, and Z. J. Liu, "Research of cluster-based multi-hop routing protocol," *Journey of Computer Applications*, vol. 27, no. 2, pp. 351–354, 2007.

[14] M. Liu, J.-N. Cao, G.-H. Chen, L.-J. Chen, X.-M. Wang, and H.-G. Gong, "EADEEG: an energy-aware data gathering protocol for wireless sensor networks," *Journal of Software*, vol. 18, no. 5, pp. 1092–1109, 2007.

[15] P. C. Zhao, Y. Xu, and M. Nan, "A hybrid key management scheme based on clustered wireless sensor networks," in *Proceedings of the IEEE 2nd International Conference on Computer and Management*, pp. 1394–1397, 2012.

[16] X. Sun and E. J. Coyle, "The effects of motion on distributed detection in mobile ad hoc sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 460924, 14 pages, 2012.

[17] R. Lu, X. Lin, H. Zhu, X. Liang, and X. Shen, "BECAN: a bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 32–43, 2012.

[18] A. Modirkhazeni, N. Ithnin, and O. Ibrahim, "Secure multipath routing protocols in wireless sensor networks: a security survey analysis," in *Proceedings of the 2nd International Conference on Network Applications, Protocols and Services (NETAPPS '10)*, pp. 228–233, September 2010.

[19] E. Kohno, T. Okazaki, M. Takeuchi, T. Ohta, Y. Kakuda, and M. Aida, "Improvement of assurance including security for wireless sensor networks using dispersed data transmission," *Journal of Computer and System Sciences*, vol. 78, no. 6, pp. 1703–1715, 2012.

[20] C. Chen, W. Wu, and Z. Li, "Multipath routing modeling in ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '05)*, pp. 2974–2978, May 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 147419, 6 pages
http://dx.doi.org/10.1155/2013/147419

Research Article

Multisensor Data Fusion for Water Quality Evaluation Using Dempster-Shafer Evidence Theory

Jian Zhou,[1,2] Linfeng Liu,[1,2] Jian Guo,[1,2] and Lijuan Sun[1,2]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China*

Correspondence should be addressed to Jian Zhou; zhoujian@njupt.edu.cn

Received 8 August 2013; Accepted 29 September 2013

Academic Editor: Chongsheng Zhang

A multisensor data fusion approach for water quality evaluation using Dempster-Shafer evidence theory is presented. To evaluate water quality, each sensor measurement is considered as a piece of evidence. Based on the water quality parameters measured by sensor node, the mass function of water quality class is calculated. Evidence from each sensor is given a reliability discounting and then combined with the others by D-S rule. According to the decision rule which uses the fusion mass function values, the class of water quality can be determined. Finally, experiments are given to demonstrate that the proposed approach can evaluate water quality from uncertain sensor data and improve evaluation performance.

1. Introduction

Water quality evaluation is important in providing a reliable supply of potable water. Empirical evidence shows that water quality parameters, such as dissolved oxygen (DO), NH_3-N, total phosphorus (TP), and total nitrogen (TN), are sensitive indicators of contaminants. In 2005, Hall and Szabo [1] demonstrated that changes in water quality parameters, which potentially indicate contamination, can be detected using sensors. Then, wireless sensor network (WSN) has been extensively applied in monitoring and evaluating water quality [2–4].

Multisensor data fusion is a technology to enable combining information from several sources in order to form a unified picture [5]. It is an important tool for improving the performance of monitoring system when various sensors are available. Multi-sensor data fusion seeks to combine data from multiple sensors to perform inferences that will be more efficient and potentially more accurate than if they were achieved by means of a single sensor. During recent years, the multi-sensor data fusion technique has received much attention, but it is more about applications to target identification, signal and image processing, and biomedical engineering [6–9]. In this paper, we apply multi-sensor data fusion technology in water quality evaluation.

The main problem is that data obtained from sensors have different degrees of uncertainty [10]. This uncertainty may arise for a number of reasons. For instance, the sensing error increases with the age of sensor, and the sensor is disturbed by environment. The quality of the wireless links is another major limiting factor. Furthermore, this uncertainty may lead to conflicting conclusion. Since data obtained from the sensors is inherently incomplete, uncertain, and imprecise, it is imperative that a fusion mechanism be devised so as to minimize such imprecision and uncertainty.

Dempster-Shafer evidence theory (D-S evidence theory) and Bayesian methods are commonly used to handle uncertainty. The basic strategy of Bayesian methods is that if the prior probabilities and conditional probabilities are determined in advance, then the posteriori probabilities can be estimated using Bayes formula. Examples of applying Bayesian methods for multi-sensor data fusion can be found in [11–13]. Nonetheless, effective fusion performance can be achieved only if adequate and appropriate a priori and conditional probabilities are available. In some situations,

assumptions can be made with respect to a priori and conditional probabilities, but these assumptions can turn to be unreasonable in many other situations. D-S evidence theory can be regarded as an extension of classical probabilistic reasoning, which makes inferences from incomplete and uncertain data provided by different independent sources. The application of D-S evidence theory in multi-sensor data fusion can be found in [14–17]. A key advantage of D-S evidence theory is its ability to deal with uncertain data without adequate priori probabilities.

This paper introduces a novel multi-sensor data fusion approach for water quality evaluation using D-S evidence theory. We view each sensor measurement as a piece of evidence that reveals some uncertain information about the water quality. And we get water quality evaluation through the fusion of uncertain data from each sensor. The rest of this paper is organized as follows. In Section 2, we introduce some preliminary concepts of the evidence theory. We present our multi-sensor data fusion approach for water quality evaluation using D-S evidence theory in terms of mass function based on water quality parameters, reliability discounting, and decision rule using the fusion mass function values. Section 3 describes experiments in which we demonstrate the effectiveness of the proposed approach. Section 4 provides some concluding remarks.

2. Water Quality Evaluation Using D-S Evidence Theory

2.1. D-S Evidence Theory. The D-S evidence theory originated from Dempster's work [18] and is further extended by Shafer [19] is a generalisation of traditional probability which allows us to better quantify uncertainty. The theory is based on a number of key propositions which are summarized as follows.

(1) *Frame of discernment*: let Θ be a finite set of elements; an element can be a hypothesis, an object, or in our case a water quality evaluation. We refer to Θ as the frame of discernment; the set consisting of all the subsets of Θ is called the power set of Θ and denoted by $\Omega(\Theta)$.

(2) *Mass function*: mass function m is also called a basic probability assignment function. It is defined as a mapping of the power set $\Omega(\Theta)$ to a number between 0 and 1as follows:

$$m : \Omega(\Theta) \longrightarrow [0,1]$$
$$m(\Phi) = 0, \qquad \sum_{A \subseteq \Omega(\Theta)} m(A) = 1, \tag{1}$$

where $m(A)$ is an expression of the level of confidence exactly in A. It does not include the confidence in any particular subset of A. In water quality evaluation, $m(A)$ can be considered as a degree of belief held by certain class of water quality. If $m(A) > 0$, the subset A of Θ is called a focal element. When a mass value is committed to a subset that has more than one element, it is explicitly stating that there is not enough information to distribute this belief more precisely to each individual element in the subset. In particular, the total belief is assigned to the whole frame of discernment; $m(\Theta) = 1$ when there is no evidence about Θ at all; $m(\Theta)$ is the uncertain function.

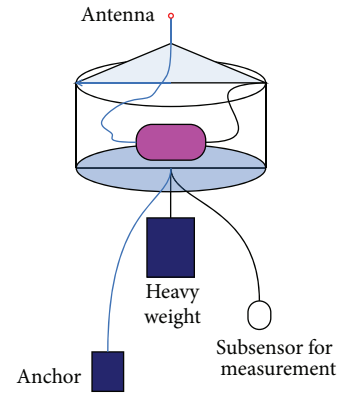


Figure 1: Planktonic sensor node model.

(3) *Dempster's rule of combination (D-S rule)*: suppose m_1 and m_2 are two mass functions formed based on data obtained from two different and independent sources in the same frame of discernment Θ. We can get a new m according to D-S rule as follows:

$$m(A) = m_1 \oplus m_2 = \begin{cases} 0 & A = \Phi, \\ \dfrac{\sum_{B \cap C = A} m_1(B)\, m_2(C)}{1 - K} & A \neq \Phi, \end{cases} \tag{2}$$

where

$$K = \sum_{B \cap C = \Phi} m_1(B)\, m_2(C). \tag{3}$$

$K \in (0,1)$ is a normalization constant and can be viewed as a measure of conflict among the sources of evidence. The higher the K is, the more conflicting are the sources. Dempster's rule of combination can blend measures of evidence from different sources.

2.2. Water Quality Evaluation. This section focuses on D-S evidence theory applications in multi-sensor environment and presents our implementation for water quality evaluation.

Consider a wireless sensor network with planktonic sensor node as shown in Figure 1, used to monitor and evaluate the water quality in a measurement area, for example, a lake or a pool. Each sensor node has several subsensors used to measure the water quality parameters. Let $F = \{f_1, f_2 \cdots f_m\}$ denote the water quality parameters measured by each sensor node, depending on the application; f may represent quality parameter such as DO, NH_3-N; m is the number of water quality parameters. Through the water quality parameters, we can determine the class of water quality.

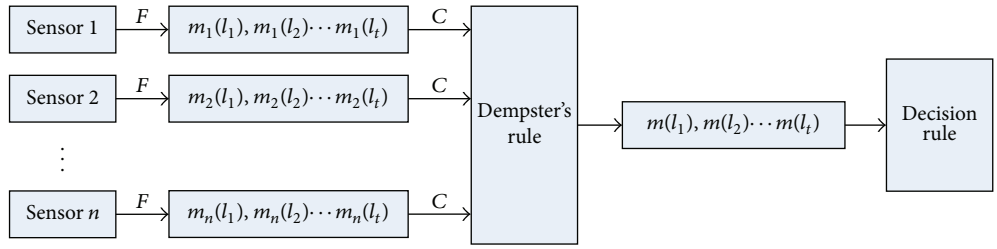


Figure 2: Water quality evaluation using D-S evidence theory.

Let $\Theta = \{l_1, l_2 \cdots l_i \cdots l_t\}$ be the frame of discernment for water quality evaluation; l_i means that the water quality is classified to the l_i class; t is the number of water quality classes.

Assume N sensor nodes, for the sake of simplicity, and suppose that all sensor nodes are independent. Each sensor node measures water quality parameters (F) of measurement area. When applying the D-S evidential theory to multi-sensor data fusion, data obtained from each sensor node is the theory's evidence. Figure 2 shows the block diagram of our multi-sensor data fusion approach for water quality evaluation using D-S evidence theory.

According to the D-S evidence theory, for each sensor node, the possibility of water quality class can be described by mass function values. $m_n(l_1), m_n(l_2), \ldots, m_n(l_t)$ are mass function values obtained by F from sensor n. $m_n(l_t)$ means the confidence assigned to the l_t class of water quality provided by sensor n. Multi-sensor data fusion amounts to combine several lines of evidence to form a new comprehensive evidence. For uncertainty, each sensor node is then given a reliability discounting (C) before combination. By using the fusion mass function values obtained by D-S rule, the class of water quality can be determined.

2.2.1. Mass Function Based on Water Quality Parameters. The derivation of mass function is the most crucial step in D-S evidence theory, because it determines the reliability of conclusions. In our approach, the calculation of the mass function is based on the water quality parameters provided by sensor node.

Let S_k represent the measurement of water quality parameters obtained from sensor k:

$$S_k = \{s_1, s_2 \cdots s_i \cdots s_m\}, \tag{4}$$

where s_i is the ith element of water quality parameters.

Let F_Θ represent the features of all water quality classes:

$$F_\Theta = \begin{Bmatrix} F^{l_1} : \{f_1^{l_1}, f_2^{l_1} \cdots f_m^{l_1}\} \\ F^{l_2} : \{f_1^{l_2}, f_2^{l_2} \cdots f_m^{l_2}\} \\ \vdots \\ F^{l_i} : \{f_1^{l_i}, f_2^{l_i} \cdots f_m^{l_i}\} \\ \vdots \\ F^{l_t} : \{f_1^{l_t}, f_2^{l_t} \cdots f_m^{l_t}\} \end{Bmatrix}, \tag{5}$$

where F^{l_i} describes the feature quality parameters of the l_i class.

Intuitively, the more similar is S_k to F^{l_i}, the more probable is the l_i class of water quality, as far as sensor k is concerned. Conversely, the more dissimilar is S_k to F^{l_i}, the less probable is the l_i class of water quality, again as far as sensor k is concerned.

There are many measures for quantifying the distance between the measured parameters and the feature of water quality class. We propose to use the Minkowski distance measure, as follows:

$$d_k^{li} = \left(\sum_{x=1}^{m} \left(\frac{s_x - f_x^{li}}{f_{\max} - f_{\min}} \right)^{\alpha} \right)^{1/\alpha}, \tag{6}$$

where d_k^{li} is the distance between S_k and F^{l_i}; α is a constant; dividsion by $f_{\max} - f_{\min}$ is for normalizing. The distances between measurement of sensor k and the features of all water quality classes can be captured as follows:

$$D_k = \left\{ d_k^{l1}, d_k^{l2} \cdots d_k^{li} \cdots d_k^{lt} \right\}. \tag{7}$$

The smaller the distance d_k^{li} is, the more probable the l_i class of water quality is. Defining

$$m_k(l_i) = \frac{\left(1/d_k^{li}\right)}{\sum_{i=1}^{t} \left(1/d_k^{li}\right)}, \tag{8}$$

then we have the mass function of water quality class from sensor k:

$$m_k = \{m_k(l_1), m_k(l_2) \cdots m_k(l_i) \cdots m_k(l_t)\} \tag{9}$$

where $m_k(l_i)$ is the mass function assigned by sensor k to the l_i class of water quality.

2.2.2. Reliability Discounting. Some sensor nodes are more vulnerable to misreading or malfunctioning due to many factors, such as their age, their type, and their location. The impact of evidence is discounted to reflect the sensor node's reliability, in terms of reliability discounting C ($0 < C < 1$). For uncertainty, evidence from each sensor node is then given a reliability discounting as follows:

$$C_k = \begin{cases} \beta & N_k \le N_{\min} \\ \min\left(\dfrac{R_k}{N_k}, \beta \right) & N_k > N_{\min,} \end{cases} \tag{10}$$

where N_k is the total number of water quality evaluation from sensor k; R_k is the number of correct water quality evaluation from sensor k; β and $N_{\min}$ are fixed values. If the number of water quality evaluation is less than $N_{\min}$, we use a fixed value β as reliability discounting. β and $N_{\min}$ are usually selected as 0.9, 10, respectively.

If its reliability discounting (C) is high, this evidence will be given more weight and have greater effect on the modified combinatorial rule. Then, the mass function m_k in (9) is updated to m_k':

$$m_k' = \begin{cases} m_k'(l_i) = C_k \cdot m_k(l_i) \\ m_k'(\Theta) = 1 - \sum_{i=1}^{t} m_k'(l_i), \end{cases} \tag{11}$$

where $m_k'(l_i)$ is the modified mass function assigned by sensor k to the l_i class of water quality; $m_k'(\Theta)$ is the uncertain function (unidentified mass function) from sensor k.

2.2.3. Decision Rule Using the Fusion Mass Function Values. As stated earlier, all sensors are assumed to be independent. The multi-evidence combinatorial rule becomes

$$m(A) = m_1' \oplus m_2' \oplus \cdots \oplus m_k' \cdots \oplus m_n' = \begin{cases} 0 & A = \Phi \\ \dfrac{\sum_{\cap_{k=1}^{n} A_k = A} m_1'(A_1) m_2'(A_2) \cdots m_k'(A_k) \cdots m_n'(A_n)}{1 - K'} & A \neq \Phi, \end{cases} \tag{12}$$

where

$$K' = \sum_{\cap_{k=1}^{n} A_k = \Phi} m_1'(A_1) m_2'(A_2) \cdots m_k'(A_k) \cdots m_n'(A_n). \tag{13}$$

Using combinatorial rule, a fusion mass function ($m(A)$) converts the confidence of each class of water quality arising from different evidence sources (sensor nodes) into a fraction in $(0, 1)$. By using the fusion mass function values, water quality can be evaluated on the decision rule as follows.

(1) The current determined class of water quality should have a maximal mass function value and should be greater than a certain value; this value should be at least greater than $1/t$; t stands for the number of water quality classes.

(2) The difference of the mass function values between the current determined class of water quality and other classes should be greater than a certain gate limit value, and here it is 0.2.

(3) The uncertain function value should be less than a certain gate limit value, and here it is 0.1.

If the three rules above are not satisfied simultaneously, the current class of water quality is uncertain.

Figure 3: Sensor nodes used in experiments.

Table 1: Features of all water quality classes.

	DO	NH_3-N	TP	TN
I	7.5	0.15	0.01	0.2
II	6.0	0.5	0.025	0.5
III	5.0	1.0	0.05	1.0
IV	3.0	1.5	0.1	1.5
V	2.0	2.0	0.2	2.0

3. Experiments and Results

In this section, we give two experiments to validate the performance of the proposed approach. In our experiments, we use the senor nodes developed by ourselves to measure water-quality parameters, as shown in Figure 3. The measured parameters include DO, NH_3-N, TP, and TN; then F = {DO, NH_3-N, TP, TN}.

According to "Environmental Quality Standards for Surface Water of China GB 3838-2002", water quality is categorized as 5 classes, and the features of all water quality classes are shown in Table 1. Thus, in our experiments, the frame of discernment for water quality evaluation is Θ = {l_1 = "I", l_2 = "II", l_3 = "III", l_4 = "IV", l_5 = "V"}, where "I" means that the water quality is classified to the first class. From Table 1, (5) can be expressed as follows:

$$F_\Theta = \left\{ \begin{array}{l} F^{l_1} : \{7.5, 0.15, 0.01, 0.2\} \\ F^{l_2} : \{6.0, 0.5, 0.025, 0.5\} \\ F^{l_3} : \{5.0, 1.0, 0.05, 1.0\} \\ F^{l_4} : \{3.0, 1.5, 0.1, 1.5\} \\ F^{l_5} : \{2.0, 2.0, 0.2, 2.0\} \end{array} \right\}. \tag{14}$$

In the first experiment, we use three sensor nodes (S_1, S_2, and S_3) to monitor the water quality of a pool. The objective is to determine the water quality class of the pool based on the three sensor nodes. Table 2 is assumed to list the water quality parameters measured by sensors S_1, S_2, and S_3.

Table 2: Water quality parameters measured by sensors S_1, S_2, and S_3.

	DO	NH_3-N	TP	TN
S_1	6.10	2.550	0.122	3.432
S_2	6.65	0.660	0.038	0.706
S_3	7.21	0.467	0.022	0.690

Table 3: Mass functions from S_1, S_2, and S_3.

	I	II	III	IV	V
m_1	0.135	0.154	0.191	0.241	0.279
m_2	0.185	0.438	0.232	0.091	0.054
m_3	0.265	0.422	0.177	0.084	0.052

From (6), when $\alpha = 2$, $f_{\max} = \max(f_i)$ and $f_{\min} = 0$, we can calculate that

$$d_1^{l1} = \left(\left(\frac{6.1-7.5}{7.5}\right)^2 + \left(\frac{2.55-0.15}{2}\right)^2 + \left(\frac{0.122-0.01}{0.2}\right)^2 + \left(\frac{3.432-0.2}{2}\right)^2\right)^{1/2} = 2.0976,$$

$$d_1^{l2} = \left(\left(\frac{6.1-6.0}{7.5}\right)^2 + \left(\frac{2.55-0.5}{2}\right)^2 + \left(\frac{0.122-0.025}{0.2}\right)^2 + \left(\frac{3.432-0.5}{2}\right)^2\right)^{1/2} = 1.8534,$$

$$d_1^{l3} = \left(\left(\frac{6.1-5.0}{7.5}\right)^2 + \left(\frac{2.55-1.0}{2}\right)^2 + \left(\frac{0.122-0.05}{0.2}\right)^2 + \left(\frac{3.432-1.0}{2}\right)^2\right)^{1/2} = 1.4935,$$

$$d_1^{l4} = \left(\left(\frac{6.1-3.0}{7.5}\right)^2 + \left(\frac{2.55-1.5}{2}\right)^2 + \left(\frac{0.122-0.1}{0.2}\right)^2 + \left(\frac{3.432-1.5}{2}\right)^2\right)^{1/2} = 1.1797,$$

$$d_1^{l5} = \left(\left(\frac{6.1-2.0}{7.5}\right)^2 + \left(\frac{2.55-2.0}{2}\right)^2 + \left(\frac{0.122-0.2}{0.2}\right)^2 + \left(\frac{3.432-2.0}{2}\right)^2\right)^{1/2} = 1.0194. \quad (15)$$

From (8), we can get the mass function of water quality class from S_1: $m_1 = \{0.135, 0.154, 0.191, 0.241, 0.279\}$. Similarly, we can also calculate mass functions from S_2 and S_3, as shown in Table 3.

Suppose that the numbers of water quality evaluation from sensors S_1, S_2, and S_3 are the same 15, and the numbers of correct evaluation are, respectively, 9, 12, and 14. From (10) and (11), we can get the modified mass functions and the uncertain functions from S_1, S_2, and S_3, as shown in Table 4.

Table 4: Modified mass functions and uncertain functions from S_1, S_2, and S_3.

	I	II	III	IV	V	Θ
m_1'	0.081	0.092	0.114	0.145	0.168	0.4
m_2'	0.148	0.350	0.186	0.073	0.043	0.2
m_3'	0.238	0.380	0.159	0.076	0.047	0.1

Table 5: Fusion mass functions and uncertain functions.

	I	II	III	IV	V	Θ
$m_1' \oplus m_2'$	0.145	0.316	0.196	0.114	0.096	0.133
$m_1' \oplus m_2' \oplus m_3'$	0.193	0.483	0.172	0.072	0.048	0.032

We can see from Table 2 that the water quality parameters measured by S_1 are obviously different from S_2 and S_3. S_1 gives different conclusion. For uncertainty, the data provided by S_1 may be incorrect. Only consider the data of S_1 and S_2 from Table 4; two pieces of evidence are in conflict. According to multievidence combinatorial rule, which is realized by (12) and (13), we can combine the evidence provided by S_1 and S_2, as shown in the first row of Table 5. The second class of water quality has the maximal mass function value. It is seen that our approach is effective when there is conflict of evidence. However, according to our decision rule, the uncertain function value is great and we cannot determine the class of current water quality.

Then, consider the data of S_3; the fusion mass function and uncertain function combining S_1, S_2, and S_3 are also shown in Table 5. We reach the water quality evaluation result in favor of "II" with a degree of belief of 0.483. And the uncertain function value is reduced to 0.032. According to the decision rule, we can determine that the water quality of the pool is the second class. This result is the combination of the three sensor nodes and is considered to be more believable compared to that of the single sensor node.

We can see clearly from Table 5 that by comparing the fusion mass function value with the single sensor mass function value, the mass function value of the current determined water quality class is enlarged, while the difference of the mass function values between the current determined class of water quality and other classes is enlarged, and at the same time, the uncertain function value is reduced.

In the second experiment, we use different numbers of sensor nodes to monitor the water quality of a small lake at different time and different locations and use the approach of this paper to evaluate the water quality. Here we already know that water quality of this lake is the fourth class. The objective is to demonstrate that our approach could improve the accuracy of water quality evaluation. According to simulation, Table 6 shows the correct rate and uncertain rate compared between different numbers of sensor nodes used in multi-sensor data fusion for water quality evaluation.

From Table 6, it can be seen that our approach can increase correct rate of water quality evaluation, compared to the approach using single sensor node. It can also be seen that the fusion can reduce the uncertain rate of water quality evaluation.

Table 6: Correct rate and uncertain rate with different sensor nodes combination.

	Single sensor node	Three sensor nodes	Five sensor nodes
Correct rate (%)	38	84	96
Uncertain rate (%)	46	10	2

In summary, our experiments have indicated that the proposed multi-sensor data fusion approach for water quality evaluation using D-S evidence theory has improved water quality evaluation performance greatly.

4. Conclusions

We have applied multi-sensor data fusion technology in water quality evaluation in this paper. We have introduced a novel multi-sensor data fusion approach for water quality evaluation using D-S evidence theory. Within our work, we have proposed a method of calculating mass function based on water quality parameters and proposed a reliability discounting to reflect the sensor node's reliability. Furthermore, we have proposed the decision rule to determine the class of water quality by using the fusion mass function values. Our experiments have indicated that the proposed approach can evaluate water quality from uncertain sensor data, increase correct rate, reduce uncertain rate of water quality evaluation, compared to the approaches using individual sensors.

Acknowledgments

This work is sponsored by the National Natural Science Foundation of China (no. 71301081), Natural Science Foundation of Jiangsu Province (no. BK20130877, BK2012833), the Natural Science Foundation of Jiangsu Higher Education (no. 12KJB520011), and the Scientific Research Foundation of Nanjing University of Posts and Telecommunications (no. NY213035).

References

[1] J. Hall and J. Szabo, *WaterSentinel Online Water Quality Monitoring as an Indicator of Drinking Water Contamination*, Environmental Protection Agency, 2005.

[2] M. W. Koch and S. A. McKenna, "Distributed sensor fusion in water quality event detection," *Journal of Water Resources Planning and Management*, vol. 137, no. 1, pp. 10–19, 2011.

[3] I. Stoianov, L. Nachman, A. Whittle, S. Madden, and R. Kling, "Sensor networks for monitoring water supply and sewer systems: lessons from Boston," in *Proceedings of the 8th Annual Water Distribution Systems Analysis Symposium*, pp. 1–17, August 2006.

[4] E. Karami, F. M. Bui, and H. H. Nguyen, "Multisensor data fusion for water quality monitoring using wireless sensor networks," in *Proceedings of the International Conference on Communications and Electronics (ICCE '12)*, pp. 80–85, August 2012.

[5] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavib, "Multisensor data fusion: a review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[6] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 12, pp. 1696–1710, 2006.

[7] I. Corona, G. Giacinto, C. Mazzariello, F. Roli, and C. Sansone, "Information fusion for computer security: state of the art and open issues," *Information Fusion*, vol. 10, no. 4, pp. 274–284, 2009.

[8] A. Ardeshir Goshtasby and S. Nikolov, "Image fusion: advances in the state of the art," *Information Fusion*, vol. 8, no. 2, pp. 114–118, 2007.

[9] A. Ross and A. Jain, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2115–2125, 2003.

[10] A. Ranganathan, J. Al-Muhtadi, and R. H. Campbell, "Reasoning about uncertain contexts in pervasive computing environments," *IEEE Pervasive Computing*, vol. 3, no. 2, pp. 62–70, 2004.

[11] T. Bréhard and V. Krishnamurthy, "Optimal data incest removal in Bayesian decentralized estimation over a sensor network," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, pp. III173–III176, April 2007.

[12] D. H. Wilson and C. Atkeson, "Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors," in *Proceedings of the 3rd International Conference on Pervasive Computing*, pp. 62–79, May 2005.

[13] S. Maskell, "A Bayesian approach to fusing uncertain, imprecise and conflicting information," *Information Fusion*, vol. 9, no. 2, pp. 259–277, 2008.

[14] O. Basir and X. Yuan, "Engine fault diagnosis based on multi-sensor information fusion using Dempster-Shafer evidence theory," *Information Fusion*, vol. 8, no. 4, pp. 379–386, 2007.

[15] D. Zhu and W. Gu, "Sensor fusion in integrated circuit fault diagnosis using a belief function model," *International Journal of Distributed Sensor Networks*, vol. 4, no. 3, pp. 247–261, 2008.

[16] J. M. Li, S. H. Luo, and J. S. Jin, "Sensor data fusion for accurate cloud presence prediction using Dempster-Shafer evidence theory," *Sensors*, vol. 10, no. 10, pp. 9384–9396, 2010.

[17] D. B. Hou, H. M. He, P. J. Huang, G. X. Zhang, and H. Loaiciga, "Detection of water-quality contamination events based on multi-sensor fusion using an extented Dempster-Shafer method," *Measurement Science and Technology*, vol. 24, no. 5, pp. 1–18, 2013.

[18] A. P. Dempster, "A generalization of bayesian inference," *Journal of the Royal Statistical Society B*, vol. 30, no. 2, pp. 205–247, 1968.

[19] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 926830, 7 pages
http://dx.doi.org/10.1155/2013/926830

Research Article

A New Scheme Based on Pixel-Intense Motion Block Algorithm for Residual Distributed Video Coding

Lijuan Sun,[1,2,3] Zhaoxiao Lin,[1] Fu Xiao,[1,2,3] and Jian Guo[1,2,3]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China*
[3] *Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education, Nanjing, Jiangsu 210003, China*

Correspondence should be addressed to Lijuan Sun; sunlj@njupt.edu.cn

Received 11 July 2013; Accepted 1 October 2013

Academic Editor: Chongsheng Zhang

Residual Distributed Video Coding (RDVC) is a branch of the Distributed Video Coding (DVC). By reducing the coding rate, the RDVC scheme gets a better rate-distortion performance. The overall performance of the traditional pixel-based RDVC scheme is better than the traditional transform-based domain DVC scheme. But the traditional RDVC scheme also has some application limitations; for example, the performance of RDVC is bad when it is applied to intense motion regions. And the pixel-based RDVC scheme cannot fully exploit the spatial correlations of the original information. In this paper, we propose a new approach of residual coding combined IMB (intense motion block) extraction algorithm with Low Density Parity Check (LDPC) and Baum-Welch iterative decoding algorithm. Experimental results show that the proposed scheme achieves better rate-distortion performance compared to traditional DVC and RDVC scheme.

1. Introduction

With the development of wireless multimedia communication technology, the new video processing and transmission demand began to appear: such as hand-held digital camera, low-power video sensor networks, PAD, and other video equipments. Typically, this new device has the features of small battery capacity and limited computing power, so it requires a low complexity coding scheme. In conventional video coding standards, such as the ISO/IEC MPEG-x and ITU-T H.26x, the encoding complexity is 5 to 10 times [1] the decoding. Obviously, traditional coding schemes are not suit for the new video applications. So a special video coding framework which is called Distributed Video Coding (DVC) [2, 3] gets more attention from scholars.

The DVC scheme is different from the traditional video coding scheme. DVC scheme takes the intraframe coding and interframe decoding techniques and discovers the correlation of the video signal at the decoder; thus, the DVC scheme transfers the complexity from the encoding side to the decoding side. The DVC scheme has the characteristics of low complexity encoding and good robustness which can meet the new demand for video applications.

The Residual Distributed Video Coding (RDVC) [4] system is another research direction of DVC. In RDVC scheme, some of the similarities of the frame with respect to the known reference are utilized at the encoder, and the conditional decoding with respect to a better side information frame is still benefited at the decoder. Compared to the traditional Wyner-Ziv coding scheme, the residual scheme can reduce the information entropy of the encoding end and obtain a higher coding efficiency. But the scheme also has some limitations. First, the RDVC scheme shows poor performance when it is used to the regions of a frame with intense motion [5]. In addition, compared to the scheme based on DCT domain, the RDVC scheme which is based on pixel domain lacks attention on the exploration of spatial information, and the overall performance of RDVC based on pixel domain is not better than the DCT-DVC scheme [4]. For the above two points, we propose a new RDVC coding

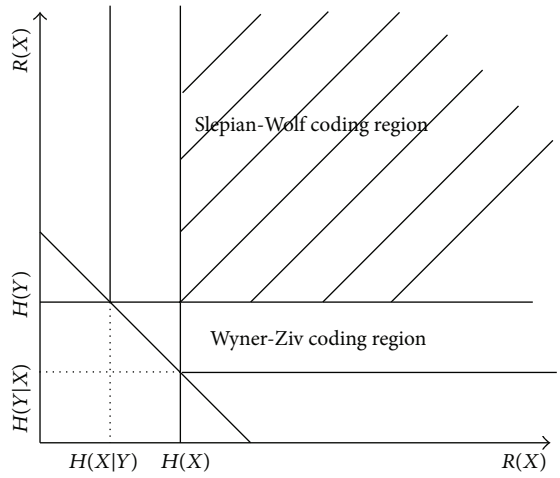


Figure 1: Diagram of Slepian-Wolf and Wyner-Ziv theorem.

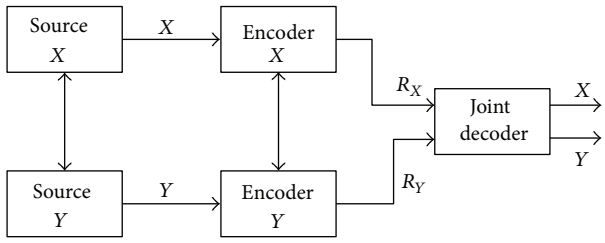


Figure 2: Traditional video coding scheme.

scheme based on pixel domain. At the encoding end, we use the intense motion block (IMB) algorithm to distinguish the intense motion blocks (IMBs) and non-IMBs, and at decoding end, we use the method of LDPC-Baum-Welch iterative decoding algorithm [6] which can explore the spatial correlation of the original information.

The remainder of this paper is organized as follows. Section 2 gives the traditional Distributed Video Coding system and its principle. Section 3 gives some details of DVC schemes and focuses on RDVC scheme. Section 4 details the proposed new scheme of the Residual Distributed Video Coding. At last, the paper presents the results of experiment and does an analysis.

2. Theoretical Basis of Distributed Video Coding

The theoretical basis of Distributed Video Coding is the Slepian-Wolf theorem [8] and Wyner-Ziv theorem [9]. The Slepian-Wolf theorem can be simply described: source $\{(X_i, Y_i)\}_{i=1}^{\infty}$ is an independent and identically distributed video sequence. Entropies of Source X and Y are $H(X)$ and $H(Y)$. Output streams of encoder X and Y are $R(X)$ and $R(Y)$. The decoding end uses the joint decoding method and gets

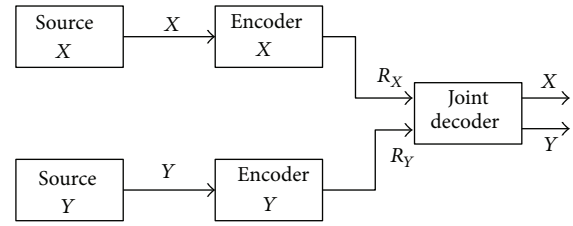


Figure 3: Distributed compression of two statistically dependent random processes, X and Y, the decoder jointly decodes X and Y and thus may exploit their mutual dependence.

the decoded information. If you want lossless recovery source X and Y, (1) and (2) are needed

$$R(X) + R(Y) \geq H(X, Y) \tag{1}$$

$$R(X) \geq H(X \mid Y), \qquad R(Y) \geq H(Y \mid X). \tag{2}$$

Figure 1 is the diagram of Slepian-Wolf theorem and Wyner-Ziv theorem, it shows the regions that meet the requirement of (1) and (2).

Slepian-Wolf theorem and Wyner-Ziv theorem proved that the method which uses independent coding and joint decoding can get the same performance with the traditional joint coding and decoding method. Figures 2 and 3 are the schematic diagram of traditional video coding system and Distributed Video Coding system.

3. Typical Distributed Video Coding

3.1. Distributed Video Coding Scheme Based on DCT Domain. The traditional DVC based on pixel domain ignores the exploration of spatial correlations between the original video information. In order to achieve higher compression efficiency, Girod et al. proposed a new DVC system based on Discrete Cosine Transform (DCT) domain [3]. By using the method of DCT, the program fully explored the spatial correlation of the source information. Figure 4 is the system diagram of DCT-based DVC scheme.

By exploring the spatial correlation of original information, the DCT scheme based on DCT domain obtain a good performance of compression efficiency. But the encoder of the DCT-based DVC scheme is complicated than the traditional pixel-based DVC scheme and demands more storage and higher computing power.

3.2. Residual Distributed Video Coding Scheme Based on Pixel Domain. Compared with traditional video coding system, the overall performance of the Distributed Video Coding system is to be further improved. According to information theory, if only the decoding side can get the side information, the coding efficiency of the system is low; that is, if the encoding side can obtain the part of the relevant information, the overall system performance of the Distributed Video Coding can be better. In order to further reduce the entropy of the input source, improve the compression of the video sequence, and ultimately obtain the improvements of the

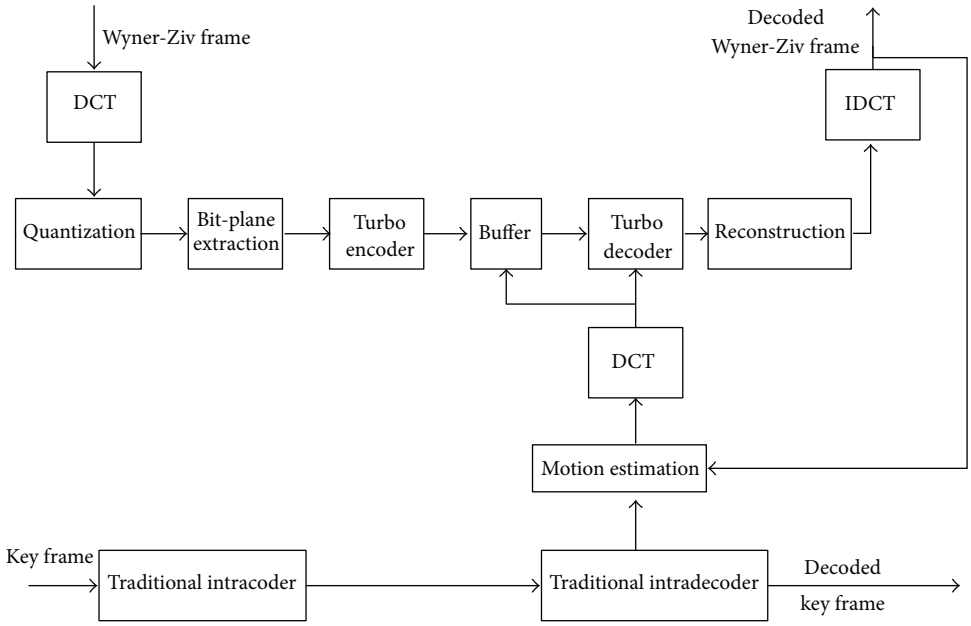


Figure 4: Traditional DCT-based DVC scheme [7].

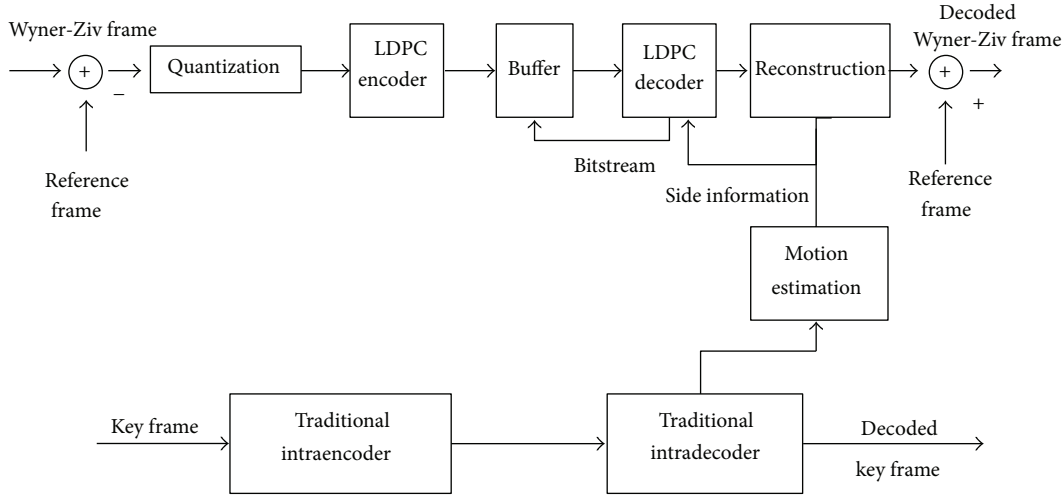


Figure 5: Pixel-based RDVC scheme [4].

system rate-distortion performance, Aaron et al. proposed the Residual Distributed Video Coding system [4], as shown in Figure 5.

The implementation steps of the RDVC scheme are as follows. In order to reduce the input information entropy and reduce the bit rate of the encoding end, the RDVC scheme makes the Wyner-Ziv frame subtract the reference frame in pixel domain. And then, quantifying and encoding the residual frame are done. The decoder makes the fusion of the residual frame and the reference frame to reconstruct the original frame. Compared to the original Wyner-Ziv frame, the information of the residual frame is smaller, and the overall performance is better than the traditional Distributed Video Coding based on pixel domain.

But the residual scheme has an application limitation: applying the RDVC method to the intense motion blocks of an image frame may cause excessive distortion of the decoded image. In order to overcome the defect, some scholars have proposed some methods [5]. For us, to solve the problem, we proposed a method using the IMB extraction algorithm at the encoder. To avoid the limitations of the residual coding, different blocks are encoded with different coding method in our RDVC scheme. In addition, considering that the traditional RDVC scheme failed to fully exploit the spatial correlation of the original information and the performance is not better than the traditional DCT-based DVC scheme, we applied the LDPC-Baum-Welch iteration decoding algorithm into our Residual Distributed Video Coding scheme. The new

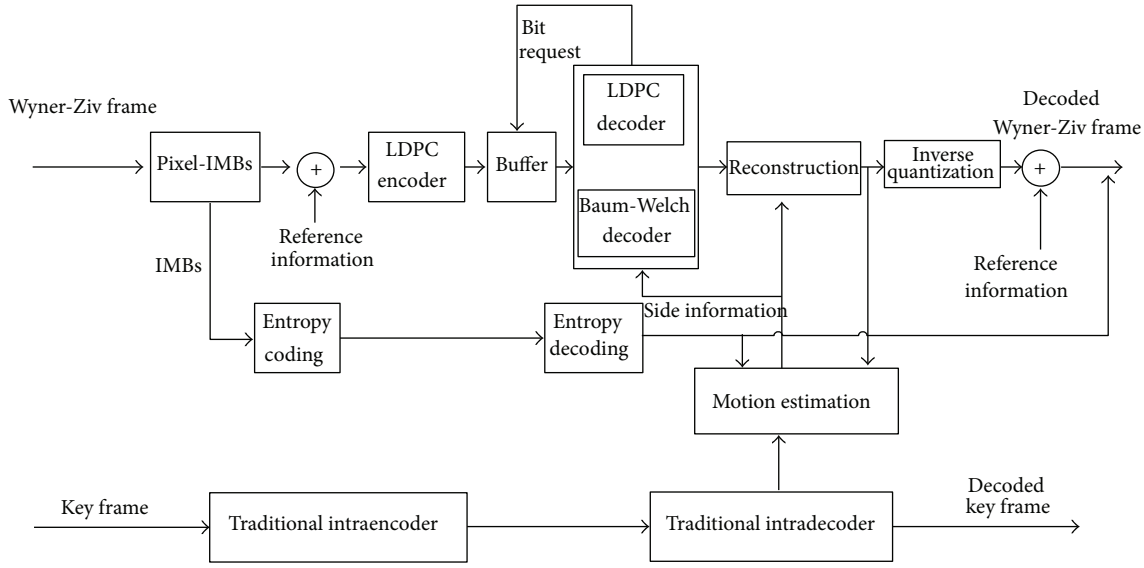


Figure 6: Proposed RDVC scheme based on Pixel-IMB algorithm.

proposed RDVC scheme of this paper effectively played the advantages of the RDVC, avoided the application limitations, minimized the complexity of the encoder, and improved the system performance.

4. Proposed RDVC Scheme Based on Pixel-IMB Algorithm

4.1. Framework of Proposed RDVC Scheme. As discussed in Section 3, we proposed a new RDVC scheme which uses IMB extraction algorithm at the encoding end and uses the LDPC-Baum-Welch algorithm at the decoding end. Figure 6 shows the general framework for the proposed RDVC scheme.

4.2. Encoder Design of the Proposed RDVC Scheme. As shown in Figure 6, at the encoder end, a subset of frames, regularly spaced in the sequence, serve as key frames, K, which are encoded and decoded using a conventional intraframe 8 × 8 Discrete Cosine Transform (DCT) codec, such as JPEG codec.

The frames between the key frames are "Wyner-Ziv frames" and encoded according to the IMB discrimination algorithm.

According to (3), we calculate the value of SAD where B_i represents a block of the frame, (i, j) denotes a pixel location, $X_n(i, j)$ represents the pixel value of the current frame of current position, and the $X_r(i, j)$ represents the pixel value of the corresponding location in the reference frame. The reference frame is the previous frame. If SAD $>$ T, we determined the block as an IMB, else the block is determined as non-IMB. T is a threshold which is from the experimental summary. Consider the following:

$$\mathrm{SAD} = \sum_{(i,j)\in B_i} \left| X_n(i,j) - X_r(i,j) \right|. \tag{3}$$

The intense motion blocks are encoded with residual distributed coding method, while the non-IMBs are encoded with entropy coding method. The residual information is from the difference of the intense motion block and the reference block.

4.3. Decoder Design of the Proposed RDVC Scheme. At the decoder, we take the LDPC-Baum-Welch (LDPC and Baum-Welch) iterative decoding scheme. The Baum-Welch decoding algorithm is based on the theory of Markov model. We look at the video image as the relevant source and use the Baum-Welch algorithm to estimate the transition probabilities of the Markov model. Thus, we can explore the spatial correlation among the source and achieve the compression.

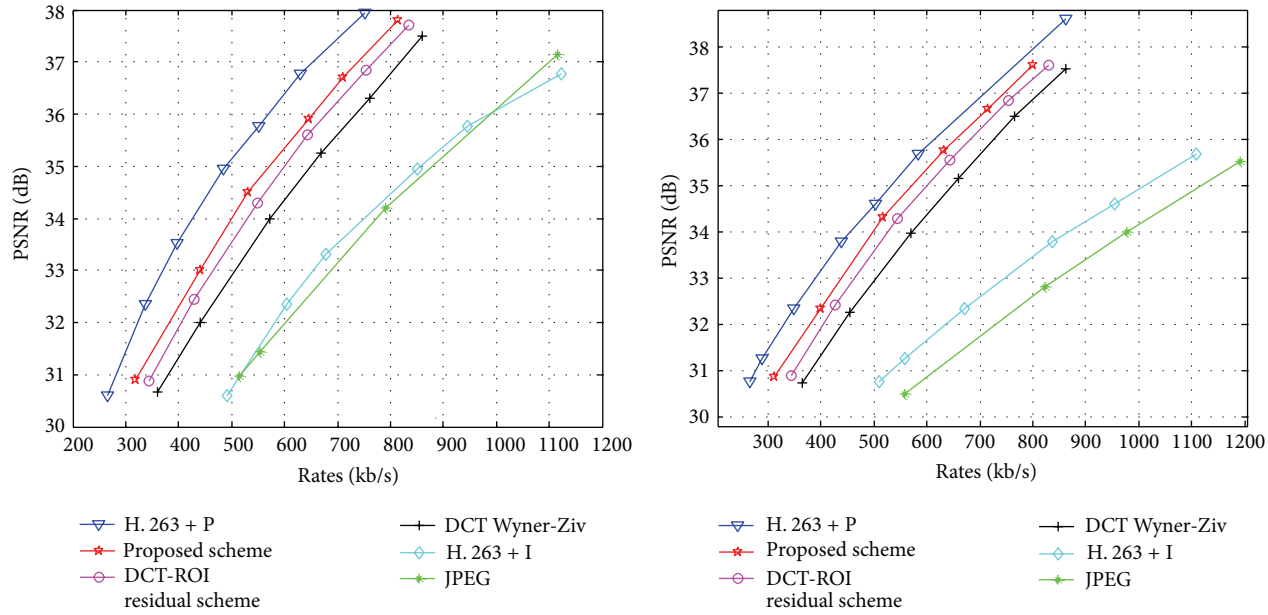
The detailed process is as follows.

Provided the source X is a Markov model. The Baum-Welch decoder accepted the coded residual information $X_{\mathrm{res}(i)}$ from LDPC decoder and the side residual information $Y - X_{\mathrm{ref}(i)}$ from entropy decoder and traditional intradecoder. The variable i represents the index of the block in a frame. The Baum-Welch decoder estimates the bit information according to the forward and backward transfer equations. The forward equation is shown as (4). The backward equation is similar to the forward equation, where s_0 and s_1 are two states of source X. $\alpha_k(s_j)$ is the probability that the information sequence reachs the state s_j, and $\beta_k(s)$ is backward probability

$$\alpha_k\left(s_j\right) \propto \alpha_{k-1}\left(s_0\right) p_{0j} + \alpha_{k-1}\left(s_1\right) p_{1j}. \tag{4}$$

The LDPC decoder accepted the information from Baum-Welch decoder and the residual information. The two decoders mutually exchange information and iteratively decode the original information.

Usually, the image sequence is two-dimensional Markov model, and the state of source X is huge, so the transfer equation which is used to estimate the probability is simplified,



(a) RD performance for the sequence of "carphone"

(b) RD performance for the sequence of "salesman"

Figure 7: The simulation results of the proposed scheme.

showing as (5), where w represents the width of the frame. p_{hij} is the transfer probability. Consider the following:

$$\alpha_k(s_j) \propto \sum_{h,i\in\{0,1\}} \alpha_{k-w}(s_h)\,\alpha_{k-1}(s_i)\,p_{hij}. \tag{5}$$

5. Experiment Results

The experiment of the proposed scheme chose the standard video sequences "carphone" and "salesman." The image format is QCIF (177 × 144), and the coding frequency is 30fps. The rule of Wyner-Ziv frames (W) and key frames (K) is "K-W-K-W". The key frames use the JPEG coding method [10], while the Wyner-Ziv frames use the IMB algorithm to distinguish the IMB and non-IMB. To the non-IMB, we use the RDVC coding and decoding method and LDPC-Baum-Welch algorithm. The reference frame is the previous frame. The check matrix of LDPC codes is generated by the PEG method [11–13]; rate is 7/8. According to the experiment results, we chose the threshold T of IMB algorithm as 64. We compared the rate-PSNR performance of these schemes as below. The encoder of H.263+ is TMN8 [14]. The results of the experiment are shown in Figure 7.

(1) H.263+interframe (I-P-I-P) coding—H.263+P.

(2) H.263+intraframe (I-I-I-I) coding—H.263+I.

(3) Conventional JPEG codec—JPEG.

(4) Pixel-IMB-based Residual Distributed Video Coding—Proposed scheme.

(5) DCT-ROI-based Wyner-Ziv residual video coding—DCT-ROI-residual scheme [15].

(6) Conventional DCT-based Wyner-Ziv coding—Wyner-Ziv.

From Figure 7, we can draw a conclusion that the proposed scheme of this paper has 1 dB improvement compared to the DCT Wyner-Ziv scheme, and approximately 0.2 dB improvement to the DCT-ROI-residual scheme. Though there is still a performance gap between our proposed scheme and the H.263 interframe scheme, our scheme is easier at coding end which is fit for Wireless Multimedia Sensor Networks.

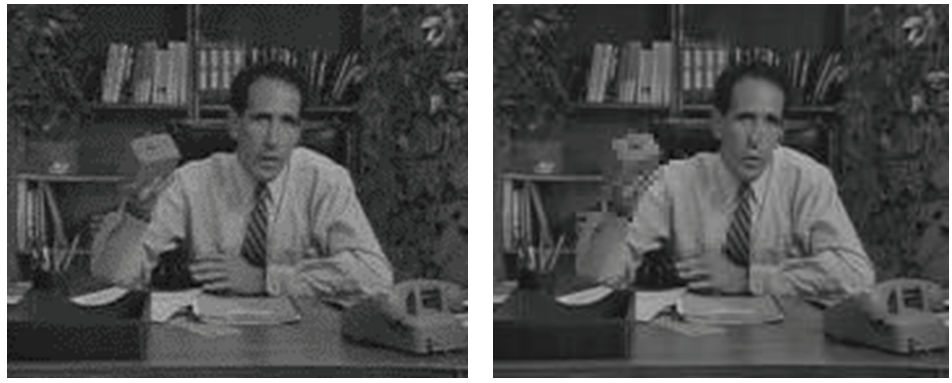
Figures 8 and 9 show the decoded 6th frame of sequences "carphone" and "salesman" using the proposed scheme and traditional RDVC scheme which is based on pixel domain. The scheme we proposed has an improvement on the quality of the decoded image. On the whole, our scheme is effective. Compared to the DCT-based DVC scheme and the traditional H.263 interframe scheme, the proposed scheme is easier at coding. By using the IMB algorithm, the proposed scheme reduced the input entropy and reduced the energy consumption. From the experiment results, we draw that our proposed scheme has an overall improvement. Because of the using of IMB algorithm, the proposed scheme avoided the application limitation of RDVC. And because of the using of LDPC-Baum-Welch algorithm, the scheme explored the spatial correlation of the original information which is more advanced than the traditional pixel-based RDVC scheme and the traditional DCT-based DVC scheme, which is more complex at encoder.

6. Conclusion

The paper proposed a new RDVC scheme based on traditional pixel-based RDVC scheme. The proposed scheme used the LDPC-Baum-Welch algorithm which can explore

(a) Pixel-IMB scheme (PSNR = 30.39 dB) (b) Traditional scheme (PSNR = 29.47 dB)

Figure 8: Decoded "carphone" sequence of proposed scheme and traditional RDVC scheme.



(a) Pixel-IMB scheme (PSNR = 36.84 dB) (b) Traditional scheme (PSNR = 35.70 dB)

Figure 9: Decoded "salesman" sequence of proposed scheme and traditional RDVC scheme.

the spatial correlation of original information. Thus, the proposed RDVC scheme can achieve better performance at pixel domain compared to traditional DCT-based scheme. The proposed scheme used IMB algorithm which can avoid the shortcoming of traditional RDVC scheme. The experiment proved that the proposed scheme reduced the complexity of the RDVC encoder and made the best use of RDVC. The proposed scheme is more suitable to the WMSNs.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant nos. 61171053, 61300239, and 61373137 and the Doctoral Fund of Ministry of Education of China under Grant no. 20113223110002.

References

[1] J. Ostermann, J. Bormans, P. List et al., "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, 2004.

[2] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, 2004.

[3] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

[4] A. Aaron, D. Varodayan, and B. Girod, "Wyner-Ziv residual coding of video," in *Proceedings of the Picture Coding Symposium*, April 2006.

[5] H. Chen and E. Steinbach, "Fast motion estimation-based reference frame generation in Wyner-Ziv residual video coding," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '07)*, pp. 168–171, July 2007.

[6] K. Misra, S. Karande, K. Desai, and H. Radha, "Complexity reduction using power-law based scheduling for exploiting spatial correlation in distributed video coding," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '08)*, pp. 2224–2227, October 2008.

[7] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform-domain wyner-ziv codec for video," in *Proceedings of the Visual Communications and Image Processing*, pp. 520–528, January 2004.

[8] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[9] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[10] ITU-T and ISO/IEC JTC1, "Digital compression and coding of continuous-tone still images," ISO/IEC, 10918-1 - ITU-T Recommendation T. 81 (JPEG), 1992.

[11] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Progressive edge-growth tanner graphs," in *Proceedings of the IEEE Global Telecommunicatins Conference (GLOBECOM '01)*, pp. 995–1001, November 2001.

[12] H. Xiao and A. H. Banihashemi, "Improved progressive-edge-growth (PEG) construction of irregular LDPC codes," *IEEE Communications Letters*, vol. 8, no. 12, pp. 715–717, 2004.

[13] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, 2005.

[14] ITU-T/SG15, "Video Codec Test Model.TMN8," Portland, Ore, USA, 1997.

[15] Z. Lin, L. Sun, and F. Xiao, "A new approach to DVC combined with ROI extration and residual video coding for multimedia sensor networks," *Journal of Electronics (China)*, vol. 30, no. 1, pp. 17–21, 2013.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 839320, 8 pages
http://dx.doi.org/10.1155/2013/839320

Research Article

An Approach Based on Chain Key Predistribution against Sybil Attack in Wireless Sensor Networks

Chunling Cheng,[1,2,3] Yaqiu Qian,[1] and Dengyin Zhang[3]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China*
[3] *Key Lab of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education, Jiangsu Province, Nanjing, Jiangsu 210003, China*

Correspondence should be addressed to Chunling Cheng; chengcl@njupt.edu.cn

Received 8 July 2013; Accepted 12 September 2013

Academic Editor: Zhijie Han

In wireless sensor networks (WSN), Sybil attack can destroy the routing and data distributed storage mechanisms through fabricating identity information of legitimate nodes. This paper presents a chain key predistribution based approach against Sybil attack. To enhance the security of common keys between neighboring nodes, during the chain key predistribution phase, our approach uses a lightweight hash function to generate several chain keys by hashing the unique identity information of every node sequentially in the trusted base station. These chain keys construct a pool of chain keys. During the pairwise key authentication establishment phase, a node-to-node chain key based authentication and exchange (CK-AE) protocol is proposed, by which each node can share the unique pairwise key with its neighboring node. The CK-AE protocol is provably secure in the universally composable security model (UCSM). Finally, we analyze our approach from the resilience of network and the performance overhead, and the results show that our approach can not only enhance the ability of resilience to Sybil attack, but also reduce the communication overhead significantly at the cost of a certain amount of computational overhead.

1. Introduction

In recent years, with the wide application of wireless sensor networks (WSN), the increasing number of the terminal devices brings the explosive growth of data. The WSN is usually deployed in many areas, including traffic control, health care, environmental monitoring, and battlefield surveillance, where the sensor nodes obtain the useful information by gathering, storing, and analyzing the large number of data [1]. However, due to the openness of deployment, the limitation of resource and the Big Data environment, security emerges as a challenging issue in ubiquitous WSN. Sybil attack is a particularly harmful attack which can forge a large number of fake identities to disrupt the routing protocol, distributed storage, and malicious behavior detection [2]. Moreover, the entire communication of the network will even be destroyed. Therefore, it is particularly important to develop an effective approach to defend against Sybil attack.

So far, some defense approaches against Sybil attack in WSN are mainly based on a random key predistribution scheme [3]. In the random key predistribution scheme, each node is preloaded with a key ring (blocks of keys) randomly selected from a large pool of keys. After the deployment step, every node exchanges several keys which belong to its key ring with each of its neighbors. If two neighbor nodes share at least one key (common key), they establish a secure link and calculate their pairwise key (session key) which is one of the common keys. The process of establishing a pairwise key can authenticate the node identity; therefore, the fake identities of a Sybil attacker can be distinguished from the legitimate nodes. Furthermore, a random key predistribution scheme is implemented before deployment; thus the authentication of node identity can depend little on the trusted base station, which reduces the computation and communication overhead of dynamically generating pairwise keys.

Karlof and Wagner proposed a mechanism based on a unique common key for WSN against Sybil attack [4]. Every node shares a unique common key with a trusted base station. When two nodes need to communicate with each other, they first signal their desire to the base station. Then the base station verifies the identifications of the two nodes via the unique common key and distributes a pairwise key to them. After that, two nodes use the distributed key to verify each other's identity and establish a session link by a Needhan-Schroeder like protocol. During the establishment of session between two nodes, the base station must be requested to authenticate the node identities, which leads to a heavy load in the base station if the number of the nodes is large in a WSN. Besides, using the same distributed key may cause the situation that the attacker can easily capture the entire network by capturing only one node. Based on the idea of the pairwise key in [4], Newsome et al. proposed a random key predistribution scheme against Sybil attack in WSN [5]. In the scheme, a random set of keys or key-related information is assigned to each node. Then each node can discover the common keys it shares with its neighbors and the common keys will be used as a session key to ensure node-to-node secrecy. Nevertheless, the set of keys or key-related information is just determined by a pseudorandom function (PRF) in this scheme; therefore, the unsecure keys or key-related information may lead to unsecure session keys between the neighbor nodes. Once a Sybil attacker breaks the PRF, it will be able to quickly obtain the keys or key-related information in other nodes and disguise their identities by forging session keys. On the basis of random key predistribution scheme, Roberto et al. proposed a pairwise key establishment mechanism based on the node identity information in an attempt to protect the WSN from the Sybil attack [6]. However, due to the difficulty of constructing the specific node identity information, the mechanism is not practical and effective. Furthermore, Feng and Ma improved the Roberto's mechanism and presented a novel approach which is node identity witness information validation for random secret information predistribution to defend against Sybil attack [7]. But in this approach, the transmitter/receiver must send more messages during the process of establishing a secure link, which means more communication overhead.

Qian proposed an improved key predistribution scheme in which each node calculates the derived keys by using a hash function once [8]. This solution enhances the security of the original keys. However, the derived keys are calculated by every node after deployment. So, the computational overhead of the nodes is increased. In order to enhance the resilience of the network against attackers, Bechkit et al. proposed a novel hash-based key predistribution scheme [9]. Before deployment a hash function h is preloaded to the memory of each node. Then every node applies the hash function h to each key of its key ring. After that, the neighbor nodes calculate the pairwise keys by the hash function h and establish a secure link. However, in this solution the calculated pairwise keys are not unique. So the probability of successfully forging the pairwise key by a Sybil attacker is increased. If the forged pairwise key is the same as the legitimate one, the communication of the neighbor nodes will be disrupted by the Sybil attacker's fake identity.

As described above, the existing approaches play a certain part in defending against Sybil attack in WSNs, but some issues still exist, such as high communication overhead of the trusted base station, unsecure keys, high communication overhead of the process of node identity authentication, and nonunique pairwise key and so on. In this paper, we propose a chain key predistribution based approach to protect the WSN from Sybil attack. In the chain key predistribution phase, the chain keys are calculated with the entire node identity information through a lightweight hash chaining technique in the trusted base station, which conceals the original keys and makes the common chain keys more secure that preloaded in different nodes. Then a chain key pool which constitutes several hash chains of equal length is constructed. After deployment, we develop a node-to-node chain key based authentication and exchange (CK-AE) protocol in the pairwise key authentication establishment phase which is secure against the Sybil attackers. At last, we prove the security of our approach and make analysis for the connectivity, communication overhead, storage overhead, and computational overhead.

2. A Chain Key Predistribution Based Approach against Sybil Attack in WSNs

The proposed approach is divided into two phases: the chain key predistribution phase and the pairwise key authentication establishment phase. Two phases are described in detail below.

2.1. Chain Key Predistribution Phase. The issue of the unsecure keys usually makes the nodes vulnerable to the Sybil attack. If an attacker captures one node, it can get all the key information from its memory and impersonate other legitimate nodes. Such an attack can compromise not only adjacent links of compromised links but also external links that are independent of the compromised nodes. Therefore, in order to make the key information of nodes more secure, we generate several chain keys by hashing the unique identity information of every node sequentially based on a lightweight hash function H^{Nyb} [10] in the trusted base station. As known, the main characteristic of hash functions is that knowing a value of the chain it is computationally infeasible to determine the backward values. Therefore, our chain key predistribution scheme can conceal the original keys and enhance the resilience of common keys shared by the neighbor nodes. In our scheme, when an attacker captures one or more original keys it can only discover a forward value.

In the chain key predistribution phase, the generation of the hash chains is implemented in the trusted base station, and then the hash chains are constructed into a pool of chain keys. Take the generation procedure of the hash chain L_i for example. It is assumed that seed_i is an original key and the derived factor is a set of the node identities SIDSet (an ordered sequence of values). The set SIDSet is $\{\mathrm{SID}_1, \mathrm{SID}_2, \ldots, \mathrm{SID}_j, \ldots, \mathrm{SID}_N\}$ where SID_j $(1 \leq j \leq N)$

is the unique node identity information and N is the number of the nodes. The procedure of generating the jth chain key in the hash chain L_i is given as follows:

$$k_j^i = H^{\text{Nyb}}\left(k_{j-1}^i, \text{SID}_j\right), \quad \{i, j \in N^*,\ 1 \le j \le N\}. \tag{1}$$

N^* represents a set of positive integer. According to (1), $k_1^i = H^{\text{Nyb}}(\text{seed}_i, \text{SID}_1)$, $k_2^i = H^{\text{Nyb}}(k_1^i, \text{SID}_2)$, and so on. The last value of the hash chain k_N^i is $H^{\text{Nyb}}(k_{N-1}^i, \text{SID}_N)$. We define the value K_N^i as the hash chain tag Tag_i which is calculated from seed_i. Different hash chain tags are calculated from different original keys. The other hash values $(k_1^i, k_2^i, \ldots, k_{N-1}^i)$ are defined as the chain keys. To illustrate our idea about the generation procedure of the hash chain L_i, let us refer to Figure 1.

When a given hash value k_m^i needs to be verified in the above hash chain, it is feasible to calculate k_n^i $(0 < n < m)$ $m-n$ times through the hash function H^{Nyb}. Then we compare the output $k_m^{i\,*}$ with the given hash value k_m^i. If $k_m^{i\,*} = k_m^i$; it means that $k_m^{i\,*}$ is a correct hash value. In order to construct a pool of chain keys, we select M different original keys $\{\text{seed}_1, \text{seed}_2, \ldots, \text{seed}_M\}$ from the trusted base station. By repeating the above generation procedure in Figure 1, M hash chains with the same length are generated. In our approach, the pool of chain keys is composed of M hash chains, which is shown in Figure 2. Each hash chain contains $N-1$ chain keys and the size of the pool K is $M * (N-1)$.

After the construction of the pool of chain keys, each node is preloaded with some chain key related information. The detailed performance step of the chain key distribution is given as follows. (1) Each node selects m different chain keys from the pool. It is noted that the method about selecting the chain keys from the pool is not limited. Moreover, the indexes of every selected chain key in the same hash chain are also preloaded in each node. Besides, if there is any chain key selected from one hash chain, the hash chain tag of this hash chain is also stored in the memory of the node. (2) Then the hash function H^{Nyb} is preloaded to the nodes.

Due to the limited resource of the sensor nodes, we apply the hash function H^{Nyb} which is based on WH function [11] to implement the computation of the chain keys. WH function not only has the traditional security of hash function clusters, but also gets a good efficiency about computation. Therefore, it is suitable to be applied to the actual hardware of the sensor nodes.

After the chain keys predistribution phase, all the sensor nodes are randomly deployed in a designated area. The trusted base station just participates in the first phase without communicating with the nodes in the second phase, therefore, the proposed approach has the advantage of reducing the high communication overhead of the base station. Figure 1 shows that every chain key is calculated from different original keys by hashing the unique identity information of nodes sequentially. So, the original keys and the node identities are concealed by such computation, and the resilience of shared pairwise keys between the neighbor nodes is also enhanced. When an attacker captures one or more original keys, it can not fake a derived chain key correctly with an unknown number of times.

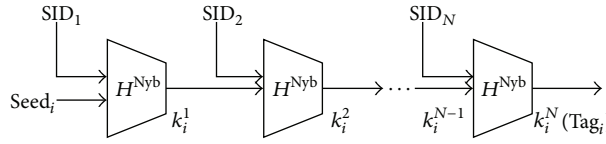


Figure 1: Generation procedure of the hash chain.

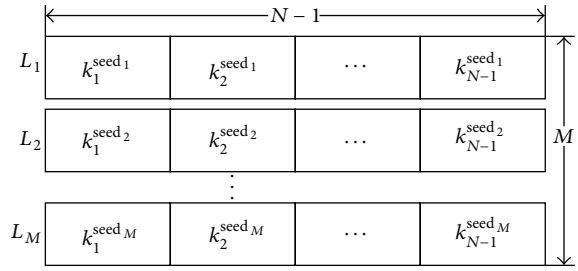


Figure 2: Composition of the pool of chain keys.

2.2. Chain Key Based Pairwise Key Authentication Establishment Phase. A session link established by a random key predistribution scheme usually leads to an important issue that the pairwise key is not unique [9, 12]. Thus, a Sybil attacker can easily disrupt the communication between the neighbor nodes. To solve this issue, we propose a node-to-node chain key based authentication and exchange (CK-AE) protocol in the pairwise key authentication establishment phase. By the CK-AE protocol, the neighbor nodes authenticate the identity information of the opposite node and then calculate a unique pairwise key to establish a secure session link for defending against the adversaries.

In the CK-AE protocol, one node stores multiple different key chains. For example, the ring of m chain keys of S_A is $\{k_{j,A}^i\}$, where $1 \le j \le m$, $1 \le i \le M$. A protocol is usually executed distributedly, so we mark the different protocol instances as Sid. It is assumed that S_A and S_B are neighbors. The detailed description of CK-AE protocol is given as follows where "$\oplus$" represents XOR operation and "|" is connection operation. The parameters of hash function H^{Nyb} are exchangeable, which means $H^{\text{Nyb}}(x, y) = H^{\text{Nyb}}(y, x)$ [10].

(1) $S_A \rightarrow S_B$: after deployment, S_A broadcasts the packet $\{\text{Sid}_x, E(\text{Tag}_{i,A}, k_{j,A}^i \mid \text{Index}(j)_A \mid \text{SID}_A)\}$ to its neighbors. In this packet, Sid_x indicates the session identifier. $k_{j,A}^i$ is a chain keys which is selected from the ring of m chain keys of S_A. $\text{Index}(j)_A$ represents the corresponding index of chain key $k_{j,A}^i$. SID_A is the node identity information of S_A. $\text{Tag}_{i,A}$ is the hash chain tag which $k_{j,A}^i$ is selected from. In our approach, we use $\text{Tag}_{i,A}$ as the broadcast communication key. $E(k, M)$ means that M is encrypted by using the key k.

(2) $S_B \rightarrow S_A$: if S_B and S_A have the chain keys which are selected from the same hash chain, S_B can decrypt the received broadcast packet by using the preloaded hash chain tag $\text{Tag}_{i,B}(= \text{Tag}_{i,A})$. After decrypting the packet, S_B will use the hash function H^{Nyb} to authenticate the received chain key $k^i_{j,A}$. During authentication, S_B first finds the hash chain L_i that $\text{Tag}_{i,B}$ belongs to, then selects the chain key $k^i_{j,B}$ which is preloaded from the hash chain L_i and its corresponding index $\text{Index}(j)_B$ in L_i. Second, S_B calculates the difference between two indexes diff = $|\text{Index}(j)_B - \text{Index}(j)_A|$. (a) If $\text{Index}(j)_B \geq \text{Index}(j)_A$, S_B calculates $k^i_{j,B}{}^* = H^{\text{Nyb}}(k^i_{j,A}, \text{SID}_{j+1})^{\text{diff}}$ (using H^{Nyb} to calculate diff times). Then if $k^i_{j,B}{}^* = k^i_{j,B}$, the chain key $k^i_{j,A}$ is verified correctly by S_B. It is means that S_B knows S_A is a legitimate node; then it can establish a pairwise key k_{AB} with S_A. (b) If $\text{Index}(j)_B < \text{Index}(j)_A$, S_B calculates $k^i_{j,A}{}^* = H^{\text{Nyb}}(k^i_{j,B}, \text{SID}_{j+1})^{\text{diff}}$. Then if $k^i_{j,A}{}^* = k^i_{j,A}$, the chain key $k^i_{j,A}$ is verified correctly by S_B. It is also means that S_B knows S_A is a legitimate node; then it can establish a pairwise key k_{AB} with S_A.

After authenticating S_A, S_B selects a random number r_B and calculates the pairwise key $\text{key}_{AB} = k^i_{j,A} \oplus k^i_{j,B}$. Moreover, S_B calculates two temporary variables $X_B = k^i_{j,B} \oplus \text{SID}_B$ and $Y_B = \text{Index}(j)_B \oplus r_B$. At last, S_B delivers the packet $\{\text{Sid}_x, X_B, Y_B, \text{SID}_B, \text{diff}, r_B\}$ to S_A by unicast.

(3) S_A: after receiving the packet from S_B, according to the session identifier Sid_x S_A learns that the packet is a response message from S_B. Then S_A calculates $k^i_{j,B}{}^* = X_B \oplus \text{SID}_B$ and $\text{Index}(j)_B{}^* = Y_B \oplus r_B$. By judging the equation diff = $|\text{Index}(j)_B{}^* - \text{Index}(j)_A|$, S_A certifies S_B whether it is a legitimate node. If the equation is correct, S_B is a legitimate node, otherwise not. Finally, S_A calculates the pairwise key $\text{key}_{AB} = k^i_{j,A} \oplus k^i_{j,B}{}^*$ which can ensure a secure session link between the neighbor nodes.

In the above pairwise key authentication establishment phase, if S_B failed to decrypt the data packets with $\text{Tag}_{i,B}$, S_B should discard the packet. By this way, we can prevent the leakage of packet information if S_B is captured by an adversary. When the key chain cannot be authenticated, the nodes must discard the packet too, because the chain key may be forged by a Sybil attacker. In addition, after establishing the unique pairwise key, the nodes will delete the sent and received packets.

3. Security Analysis

The security of a protocol is usually defined and analyzed within the universally composable security model (UCSM) [13–15]. When a protocol P is proved to be secure in this model, it is able to ensure that when the protocol P runs in parallel with other protocols or runs separately, the protocol P is still secure. Generally speaking, the definition of the security of a protocol within the UCSM which means a UC-secure protocol is given as follows: the mutual information of protocol between the parties in any real model is computationally indistinguishable for the Environment (E) with the mutual information in the ideal model which a trusted party "ideal functionality F" exists in.

In this section, we first present the ideal functionality of the CK-AE protocol within the UCSM. Then we prove the security of the CK-AE protocol according to the definition within the UCSM. At last, the defense capability against Sybil attack of our approach is analyzed.

3.1. Ideal Functionality of the CK-AE Protocol. Based on the chain key predistribution scheme, we give our formulation of the ideal functionality $F_{\text{CK-AE}}$ for CK-AE protocol within the UCSM. The ideal functionality $F_{\text{CK-AE}}$ in the ideal model which describes the functions of the CK-AE protocol is shown as follows.

Ideal Functionality $F_{\text{CK-AE}}$. Functionality $F_{\text{CK-AE}}$ interacts with the parties S_A, S_B and an ideal adversary S via the following queries.

(1) Upon receiving a query {New Session, Sid, $k^i_{j,A}$, S_A, S_B, role} from party S_A: send {New Session, Sid, S_A, S_B, role} to S_B. In addition, if this is the first NewSession query, or if this is the second NewSession query and there is a record $(S_B, S_A, k^i_{j,B})$ then record $(S_A, S_B, k^i_{j,A})$ and mark it "fresh." Other conditions are ignored.

(2) Upon receiving a query {Test Session, Sid, S_A, $k^i_{j,A}{}^?$} from S: if there is a record of the form $(S_A, S_B, k^i_{j,A})$ which is "fresh," then do: if $k^i_{j,A}{}^? = k^i_{j,A}$, then mark the record "compromised" and reply to S {correct guess}. If $k^i_{j,A}{}^? \neq k^i_{j,A}$, then mark the record "interrupted" and reply to S with {wrong guess}.

The variable role in the message is included in order to let a party know if it is playing the initiator or responder role in the protocol. In (1) the ideal functionality describes the behavior of the initiator or responder role in the CK-AE protocol. And in (2) the ideal functionality describes a situation that the ideal adversary S tries to disguise the identity of party in the CK-AE protocol. Moreover, it explains that when one party is captured, the ideal functionality will know that the protocol is under attack and it will terminate the protocol.

3.2. Security of CK-AE Protocol. It is difficult to prove the security of a protocol within the UCSM without any security assumptions as premises. But many protocols can be proved secure within the UCSM based on a common reference string (CRS) model [16]. So, in this paper we prove the security of the CK-AE protocol based on a CRS model within the UCSM. The path to proof the security is as follows:

firstly construct a scene of simulation, then prove that the mutual information of the CK-AE protocol between the real model and ideal model is computationally indistinguishable for E.

Theorem 1. *Based on the CRS model, the CK-AE protocol is secure within the UCSM.*

Proof. Assume that A is a real-model adversary which participates in the interaction with the real parties which are running the CK-AE protocol. Then we construct an ideal-model attacker S (simulator S) for such ideal functionality $F_{\text{CK-AE}}$. Based on the CRS model, S gets some common information about the parties, such as Tag_i which the parties hold. Because A is equal to its duplicate copy $\widetilde{A}$ and $\widetilde{A}$ can call S to work, S is able to get all the information that A has submitted. Now Attacker S simulates the following two situations.

(1) S simulates that A disguises the legitimate node S_A. At first, S gets the fake information $\text{diff}^?$ and $k^i_{j,A}{}^?$ that A submits. According to the CRS model, S guesses the chain key $k^i_{j,B}{}^?$ which S_B holds then submits it to the ideal functionality $F_{\text{CK-AE}}$. If S guesses correctly, $\text{diff}^?$ and $k^i_{j,B}{}^?$ (or $k^i_{j,A}{}^?$) could be used repeatedly by it; then S submits $k^i_{j,B}{}^?$ (or $k^i_{j,A}{}^?$). S must guarantee that the output $k^i_{j,B}{}^{*?}$ (or $k^i_{j,A}{}^{*?}$) which is calculated by $k^i_{j,B}{}^?$ (or $k^i_{j,A}{}^?$) and $k^i_{j,B}{}^*$ (or $k^i_{j,A}{}^*$) is indistinguishable.

(2) S simulates that A disguises the legitimate node S_B. S disguises S_A to submit $\text{diff}^?$; then according the fake variable information X_B, Y_B and a random number r_B which are submitted by A, S guesses the chain key $k^i_{j,A}{}^*$ of S_A and submits it to the ideal functionality $F_{\text{CK-AE}}$ based on the CRS model. If it guesses correctly, X_B and Y_B can be used again by S; then S submits $\text{diff}^?$ and $\text{Index}(j)_B{}^{*?}$. Similarly, the simulator S must guarantee that the output $\text{diff}^?$ (or $\text{Index}(j)_B{}^{*?}$) which is calculated by $X_B(Y_B)$ and diff (or $\text{Index}(j)_B{}^*$) are indistinguishable.

In this paper, we prove the indistinguishability of mutual information in ideal model and in real model by contradiction. Assumed that Environment (E) can distinguish the behavior of the simulator S; thus the following two cases exist. (1) When E gets the correct chain keys ($k^i_{j,A}$ and $k^i_{j,B}$) and is able to distinguish the chain keys submitted by the CRS model from the correct ones, in accordance with Theorem 1 in [12], the probability that this case happens is negligible. (2) E has got the difference information diff. If the number of the ring of chain keys m conforms to (2), and according to the Lemma 2 when some nodes are captured, the probability that the adversary obtains the difference information of the indexes of two chain keys in any other uncaptured nodes is negligible. In conclusion, the mutual information of CK-AE protocol between the parties in the ideal model is computationally indistinguishable for the E with the information in the real model. In other words, the CK-AE protocol is a UC-secure protocol. Proof ends. □

3.3. Analysis of Defense Capability against Sybil Attack. The resilience of the chain keys (the impact to secure session links between nodes brought by the exposed chain key in other captured nodes) is a significant security requirement in our approach. The stronger resilience of the chain keys is the more effective capability against Sybil attackers our approach has. The chain keys are predistributed before deployment; therefore, based on the Lemma 2 if m (the size of the ring of chain keys) conforms to (1), then the probability that attacker obtains other nodes' chain key information by capturing some nodes is negligible.

Lemma 2 (see [17]). *Assuming that the total number of nodes in WSN is N and the size of the ring of secret keys in one node is m. If the size of the pool K meets the inequality $K \geq N \log N$, and three variables K, N, and m meet* (2)*, then the wireless sensor network which is based on the secret keys predistribution is connected well. Furthermore, when some nodes are captured by adversaries, the impact to the secret keys in other uncaptured nodes is negligible:*

$$\frac{m^2}{K} = c\frac{\log N}{N}. \tag{2}$$

In our approach, we definite the resilience of chain keys of nodes as follows: the probability that each chain key is selected to each node from the pool K (equal to $(N-1) * M$) of chain keys is m/K. If a Sybil attacker captures w nodes, the probability that one of the chain keys in the uncaptured nodes is just equal to one of the chain keys in w nodes is $1-(1-m/K)^{\omega}$. Moreover, the probability that an attacker tries to forge the chain keys of legitimate nodes is $2^{-\lambda}$ [10], so the resilience of the chain keys is $2^{-\lambda} * (1-(1-m/K)^{\omega})$.

Figure 3 shows the relationship between the number of captured nodes and success attack rate in Newsome's approach [5], Feng's approach [7], HC approach [9], and our approach when the total number of nodes in WSN is 800 and the size of chain key pool is 10000.

It can be seen from Figure 3 that with the increasing number of the captured nodes, the difference of four approaches becomes more obvious. The success Sybil attack rate of first three different approaches all increase obviously, but the curve which represents our approach still stays stable. For example, when the number of captured nodes is 300, the success Sybil attack rate of the first three approaches is 67.91%, 2.12%, and 0.88%, respectively. But in our approach the probability is only 0.065%. The reason is that the hash function H^{Nyb} conceals the node identity information and the original keys by means of calculating some times depending on the difference of two indexes. Therefore, the defense capability against Sybil attack of our approach is stronger than that of three other approaches.

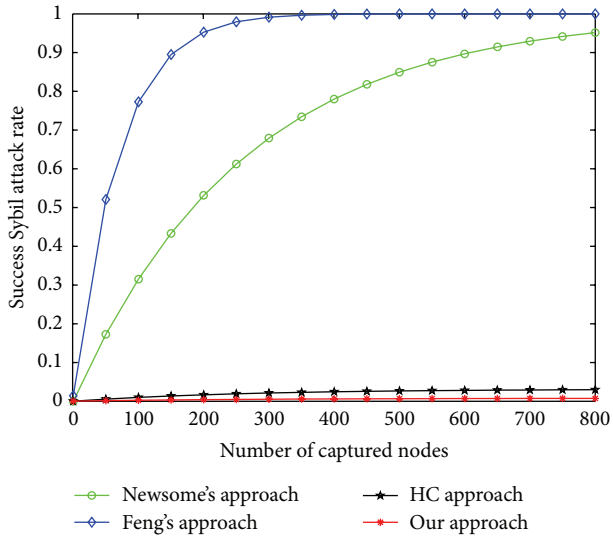


Figure 3: Defense capability against Sybil attack.

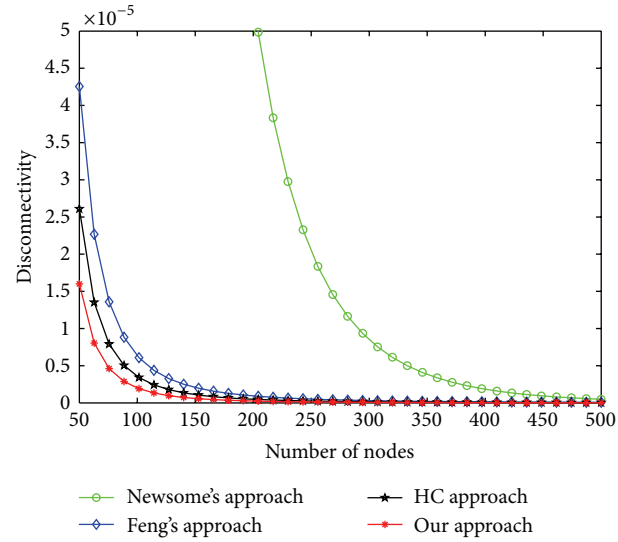


Figure 4: Disconnectivity of the network.

4. Performance Analysis

The performance of defense approaches against adversaries in WSNs can be evaluated on the basis of several criteria, such as connection, communication overhead, storage overhead, and computational overhead. In this section, we describe the performance evaluation of our approach and compare it with Newsome's approach [5], Feng's approach [7], and HC approach [9] in the above four aspects.

4.1. Connectivity. The secure connectivity (the probability that two neighbor nodes are able to establish a secure link) is considered as fraction of secured links among all possible links in the network. Thus, all defense approaches against Sybil attack based on a key predistribution scheme must ensure the connectivity of the network [18]. In our approach, if two neighbor nodes obtain the chain keys which are from the same hash chain (they share this hash chain), there is a secure link between them. Let us assume that the total number of network is N and each node is predistributed with a ring of m chain keys which are selected from a pool K at random. Besides, the pool consists of M hash chains and there are $N-1$ chain keys in each hash chain. If S_A randomly selects m chain keys from any t hash chains, S_B only can randomly select m chain keys from the other $M-t$ hash chains which S_A does not select from; therefore the disconnectivity of the network is described as follows:

$$P\left[\text{disconnectivity}\right] = 1 - P\left[\text{connectivity}\right] = \sum_{t=1}^{M-1} \frac{\binom{t\times(N-1)}{m}\binom{(M-t)\times(N-1)}{m}}{\binom{M\times(N-1)}{m}^2}. \quad (3)$$

Figure 4 shows the comparison of the connectivity of network between the above three approaches and our approach. In (3), the parameters K, N, and m should conform to Lemma 2 and $M = K/(N-1)$.

As can be seen from Figure 4, the proposed chain key predistribution scheme in our approach makes the network get a better connectivity than the others. For example, when the parameter N is 100 and accordingly K and m should be 200 and 14, respectively, the disconnectivity of the other three approaches is 6.32%, $6.32 * 10^{-4}$%, and $3.51 * 10^{-4}$%, respectively. However, it is only $2.13 * 10^{-4}$% in our approach. Besides, each node just need preload 14 chain keys that can make the network get a good connectivity; however, the other approaches need more than 20 chain keys preloaded in the memory of one node. Therefore, in the same case the proposed chain key predistribution scheme is more effective in reducing the storage requirement of the nodes. What causes a better connectivity is that hashing the unique identity information of every node sequentially to construct a pool of chain keys enhances the correlation of the common keys hold by different nodes and reduces the storage requirement of the ring of chain keys.

4.2. Communication Overhead. Communication represents the massage exchange between the nodes in WSNs, which consumes the limited energy resource of network. It is one of the important factors to evaluate the performance of defense approaches. We compare the communication overhead of our approach with that of the other three approaches. Based on the energy-consuming model in [19], we carry out simulations under the ONE software. Experimental parameters are set as follows: node deployment area = 1400 m * 1400 m, original number of nodes = 100, step = 20, communication range = 100 m, transmit speed of the node interface = 250 kBps, and the radio dissipates E_{elec} = 50 nJ/bit and ε_{amp} = 100 pJ/bit/m^2 for the transmit amplifier. The experimental results are shown in Figure 5.

In Feng's approach, two more messages need to be exchanged by the transmitter and receiver in the process of establishing the pairwise key; therefore, the complexity of

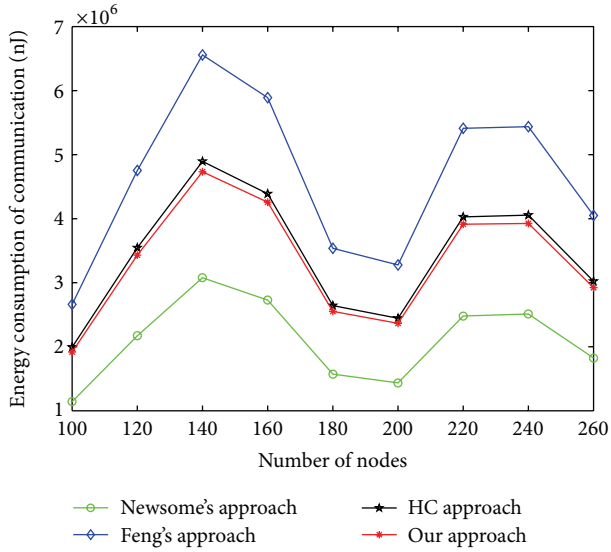


Figure 5: Energy consumption of communication.

communication is $O(m + 2)$. In Newsome's approach, the complexity of communication is $O(m)$. Because HC approach maintains the same pairwise key establishment phase with the Newsome's approach which is based on the key identifier exchange, it introduces the same complexity of communication as Newsome's approach. Only one message exchange needed to establish the pairwise key in our approach, so the performance of the CK-AE protocol is better than Feng's approach obviously. The complexity of communication in our approach is $O(m)$.

As can be seen from Figure 5, with the increasing number of nodes, the energy consumption of communication in our approach is always lower than that of HC approach, which is because the bits of the transmitted/received message are shorter than that of HC approach under the same complexity of communication. But compared with Newsome's approach, our approach needs to transmit/receive a little more bits of message (such as Tag_i) in order to conceal the original keys and enhance the resilience of network against the Sybil attack. In addition, Figure 5 shows that the energy consumption of communication increases or decreases irregularly, which is caused by the random deployment of sensor nodes in the simulation. When the number of nodes in WSN is 140, the number of neighbor nodes is the largest. Thus the energy consumption of communication is larger than other cases in Figure 5.

4.3. Storage Overhead. In all key predistribution scheme based defense approaches, each node needs to store a certain amount of keys. At first, we analyze the storage requirement of each node in Newsome's approach. $M_{\text{size}} = m \times (k_{\beta_i\,\text{size}} + \beta_{i\,\text{size}})$. $k_{\beta_i\,\text{size}}$ is the size of a given key, $\beta_{i\,\text{size}}$ represents the size of the key identifier, and m is the number of keys. Similarly, HC approach requires the same storage memory as the Newsome's approach. In Feng's approach, the required storage memory is $K \times Z(k_i)_{\text{size}} + m \times w_j(k_i)_{\text{size}}$. Due to K (the size of the pool) accumulated values are needed to be stored; therefore when K is a large number, a lot of storage memory will be occupied. In our approach the required memory for each node is $m \times (k^i_{j\,\text{size}} + \text{Index}(j)_{\text{size}} + \text{Tag}_{i\,\text{size}})$, where $k^i_{j\,\text{size}}$ is the size of chain keys, $\text{Index}(j)_{\text{size}}$ is the size of the corresponding index, and $\text{Tag}_{i\,\text{size}}$ is the size of the tags of the selected hash chain. Due to the additional storage of the tags, the occupied storage memory is more than that of the Newsome's and HC approaches but less than that of Feng's approach. It is noticed that when $\text{Tag}_{i\,\text{size}} = 128$ bits and m is less than 17 according to (1), the additional storage memory is 2176 bits (272 B) at most. However, the size of existing sensor nodes' memory is much bigger than 272 B. For example, Micaz has a memory of 512 KB which is equipped with CC2420. Therefore, our approach is suitable for the exiting nodes.

4.4. Computational Overhead. In the pairwise key establishment phase, the complexity of the computation for transmitter/receiver in the Newsome's, Feng's and HC approaches is $O(m)$, $O(m + 2H \times m)$ and $O(H \times m)$, respectively, where m is the number of messages that is transmitted or received and H is a pseudorandom function. In the CK-AE protocol of our approach, in order to determine a unique pairwise key between the neighbor nodes, the transmitter encrypts the message once then the receiver decrypts the message and applies the hash function H^{Nyb} a number of times (diff times) to compute the unique pairwise key. The average times of the above hash computation is $(2/N(N+1)) \sum_{i=0}^{N-1}((N-i)(N-i-1)/2)$, so the complexity of the computation of transmitter is $O(m + m \times E)$ and that of receiver is $O(m \times H + N)$. By contrast, our approach consumes a little more energy for computation than Newsome's and HC approaches, but less than that of the Feng's approach. But current studies have shown that the energy consumption of communication is much larger than that of computation [20]. Therefore, our approach is more suitable for the resource-limited WSN because of that the approach can get a better resilience of network against Sybil attack.

As shown above, the proposed approach not only enhances the security of the keys, which brings a better connectivity and stronger resilience of network but also reduces the significant communication overhead. Our approach solves the existing issues well, but it may incur unbalanced computational overhead.

5. Conclusion

The existing defense approaches are likely to be unsuitable to solve the problems for Sybil attack in WSN, such as high communication overhead of the trusted base station, unsecure keys, high communication overhead of the process of node identity authentication, and nonunique pairwise key. Focusing on these problems, we propose an approach based on chain key predistribution to defend against the Sybil attack in this paper. First, several chain keys are generated by hashing the unique identity information of every node

sequentially in the trusted base station, and they are constructed to a pool of chain keys. Then through the CK-AE protocol, neighbor nodes authenticate with each other and establish a unique pairwise key to protect the WSN from fake pairwise keys by Sybil attackers. This approach is provably secure against Sybil attack. The theoretical analysis shows that compared with other approaches, our approach can enhance the resilience of the WSN and reduce the communication overhead.

Acknowledgments

The paper is sponsored by the Research and Innovation Projects for Graduates of Jiangsu Province (nos. CXZZ12_0483 and CXLX12_0481), the Science and Technology Support Program of Jiangsu Province (no. BE2012849), the National Natural Science Foundation of China (no. 61071093), and the Priority Academic Program Development of Jiangsu Higher Education Institutions (yx002001).

References

[1] F. Xiao, J. K. Liu, and J. Guo, "Novel side information generation algorithm of multiview distributed video coding for multimedia sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 582403, 7 pages, 2012.

[2] P. Anitha, G. Pavithra, and P. Periasamy, "An improved security mechanism for high-throughput multicast routing in wireless mesh networks against Sybil attack," in *Proceedings of the IEEE International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME '12)*, pp. 125–130, Tamilnadu, India, 2012.

[3] M. A. Simplício, P. S. L. M. Barreto, C. B. Margi, and T. C. M. B. Carvalho, "A survey on key management mechanisms for distributed wireless sensor networks," *Computer Networks*, vol. 54, no. 15, pp. 2591–2612, 2010.

[4] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[5] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: analysis & defenses," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 259–268, April 2004.

[6] R. Di Pietro, L. V. Mancini, and A. Mei, "Random key-assignment for secure wireless sensor networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 62–71, ACM, October 2003.

[7] T. Feng and J. Ma, "New approach against Sybil attack in wireless sensor networks," *Journal on Communications*, vol. 29, no. 6, pp. 13–19, 2008.

[8] S. Qian, "A novel key pre-distribution for wireless sensor networks," *Physics Procedia*, vol. 25, pp. 2183–2189, 2012.

[9] W. Bechkit, Y. Challal, and A. Bouabdallah, "A new class of Hash-Chain based key pre-distribution schemes for WSN," *Computer Communications*, vol. 36, no. 3, pp. 243–255, 2012.

[10] K. Nyberg, "Fast accumulated hashing," in *Proceedings of the 3rd Fast Software Encryption Workshop*, pp. 83–87, Springer, Berlin, Germany, 1996.

[11] K. Yukdrl, J. Kaps, and B. Sunar, "Universal hash functions for emerging ultra-low-power networks," in *Proceeding of the Communications Networks and Distributed Systems Modeling and Simulation Conference*, pp. 89–95, San Diego, Calif, USA, 2004.

[12] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, Washington, DC, USA, November 2002.

[13] T. Wu, Y. Tseng, and T. Tsai, "A revocable ID-based authenticated group key exchange protocol with resistant to malicious participants," *Computer Networks*, vol. 56, no. 12, pp. 2994–3006, 2012.

[14] R. Canetti and M. Vald, "Universally composable security with local adversaries," in *Security and Cryptography for Networks*, pp. 281–301, Springer, Berlin, Germany, 2012.

[15] R. Dowsley, J. Müller-Quade, A. Otsuka, G. Hanaoka, H. Imai, and A. C. A. Nascimento, "Universally composable and statistically secure verifiable secret sharing scheme based on pre-distributed data," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 94, no. 2, pp. 725–734, 2011.

[16] Y. Tian, J. Ma, and C. Peng, "Universally composable mechanism for group communication," *Chinese Journal of Computers*, vol. 35, no. 4, pp. 645–655, 2012.

[17] D. Roberto, V. Manceng, and M. Alessandro, "How to design connected sensor networks that are provably secure," in *Proceedings of the 2nd IEEE International Conference on Security and Privacy for Emerging Areas in Communication Networks*, pp. 259–266, Baltimore, Md, USA, 2006.

[18] Y. Wang, Y. Liu, and H. Jin, "The study on key pre-distribution methods for wireless sensor networks," *Physics Procedia*, vol. 25, pp. 560–567, 2012.

[19] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS-33)*, pp. 1–10, January 2000.

[20] Z. G. Wan, Y. K. Tan, and C. Yuen, "Review on energy harvesting and energy management for sustainable wireless sensor networks," in *Proceedings of the 13th IEEE International Conference on Communication Technology (ICCT '11)*, pp. 362–367, Jinan, China, September 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 642986, 13 pages
http://dx.doi.org/10.1155/2013/642986

Research Article

Research on Migration Strategy of Mobile Agent in Wireless Sensor Networks

Dai Ting,[1,2] Huang Haiping,[1,2,3] Lu Yang,[1,2] Wang Ruchuan,[1,2,4] and Pan Xinxing[1,2]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China*
[3] *College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[4] *Key Lab of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*

Correspondence should be addressed to Huang Haiping; hhp@njupt.edu.cn

Received 16 July 2013; Accepted 9 September 2013

Academic Editor: Chongsheng Zhang

Big data and distributed computing are of great importance in wireless sensor networks (WSNs). They are always bonded together, and the latter one upholds the former one. In distributed computing, mobile agent model is the mainstream technology. With autonomy, communicativeness, mobility, and role, mobile agent model is more suitable for large-scale, resources-restrained WSN to deal with big data. This paper mainly studies migration schemes for mobile agents, determines core factors of migration strategy, and proposes SMLA and DMLA algorithms. And aiming at revealing the characteristics of target tracking, this paper puts forward pid-DMLA algorithm; and considering multiple agents' cooperation, it presents Mpid-DMLA algorithm. Moreover, this paper evaluates and analyzes the advantages and disadvantages of the above-mentioned four algorithms by simulations.

1. Introduction

In wireless sensor networks (WSNs), big data is the intensification of information architecture, while distributed computing is the intensification of infrastructure and resource [1]. Distributed computing models can bring benefits to big data. And there are two categories of distributed computing: client/server model and mobile code model. Mobile agent model consists of remote computation, code on demand, and mobile agent model [2].

In client/server model, client asks for services, while server provides resources, services, and methods for executing services. There are many distributed systems adopting this model [3, 4]. Mainwaring proposed a habitat monitoring system, which is accomplished by transmitting data to gateway [3]. Ivy is the distributing system in WSN studied by the University of California, Berkeley, CA, USA [4]. Although the client/server model has been widely used, its drawbacks are very obvious, especially in WSN [5]. Firstly, it needs a series of supernodes as a processing center. Secondly, in multiclient systems, overdependence on bandwidth reduces the transmitting performance. Thirdly, the nodes which are closer to the server need to transmit more data; thus, they consume more energy.

Mobile agent model supports distributed computing, too. The processing center/centralized information system (CIS) is in charge of sending and recovering mobile agents, while agents migrate between sensor nodes and are responsible for data exchange. A typical exemplar is the data fusion algorithm based on mobile agent to track objects in WSN [6]. In general, mobile agent is a special program which can be executed autonomously. And it is also defined as an entity with the following four attributes: agent identifier, data space, method, and migration table. Agent identifier describes the uniqueness of each agent; data space carries part of the integrated results; method is used to deal with missions or execute codes; and migration table stores the routing information of an agent and it can be preset or modified dynamically.

Compared with traditional computing models which need amount of data interaction between nodes and CIS (or base station), mobile agent model has many merits. As

illustrated above, in WSN, mobile agents dynamically migrate to the CIS with service request. With CIS accomplishing the computing operations locally and returning the results to the nodes, agents can be devoid of the transmission of big data and the reliance on the network bandwidth, which improves the efficiency. The specific advantages of mobile agent model, especially, in terms of energy consumption, reliability, and large-scale distribution, are declared in [7]. However, the consequent challenges are not negligible, like the high failure rate, unreliable communication link, high dependence on bandwidth, and so on.

2. Related Work

Although mobile-agent-based distributed model has exceptional advantages in WSN [8], dynamic migration is necessary in practical deployment. Dynamic migration, presented by migration table, has two behaviors: nodes selection and migration sequences. Therefore, migration table's efficiency, in some sense, decides accuracy of data integrations, time of migrations, energy consumption, and performance of the whole network.

In response to the challenges embodied in the specialty of WSN, the design of mobile agent's migration table needs to adequately consider the following three aspects. Firstly, agent entity must be as small as possible to reduce migrating energy consumption. Secondly, migration table has to guarantee rational precision of data integrations to obtain sufficient sensor nodes. Thirdly, in the precondition of the second aspect, migration table needs to be short enough. In conclusion, a suitable migration table should balance the cost and precision.

Thus, in this paper, we mainly concentrate on the design of the migration table which can reduce agent's migration energy consumption while guaranteeing reliable migration methods.

Briefly, mobile agent migration has two categories: static migration list algorithm (SMLA) and dynamic migration list algorithm (DMLA). In SMLA, CIS analyzes the whole network's resources and designs migration paths that are inserted in agent entities initially. However, static SMLA cannot respond to the changes of the network timely; the changes happen when one or more links between nodes break, while, in DMLA, migration table is updated at each migration interval according to the sensibility of the network.

Without loss of generality, mobile agent migration algorithms can be illustrated by target tracking. Considering movements of the target entity, Zhao et al. proposed the information-driven-based tracking method [9]. Liu et al. proposed the collaborative in-network method to track target [10], which effectively reduced overheads and energy consumption. Kim et al. transferred the distributed tracking issue to the continuous Bayesian estimation problem and adopted the information-driven-based sensor querying scheme to select the next hop [11]. Wang et al. proposed a heuristic target localization method based on entropy [12], which is more efficient than the information-based computing method. In this paper, we will propose the dynamic migration list algorithm based on the predicting information drive (pid-DMLA) to solve the motion problems of migrations of the target entity.

Considering the condition of using multiple agents to track an application in practice, we will also propose the dynamic migration list algorithm based on the predicting information drive of multiple agents (Mpid-DMLA).

In this paper, we would propose the migration algorithms in different scenarios: single mobile agent and multiple mobile agents, which is valuable and applicable in WSNs. The rest of this paper is organized as follows. Section 3 describes the preliminary of migration. Section 4 presents single-agent-based migration algorithms, including SMLA, DMLA, and pid-DMLA. Section 5 illustrates Mpid-DMLA. Simulation results and analyses are presented in Section 6, and conclusions are drawn in Section 7.

3. Preliminary

3.1. Assumptions. The following assumptions are considered.

(1) The location of each sensor node is known.

(2) Each node has at least two neighbor nodes.

(3) The valid perception radius of sensor node is more than half the distance between two neighboring nodes.

(4) All nodes are synchronized.

(5) All sensors are deployed in a two-dimensional region.

(6) There is only one target entity in the region, and it moves uniformly and linearly.

3.2. Evaluation Indicators of the Agent Migration Algorithm. In order to evaluate the existing agent migration algorithms, we adopt the three indicators proposed by Xu et al. [7] energy consumption, network lifetime, and migration hops.

Each migration step of mobile agent is called a hop. And the energy consumption of each hop consists of three aspects: transmission consumption E_{ab} (E_{ab} represents the energy consumption of an agent migrating from node a to node b), initialization consumption E_i (E_i is constant), and data processing consumption. The network energy consumption is primarily concentrated in transmissions.

A migration is primarily composed of four parts: sending an agent, receiving an agent, data processing, and deciding the next hop. And each consumption time of the four processes is fixed. Thus, evaluating the migration time of an agent is equivalent to calculating the migration hops.

3.3. Node's Sensor Model. We adopt the following wireless transmission model [13] proposed by Rappaport:

$$z_k(t) \propto \frac{E}{\left\| X(t) - X_k \right\|^2}, \tag{1}$$

where $z_k(t)$ is a measure value of sensor k related to the location and time, E is the signal power of the target entity, $X(t)$ represents the location of the target entity in time

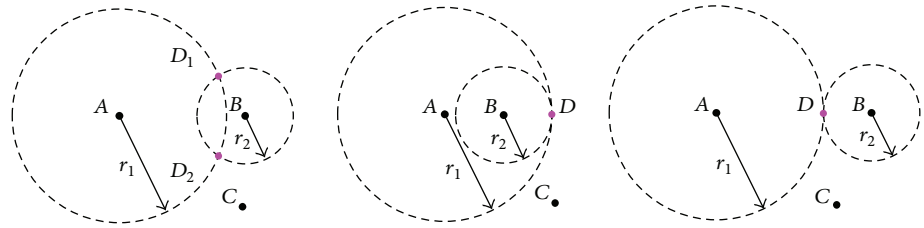


Figure 1: The target localization algorithm.

t, X_k represents the location of node k, $X(t)$ and X_k are two-dimensional vectors, and $\|X(t) - X_k\|$ represents the Euclidean distance between node k and target entity in time t.

3.4. Node's Information Collection Model. In order to quantify the data provided by sensors, we need to define the amount of information. According to (1), at any time, $z_k(t)$ is only related to $\|X(t) - X_k\|$. Thus, adopting the Gaussian model, the amount of information when the agent migrates to node k is the following:

$$I_k(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\|\widehat{X}(t)-X_k\|^2/2\sigma^2}, \tag{2}$$

where σ is the standard deviation of the amount of information, which indicates the decrease of speed of $I_k(t)$ when the distance between sensor node and target entity increases, and $\widehat{X}(t)$ represents the location of the target entity, which is calculated by the target localization algorithm in Section 3.6.

3.5. Beacon Frames. We use beacons to termly exchange information of the neighboring nodes. A beacon frame consists of the following five fields: X_k represents the location of node k; $e_k(t)$ represents the residual energy of node k in time t; $z_k(t)$ represents the sensor measurement of node k in time t; and the last two fields are sending time t_s and receiving time t_r. Beacon frames are used to provide information of location and measurement to the target localization algorithm and the agent migration algorithm. Moreover, they can be used to test the survivability of neighbor nodes.

3.6. Target Localization Algorithm. As illustrated in (2), the location of target entity is calculated by the target localization algorithm. At present, target localization algorithms mostly adopt the method of centroid weighting. And the basic idea is to make node's beacons take control of the centroid's coordinate and to reflect the general relations of the locations by using weighting factors which influence the positions of the nodes' centroids. And according to the RSSIs of the three sensors' beacons, it is easy to obtain the other node's location by the trilateration.

As shown in Figure 1, the locations of $A(x_A, y_A)$, $B(x_B, y_B)$, and $C(x_C, y_C)$ are known; and one of $D(x_D, y_D)$ needs to be tested. According to the RSSIs, we can calculate the distance of AD, BD, and CD, denoted by r_1, r_2, and r_3, respectively. Thus, we can obtain the following equation set:

$$\begin{aligned} (x_B - x_D)^2 + (y_B - y_D)^2 &= r_2^2, \\ (x_A - x_D)^2 + (y_A - y_D)^2 &= r_1^2. \end{aligned} \tag{3}$$

Apparently, the distance of A and B is less than or equal to $r_1 + r_2$. So (3) has two real solutions that are identical or not, which correspond to the coordinates of the two nodes D_1 and D_2 that are identical or not. Select the one that is nearer to C as the approximate location of D, denoted as (x_1, y_1). In the same way, we can obtain the other two approximate locations of D, denoted as (x_2, y_2) and (x_3, y_3). Therefore, the coordinate of D is (x_D, y_D), which can be calculated by the following:

$$\begin{aligned} x_D &= \frac{x_1/(r_1 + r_2) + x_2/(r_2 + r_3) + x_3/(r_3 + r_1)}{1/(r_1 + r_2) + 1/(r_2 + r_3) + 1/(r_3 + r_1)}, \\ y_D &= \frac{y_1/(r_1 + r_2) + y_2/(r_2 + r_3) + y_3/(r_3 + r_1)}{1/(r_1 + r_2) + 1/(r_2 + r_3) + 1/(r_3 + r_1)}. \end{aligned} \tag{4}$$

As a result, according to the previous target localization algorithm, we can obtain $\hat{x}(t)$.

4. Migration Algorithms Based on Single Mobile Agent

This section would propose SMLA and DMLA algorithms from static and dynamic aspects, respectively. And it would then propose pid-DMLA based on disadvantages of SMLA and DMLA.

4.1. SMLA. In SMLA algorithm, we use static information. Before distributing mobile agents, the CIS collects by communicating with other nodes, and it then generates a migration plan. In this migration plan, information collection values of all nodes in the network decrease. The specific steps of SMLA are as follows.

Algorithm 1 (SMLA).

(1) At time $t = 0$, the node that has first detected information of target entity is denoted as cluster head; it generates a mobile agent.

(2) At time $t = 0$, calculate location information of target entity, $\widehat{X}(0)$, according to the target location algorithm.

Calculate the amount of information collection, I_0, according to (2), and set $SumArray = I_0$.

Select the next hop, according to the predetermined migration sequence.

(3) At time t, when agent reaches node k, the location of target entity is $\widehat{X}(t)$ according to the target location algorithm.

Calculate the amount of information collection of node k, I_k, and update $SumArray = SumArray + I_k$.

If $SumArray$ is greater than the threshold, then go to step 4; otherwise, continue to migrate to the next hop in the migration sequence, update time $t = t + 1$, and then repeat step 3.

(4) Return to cluster head to process data.

Though, sometimes, SMLA can gain the ideal migration sequence, there are many drawbacks in practical applications. Firstly, since SMLA uses static information, sensor's transmission distance must be large enough to ensure that CIS is within one-hop range. So we need to adopt the nodes with greater power in communication capabilities. And using these nodes will inevitably result in relatively higher energy consumption. Secondly, since SMLA is a static algorithm, the static information, which is acquired before migration, lack timeliness and cannot reflect the changes of the network in the process of migrating. For example, that some nodes suddenly lose efficacy can result in the failure of the deployment of SMLA [9]. Moreover, most of WSNs are highly dynamic, so SMLA has some limitations in WSNs.

4.2. DMLA. SMLA is a centralized algorithm, and it has already decided the migration sequence before distributing mobile agents. Because WSN is a dynamic and distributed network, static routing information exposes a lot of problems in practical applications. Thus, DMLA is proposed. DMLA dynamically selects the next hop in the process of migration based on characteristics of the network.

4.2.1. The Migration Model of Mobile Agent in DMLA. In DMLA, mobile agents decide the nodes that they need to migrate to. In the surviving neighbor nodes, not all of the nodes can provide information affecting the final result. Therefore, the purpose of mobile agent migration algorithm is to produce a set of optimal subsets to determine the optimum migration sequence.

In the target entity tracking, we construct rules to choose the next hop as follows: mobile agent, which is located in node k, always looks for the next hop with minimal overheads. As the selection of the next hop is a method of balancing overheads and profit, we define the overheads function, when mobile agent moves from node i to neighbor node j at time t, as follows:

$$C_{ij}(t) = a\frac{E_{ij}}{E_{\max}} + b\left(1 - \frac{I_j(t)}{I_{\max}}\right) + \sqrt{1-a-b}\left(1 - \frac{e_j(t)}{e_{\max}}\right). \tag{5}$$

Equation (5) consists of three parts including energy consumption, information collection, and residual energy, where E_{ij} represents transmission energy consumption when the agent migrates directly from node i to node j; $I_j(t)$ represents the amount of information collection of node j at time t; $e_j(t)$ represents the residual energy of node j; $E_{\max}$ represents the maximum energy consumption of migration between two nodes, which is determined by the maximum transmission distance; $I_{\max}$ represents the maximum amount of information collection; $e_{\max}$ represents the initial energy of sensor node and it is a constant value; and a and b are the balance coefficients: $0 \le a$ and $b \le 1$.

Define the transmission energy consumption of the agent migrating from node i to node j as follows:

$$E_{ij} = P_{ij} \cdot T_{\text{trans}} = \alpha P_{r\text{-th}} \cdot \left\|X_i - X_j\right\|^2 \cdot T_{\text{trans}}, \tag{6}$$

where P_{ij} represents the transmission power of sensor nodes; $\|X_i - X_j\|$ represents the distance between node i and node j; T_{trans}, a constant, represents the time of transmitting the mobile agent; $\alpha = 4\pi^2/\lambda^2 G_1 G_2$; G_1, G_2 are all of the antenna gain factors (set them as 1); λ represents the signal wavelength; and $P_{r\text{-th}}$ represents the receiving threshold.

From (2), (5), and (6), we can gain

$$C_{ij}(t) = a\frac{\left\|X_i - X_j\right\|^2}{d_{\max}^2} + b\frac{\left\|\widehat{X}(t) - X_j\right\|^2}{dt_{\max}^2} + \sqrt{1-a-b}\left(1 - \frac{e_j(t)}{e_{\max}}\right), \tag{7}$$

where $dt_{\max}$ represents the maximum possible distance between two nodes.

Equation (7) can be rewritten as follows:

$$j^* = \arg\min\left\{a\frac{\left\|X_i - X_j\right\|^2}{d_{\max}^2} + b\frac{\left\|\widehat{X}(t) - X_j\right\|^2}{dt_{\max}^2} + \sqrt{1-a-b}\left(1 - \frac{e_j(t)}{e_{\max}}\right)\right\}, \quad j \in N_i, \tag{8}$$

where N_i represents a group of neighbors of node i; node $j^* \in N_i$ represents the next hop; and $j^* \in N_i$ can ensure less energy consumption, stronger sensing capabilities, and more residual energy.

In summary, the total cost in the migration process is

$$\text{Cost} = \sum_{p=0}^{\text{HOP}_S - 1} C_{i_p i_{p+1}}, \quad \text{s.t. } SumArray = \sum_{p=0}^{\text{HOP}_S} I_p > \text{threshold}, \tag{9}$$

where p represents the number of nodes migrated through HOP_S represents the total number of migration hops. In the practical migration, once mobile agent gains enough information, and it will terminate migration and return to the CIS.

4.2.2. The Specific Migration Procedure of DMLA. The mobile agent migrates to the alive node which has not been reached and with the minimal migration overheads according to (7). In this way, we can gain a table close to the optimal migration. If all nodes have been reached, mobile agent will return the node that distributes it. And when the mobile agent has acquired enough information, it will return to the CIS.

Since each node periodically receives beacon frames from neighbor nodes, DMLA can always use the updated local variables to determine the next hop. As only using local variables, sensor nodes can avoid the node which has a greater transmission distance, so part of the transmission energy can be saved. In addition, when some nodes suddenly lose efficacy, the whole migration process will not make mistakes. So DMLA algorithm is a fault-tolerant algorithm. The specific migration steps of DMLA are as follows.

Algorithm 2 (DMLA).

(1) At time $t = 0$, the node that has first detected information of target entity is denoted as cluster head;

it generates a mobile agent.

(2) At time $t = 0$, calculate location information of target entity, $\widehat{X}(0)$, according to the target location algorithm.

Calculate the amount of information collection, I_0, according to (2), and set $SumArray = I_0$.

Select the next hop, according to (8).

(3) At time t, when agent reaches node k, the location of target entity is $\widehat{X}(t)$ according to the target location algorithm.

Calculate the amount of information collection of node k, I_k, and update $SumArray = SumArray + I_k$.

If $SumArray$ is greater than the threshold, then go to step 4; otherwise, continue to migrate to the next hop according to (8), update time $t = t+1$, and then repeat step 3.

(4) Return to cluster head to process data.

4.3. pid-DMLA

4.3.1. The Proposition of pid-DMLA. In the cooperation processing applications, the movement of target entity is one of the important factors that affect the final result [14–19]. DMLA algorithm does not take the direction of the movement of target entity into account. So in some cases, it cannot select the optimum next hop. For example, in Figure 2, at time t, the mobile agent is located in node A. Nodes B and C are neighbors of A, and they are equidistant to A. According to DMLA, mobile agent calculates node B as the next hop. This is because node B is nearer to the target entity than node C. Then, at time $t + 1$, the mobile agent is located in node B. But the target entity moves to a position closer to node C. Thus, DMLA, on the contrary, selects the node which is farther from the target and has smaller amount of information collection and measurement value

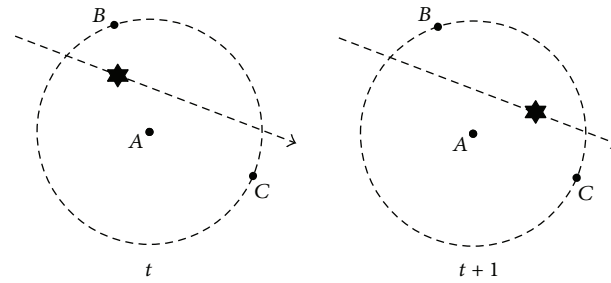


Figure 2: The movement of the target entity.

as the next hop. Therefore, when designing agent migration algorithms, we need to focus on predicting the movement of the target entity and selecting the node with the maximum information. And we will propose the pid-DMLA algorithm to solve the problems of the movement of the target entity in the migration process.

4.3.2. Agent Migration Model in pid-DMLA. Suppose that the moving direction and speed of the target entity do not change. Then, the change of the target entity's displacement is a constant, at each identical time interval, as follows:

$$X(t+1) - X(t) = X(t) - X(t-1). \tag{10}$$

Equivalently,

$$X(t+1) = 2X(t) - X(t-1). \tag{11}$$

As illustrated in (11), the position information $X(t + 1)$ at time $t + 1$ can be deduced from the positions at time $t - 1$ and time t. In this way, the location information $\widehat{X}(t-1)$ and $\widehat{X}(t)$ can be calculated by target location algorithm; and, according to (11), the position of target entity, $\widehat{X}(t+1)$, can be estimated at time $t + 1$ as follows:

$$\widehat{X}(t+1) = 2\widehat{X}(t) - \widehat{X}(t-1). \tag{12}$$

Update the overhead function in (7), and get

$$C_{ij}(t) = a\frac{\left\|X_i - X_j\right\|^2}{d_{\max}^2} + b\frac{\left\|\widehat{X}(t+1) - X_j\right\|^2}{dt_{\max}^2} + \sqrt{1-a-b}\left(1 - \frac{e_j(t)}{e_{\max}}\right), \tag{13}$$

where $\|\widehat{X}(t + 1) - X_j\|$ represents the distance between node j and the target entity at time $t + 1$.

And the next hop is

$$j^* = \arg\min\left\{a\frac{\left\|X_i - X_j\right\|^2}{d_{\max}^2} + b\frac{\left\|\widehat{X}(t+1) - X_j\right\|^2}{dt_{\max}^2} + \sqrt{1-a-b}\left(1 - \frac{e_j(t)}{e_{\max}}\right)\right\}, \quad j \in N_i. \tag{14}$$

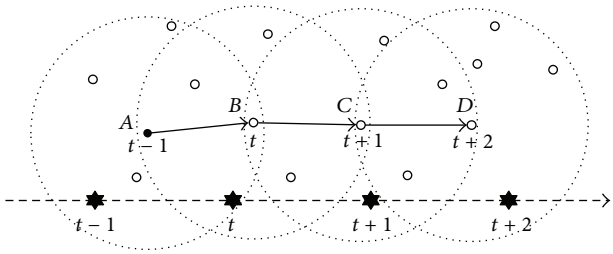


Figure 3: The migration of pid-DMLA.

At time $t = 0$, since the position of the target entity at the previous time is unknown, the amount of information collection can be calculated by the DMLA. At time $t > 0$, mobile agent migrates to the node closer to the estimated position of the target entity. So it can obtain larger sensing measurement and information collection, which is the core of pid-DMLA.

4.3.3. The Specific Migration Procedure of pid-DMLA. The process of mobile agent migration of pid-DMLA is shown in Figure 3. In a period of time, the target entity moves along the dotted line, and mobile agent migrates from node A to D through B and C. As shown in Figure 3, mobile agent always migrates to the node nearest to the next position of the target entity.

The specific migration procedure of pid-DMLA is as follows.

Algorithm 3 (pid-DMLA).

(1) At time $t = 0$, the node that has first detected information of target entity is denoted as cluster head;

it generates a mobile agent.

(2) At time $t = 0$, calculate location information of target entity, $\widehat{X}(0)$, according to the target location algorithm.

Calculate the amount of information collection, I_0, according to (2), and set $SumArray = I_0$.

Select the next hop, according to (14).

(3) At time t, when agent reaches node k, the location of target entity is $\widehat{X}(t)$ according to the target location algorithm.

Calculate the amount of information collection of node k, I_k, and update $SumArray = SumArray + I_k$.

If $SumArray$ is greater than the threshold, then go to step 4; otherwise, continue to migrate to the next hop according to (14), update time $t = t + 1$, and then repeat step 3.

(4) Return to cluster head to process data.

5. Migration Algorithm Based on Multiple Mobile Agents

5.1. The Architecture of Mpid-DMLA. Many researchers have proposed and realized feasible ways to organize the migration of a single mobile agent, such as Agilla program [18] and SensorWare program [19]. Consider the functional requirements of these programs. Firstly, they have not formed an architecture to manage the agent migration mechanisms in WSN. Secondly, they lack migration algorithms when a tracking application needs the collaboration of multiple agents. So when dealing with multiple agents' migration, we need a specific agent to manage the applications.

Therefore, we will propose Mpid-DMLA to solve this problem.

Suppose that multiple target entity applications are deployed in an area of sensors; each application is responsible for monitoring a part of the area and composed of one or more mobile agents. Therefore we have the following definitions.

Definition 4 (code segment application). A centralized program which consists of multiple mobile agents deployed in the WSN to accomplish a specific task is called centralized code segment application of multiple agents, which is called code segment application for short.

Code segment applications are associated with sensing data that need to be evaluated, and they need to cooperate and interact based on the need of programs. On the other hand, according to the Deluge & Mate model [20], code segment applications need to redeploy themselves so that they can be distributed by the CIS.

To track targets, code segment applications require the following four functions [21].

(1) *Sensing.* Sensor nodes are distributed around the target node, and they test and collect information of the surrounding environment.

(2) *Processing Information.* After acquiring sensor information, most of the nodes send it to the CIS for processing, while cluster heads will directly process information.

(3) *Communication.* In order to maintain the application, sensor nodes need to keep exchanging information with other nodes or CIS.

(4) *Migration.* Mobile agents migrate based on the need of code segment application.

Constructing the architecture of multiple mobile agents also requires the following definitions.

Definition 5 (main component agent). The main component agent is responsible for managing migration information and activity cycle of all component agents, processing sensing data of other agents, and collecting and exchanging the information of subcomponent agents. Each code segment application has one and only main component agent.

Definition 6 (subcomponent agent). The subcomponent agent is used to obtain the data of nodes and report them to the main component agent. Each code segment application has one or more subcomponent agents.

Definition 7 (communication component agent). The communication component agent is used to collect and exchange the entity information of WSN by migrating. Each code segment application has one or more communication component agents.

Definition 8 (middleware software). The middleware software, located between the physical and the operating system layers, is the software entity running in mobile agents.

5.2. The Migration Model of Mobile Agent in Mpid-DMLA. When a code segment application changes monitoring positions, all of the mobile agents in this application need to migrate together to reconstruct a new code segment application. In Agilla, the main component agent controls the entire process by sending the migration sequence, while other mobile agents can communicate by accessing the independent tuple space [22]. And the engine virtual machine works as the kernel to provide services and controls all of the concurrent execution of all of the agents in a node. However, with the increase of the mobile agents, the possibilities of the loss of packet and the failure of migration will increase, too. Suppose that a code segment application is composed of N mobile agents. Define P as the failure coefficient of a single agent. And then we can obtain the success coefficient of migration, in the entire application, as follows:

$$P_{\text{success}} = (1 - P)^N. \tag{15}$$

Communication overheads, based on pid-DMLA, are given as follows:

$$C_{ij}(t) = \left[a\frac{\left\|X_i - X_j\right\|^2}{d_{\max}^2} + b\frac{\left\|\widehat{X}(t+1) - X_j\right\|^2}{dt_{\max}^2} + \sqrt{1-a-b}\left(1 - \frac{e_j(t)}{e_{\max}}\right)\right] \times \left(P_{\text{success}}\right)^{-1}. \tag{16}$$

And the next hop is

$$j^* = \arg\min\left\{\left[a\frac{\left\|X_i - X_j\right\|^2}{d_{\max}^2} + b\frac{\left\|\widehat{X}(t+1) - X_j\right\|^2}{dt_{\max}^2} + \sqrt{1-a-b}\left(1 - \frac{e_j(t)}{e_{\max}}\right)\right] \times \left(P_{\text{success}}\right)^{-1}\right\}. \tag{17}$$

5.3. The Specific Migration Procedure of Mpid-DMLA. In the construction of the target entity tracking, we assume that subcomponent agent is a small program of lightweight code which is suitable for migration. And it can independently access the target entity's location. As shown in Figure 4, hollow circles represent sensor nodes which are distributed in the sensing region; code segment is composed of one main component agent (solid circle), four subcomponent agents (solid squares) and one communication component agent (solid triangle). Each agent runs on one node, executing code segment applications. The migration of code segment application means the migration of the six mobile agents, which is the migration of multiple mobile agents [19]. The following algorithm can be used to describe the above-mentioned procedure.

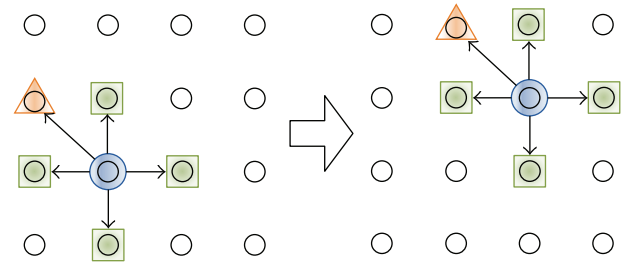


Figure 4: Migration based on predicting information drive of multiple mobile agents.

Algorithm 9 (Mpid-DMLA).

(1) A main component agent, a communication component agent, and several subcomponent agents are deployed in different sensor nodes.

(2) Sensor nodes are monitoring around the target entity.

The node that has first detected information of target entity is denoted as cluster head; the main component agent then migrates to it.

(3) At time $t = 0$, calculate location information of target entity, $\widehat{X}(0)$, according to the target location algorithm.

Calculate the amount of information collection, I_0, according to (2).

Define the threshold of the amount of information as *Threshold_UP*, and let *Threshold_UP* $= I_0$.

The main component agent generates new communication component agent and transfers I_0 and *Threshold_UP* to it.

Then, the communication component agent sets to migrate.

(4) At time t, when agent reaches node k, the location of target entity is $\widehat{X}(t)$ according to the target location algorithm.

Calculate the amount of information collection of node k, I_k, and update *Threshold_UP* $=$ *Threshold_UP* $+ I_k$.

If *Threshold_UP* is greater than the threshold, then go to step 5; otherwise, select the next hop for migration according to (17), update time $t = t + 1$, and then repeat step 3.

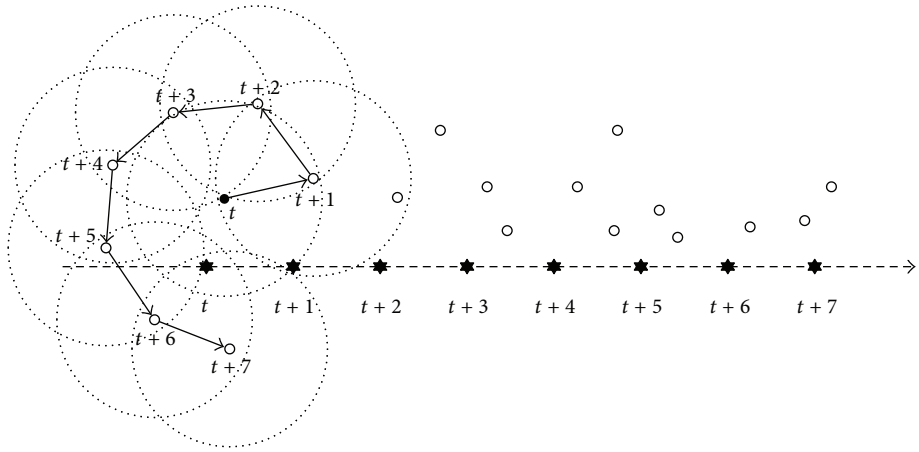


Figure 5: The migration process of SMLA.

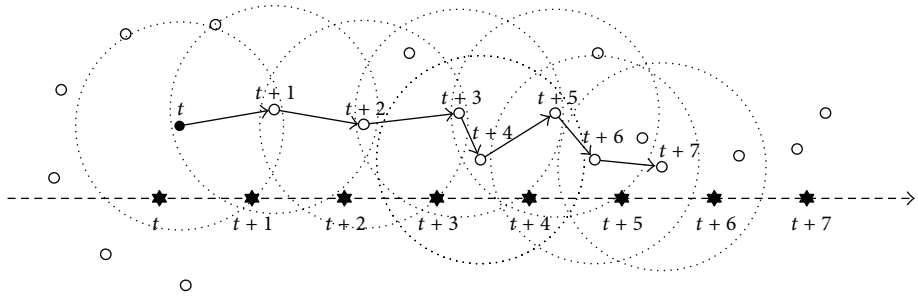


Figure 6: The migration process of DMLA.

(5) Communication component agent stores the information of the migration path: $(\{I_0\ (t = 0), I_1\ (t = 1), \ldots, I_k\ (t = k)\}, \{X\ (t = 0), X\ (t = 1), \ldots, X\ (t = k)\})$ and other relevant information.

And it interacts with the main component agent.

(6) The main component agent generates the migration sequence $\{N_1, N_2, \ldots, N_k\}$, according to location of target entity.

Send it to each subcomponent agent (this step is optional for different applications).

(7) The main component agent sends complete instructions to the lower-layered middleware software.

Method functions which are predefined in the middleware software recycle the communication component agent and the subcomponent agents.

(8) The main component agent goes to the destination node based on the migration sequence.

If the migration fails, send failure information to CIS to terminate this migration; otherwise, generate new communication component agent and subcomponent agents, and this migration ends.

6. Simulation and Analyses

6.1. The Effect of Single Mobile Agent in Target Tracking. Before the simulation experiments, we firstly use target tracking as a sample to demonstrate the migration effect of a single mobile agent of different migration algorithms. As shown in Figures 5, 6, and 7, suppose that a car is running from left to right through the area covered by WSN. The small circle represents the sensor node; solid small circle represents the cluster head, which distributes agents; large dotted circle represents the communication range of sensor node, and each node selects the next hop only within its communication range. The above-mentioned figures illustrate the specific locations of the car and agent from time t to time $t + 7$.

The migration process of SMLA is shown in Figure 5. At time t, CIS distributes the mobile agent to the node which is the nearest to the target entity, according to the predetermined migration sequence. When the object is stationary or its motion can be estimated, this static approach is very suitable. However, it loses efficacy in some conditions. For example, if the object remains in the location at time $t+1$, then the migration sequence will be the positions that are around the target entity. But once the target entity starts to move not according to the predetermination, SMLA will fail.

DMLA can, in some sense, optimize the performance of SMLA. As shown in Figure 6, mobile agent tracks the movement of the target entity. However, the movement of

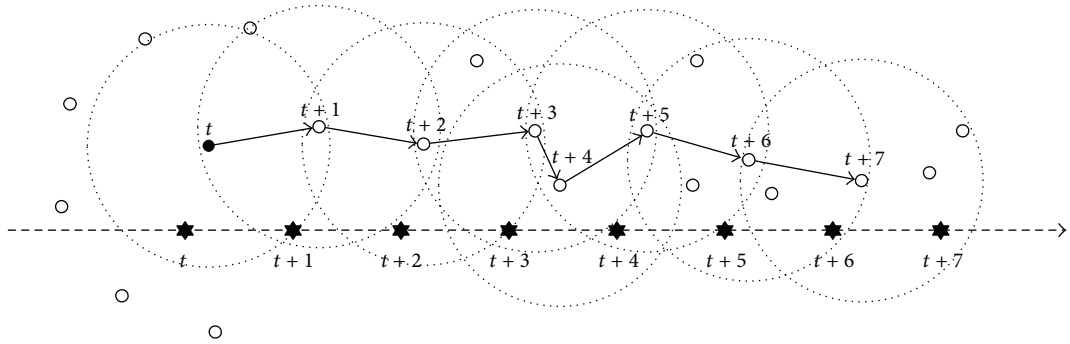


Figure 7: The migration process of pid-DMLA.

Table 1: Basic simulation parameters of single mobile agent migration algorithms.

Network area	500 × 500 m
Number of nodes	200 or 800
Node's initial energy	50 J
Node's sensing radius	5 m
Energy consumption for data processing	0.3 J
Energy consumption for sending beacon frames	0.1 J
The interval of node broadcasting beacon frames	0.1 s
The speed of the target entity	0~50 m/s
The threshold of the amount of information	18
The size of a mobile agent	800 bits
Simulation time	100 s

the target entity is unpredictable, so it is not accurate to decide migration path only by the amount of information, which is, however, related to the distance to the target entity.

The pid-DMLA can predict movement information of target entity dynamically, which is a better solution to such problems. As shown in Figure 7, at time $t + 5$, mobile agent selects the next hop according to the predicted information, which is different from DMLA. In brief, pid-DMLA rebuilds migration sequence based on predicting the movement of the target entity to achieve the purpose of tracking.

6.2. Performance Analysis on the Single Mobile Agent Migration Algorithms. We use simulation software, NS2, to evaluate the three single mobile agent migration algorithms' performances. The default simulation parameters are shown in Table 1.

Suppose that once the agent migrates one hop, all sensor nodes in the network consume 0.2 J. For SMLA, the transmission distance is set to 30 m, while, for DMLA and pid-DMLA, the transmission distance is set to be 10 m. Therefore, the consumption of transmission, for SMLA, is three times more than that of sending. The reason for this setting is that SMLA needs bigger transmission range to collect information. The direction and size of movement of the target entity are constant, denoted as v (m/s). Once the target entity reaches the edge of the sensing area, it reverses direction immediately, which ensures its being always within the sensing area.

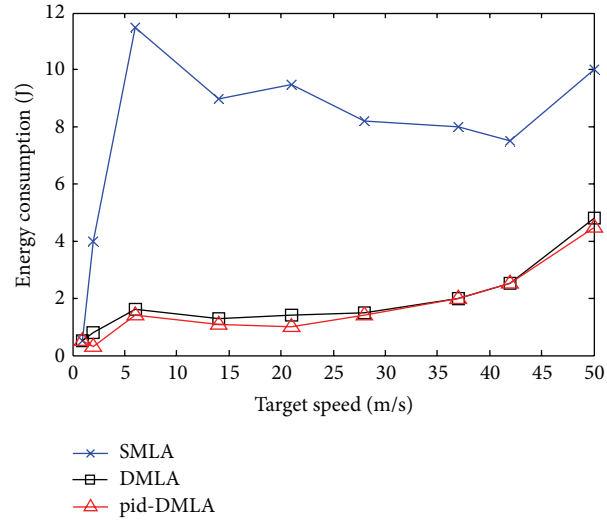


Figure 8: Energy consumption of the single agent migration algorithms in Scenario 1.

Scenario 1. Set the number of nodes as 200, the target entity speed as 0~50 m/s, and the simulation time as 100 s, which can guarantee a complete migration. The simulation results are shown in Figures 8 and 9. It is obvious that DMLA and pid-DMLA consume less energy than SMLA in the whole network. This is because the communication cost of SMLA is higher than that of DMLA. But with the increasing speed of the target entity, the effect of SMLA algorithm in tracking the target entity becomes less and less obvious; therefore, migration may terminate, and the energy consumption, on the contrary, becomes slower. On the other hand, when the speed of target entity is very low, there are no significant differences among the three migration schemes.

Scenario 2. Set the speed of target entity as 5 m/s and the number of nodes as 800. Generally, the coverage and the number of nodes are positively correlated. The simulation results are shown in Figures 10 and 11. It is clear that pid-DMLA has the lowest energy consumption when the number of nodes is more than 100. This is because with the increase of the nodes the choices of predicting the next hop become

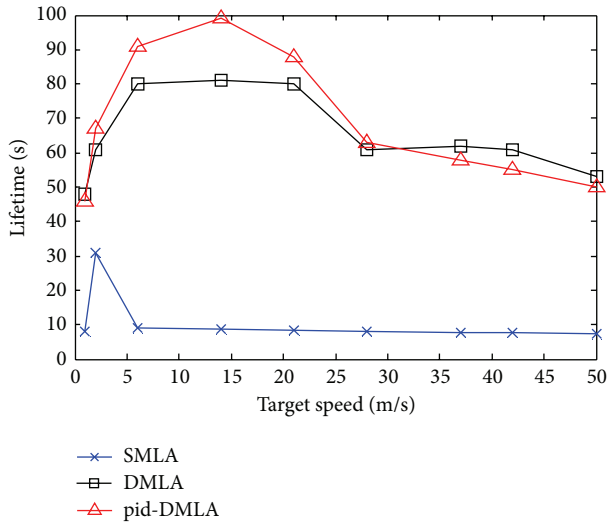


Figure 9: Lifetime of the single agent migration algorithms in Scenario 1.

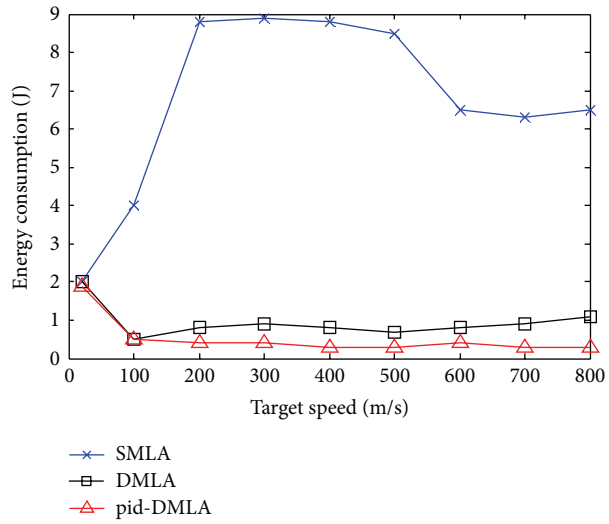


Figure 10: Energy consumption of the single agent migration algorithms in Scenario 2.

more; as a result, the overheads become less [23]. When the number of nodes is less than 50, the performance of SMLA is better than those of the other two. This is because with less nodes static and dynamic migration algorithms share the more similar migration path; however, the migration table has been predetermined in SMLA; thus, it, on the contrary, has the least overheads. On the other hand, when the density of nodes is high, there would be more neighbors and more migration choices. Thus, DMLA and pid-DMLA outperform SMLA.

6.3. Multiple Mobile Agents Migration Algorithm Performance Analysis. Before the assessment of Mpid-DMLA, we need to consider two problems [24, 25]. The first is the effect of code

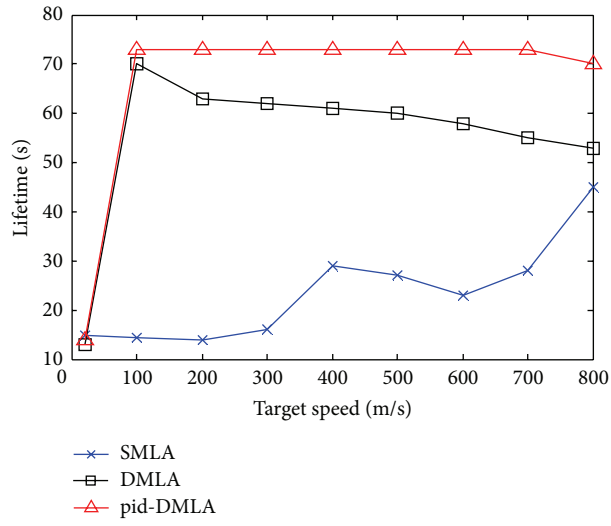


Figure 11: Lifetime of the single agent migration algorithms in Scenario 2.

segment applications on the amount of communication in the migration process. To ensure the comprehensiveness of the evaluation, set the number of communication component agents as 1 and the number of subcomponent agents as 5, 10, or 20, and evaluate Mpid-DMLA based on code segments with different scales. The second is the effect of different environment and running platform in practical operation on the amount of information and the success rate of migration. Therefore, we only consider the relation of the number of communication and energy instead of the influence of the agent's size.

The performance of the migration of mobile agent in Agilla system has been evaluated in [19]. First, Agilla system is not designed to support multiple mobile agents specifically, which means that it is not aware of the number of the agents in the code segment application. And the system's overheads are related to the size of mobile agent. So, in Agilla, all kinds of agents are considered to be identical, and there is no discrimination between the main and communication component agents. However, this does not affect the simulation results because all agents are uniformly assumed in the simulation in this paper.

The purpose of multiple mobile agents migration is to complete the entire application migration with limited energy. The default simulation parameters are shown in Table 2.

And when main component agent recycles and sends mobile agents, suppose that the energy consumption of the node where main component agent stays is neglected, while that of other nodes is 0.05 J.

Figures 12, 13, and 14 illustrate the relation between the number of application code segments and the success rate of one hop migration, when the numbers of subcomponent agents are 5, 10, and 15, respectively. The success rate of Mpid-DMLA is higher, when the number of subcomponent agents is 5, than those when the numbers of subcomponent agents

Table 2: Basic simulation algorithm parameters of a single mobile agent migration.

Network area	100×100 m
Number of nodes	7×7
Number of main component agents per code segment application	1
Number of subcomponent agents per code segment application	5, 10, or 15
Number of communication component agents per code segment application	1
Node's initial energy	2 J
Energy consumption of a node when an agent reaches it	0.05 J
Interaction consumption when an agent migrates	0.1 J

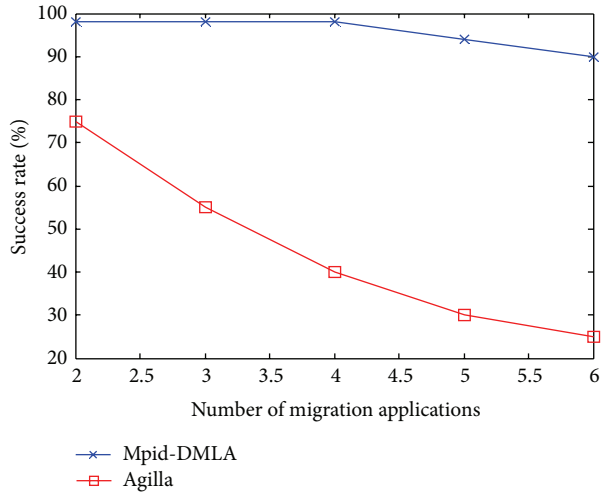


Figure 12: The relation between the number of code segment applications and the success rate of migration (I).

are 10 and 20. It is obvious that the successful rate of Mpid-DMLA is generally not less than 50%, which is much larger than that of Agilla.

Moreover, Figure 15 demonstrates the relationship between the migration hops and the amount of communication, when the subcomponent agents are 5. As shown in Figure 15, with the same migration hops, the amount of communication of Mpid-DMLA is much less than that of Agilla. As the number of data packet reflects the amount of energy consumption, Mpid-DMLA consumes less energy than Agilla.

Although the simulation results show that Mpid-DMLA is a suitable migration scheme for multiple mobile agents, it bases on some specific assumptions. In general, this scheme still encounters the following three issues.

(1) When the main component agent loses efficacy in the process of migration, code segment applications cannot be rebuilt.

(2) When subcomponent agents lose efficacy after deployment, code segment applications will keep waiting because the applications cannot be recycled.

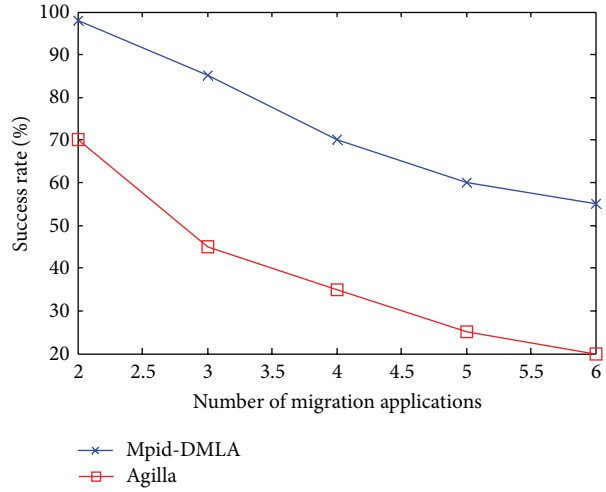


Figure 13: The relation between the number of code segment applications and the success rate of migration (II).

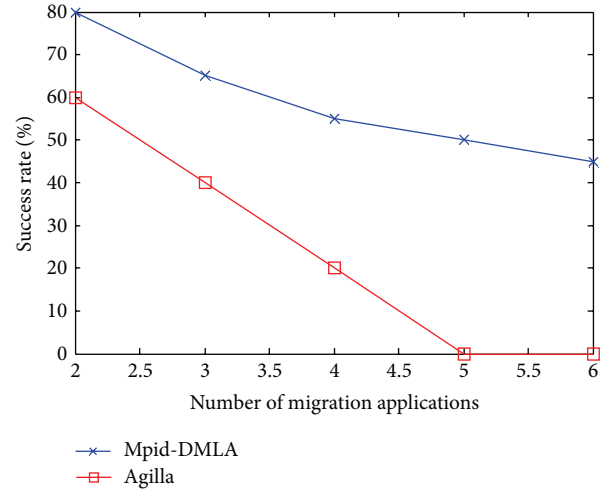


Figure 14: The relation between the number of code segment applications and the success rate of migration (III).

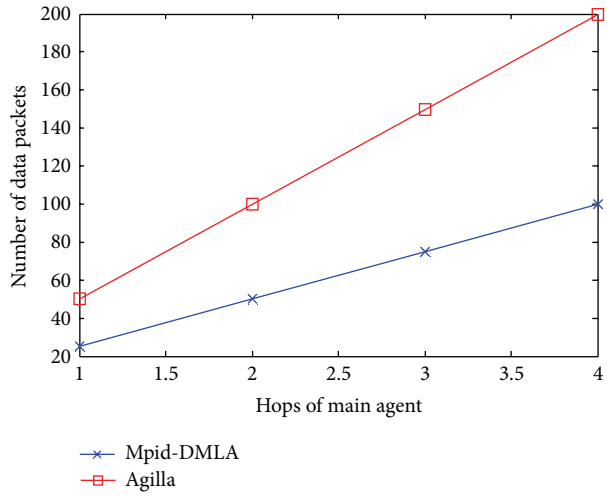


Figure 15: The relation between the migration hops and the amount of communication.

(3) If the main and the subcomponent agents, which lose efficacy in issue 1 or 2, are left in the nodes without being recycled, then other mobile agents cannot reach these nodes.

Therefore, in designing middleware software, it is necessary to add the feedback mechanism of main component agent, and it needs to periodically return ACK information to the higher layer of the middleware software. When the middleware finds that the main component agent loses efficacy, it needs to distribute a new one or a specialized independent agent to recycle the useless agents in the whole network.

In Agilla, the method of recycling the useless agents is by adding the unified interface into all of the agents. Removing all of the mobile agents will affect the function of code segment applications. Therefore, developing the more appropriate middleware software is the future work, which is also related to the development of the better migration strategies [26].

7. Conclusion

As big data is the intensification of information architecture while distributed computing is the intensification of infrastructure and resource, distributed computing is an appropriate method to deal with big data. In this paper, we have firstly illustrated the mobile agent model in the distributed computing. And it led to the two considerable factors of migration: the selection of nodes and of the sequence of them. Then, we have proposed the assumptions of the migration algorithms. Moreover, we have designed the node's sensor model and information collection model. Furthermore, we have formulated the beacon frames of mobile agent for communication. Generally, mobile agent migration algorithms can be transferred as the target tracking issue. Thus, we have described a simple target localization algorithm applicable for migration. On this basis, we have generalized the whole migration process of the mobile agent.

Considering the single mobile agent, we have proposed the static SMLA and the dynamic DMLA algorithms. After analyzing the drawbacks of SMLA and DMLA, we then proposed the pid-DMLA algorithm. Section 4 has demonstrated the modeling procedure of the abovementioned three algorithms; Section 6 has emulated and evaluated their performances and studied their advantages and disadvantages in different scenarios.

Considering the multiple mobile agents applied in practical target tracking, we have also put forward the Mpid-DMLA algorithm. In the process of mobile agent component constructing code segment applications, we defined three mobile agents with different functions to cooperate with each other to accomplish the migration. Section 5 illustrated the architecture, modeling procedure, and basic process of Mpid-DMLA; Section 6 compared it with Agilla and evaluated its advantages and disadvantages.

Acknowledgments

The paper is sponsored by the National Natural Science Foundation of China (61170065, 61171053, 61003039, 61100199, 61003236, 61202355, 61201163, and 61373138), the Natural Science Foundation of Jiangsu Province (BK2011755, BK2011072, and BK2012436), the Scientific & Technological Support Project of Jiangsu Province (BE2012183 and BE2012755), the Natural Science Key Fund for Colleges and Universities in Jiangsu Province (12KJA520002), the Scientific Research & Industry Promotion Project for Higher Education Institutions (JHB2012-7), the Postdoctoral Foundation (2012M511753, 1101011B, 1202034C, and 2013T60536), the Doctoral Fund of Ministry of Education of China (20113223110002 and 20123223120006), the Fund of Nanjing University of Posts and Telecommunications (NY212047), and the project funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (yx002001).

References

[1] Research Trends special issue on big data, http://www.research-trends.com/wp-content/uploads/2012/09/Research_Trends_Issue30.pdf, 2012.

[2] L. Ismail and D. Hagimont, "A performance evaluation of the mobile agent paradigm," *ACM SIGPLAN Notices*, vol. 34, no. 10, pp. 306–313, 1999.

[3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, September 2002.

[4] IVY-A sensor network infrastructure for the college of engineering, http://www-bsac.eecs.berkeley.edu/projects/ivy, 2008.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[6] Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang, "Location tracking in a wireless sensor network by mobile agents and its data fusion strategies," *Computer Journal*, vol. 47, no. 4, pp. 448–460, 2004.

[7] Y. Xu and H. Qi, "Mobile agent migration modeling and design for target tracking in wireless sensor networks," *Ad Hoc Networks*, vol. 6, no. 1, pp. 1–16, 2008.

[8] Y. Xu and H. Qi, "Distributed computing paradigms for collaborative signal and information processing in sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 8, pp. 945–959, 2004.

[9] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[10] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *Eurasip Journal on Applied Signal Processing*, vol. 2003, no. 4, pp. 378–391, 2003.

[11] J.-W. Kim, J.-S. In, K. Hur, J.-W. Kim, and D.-S. Eom, "An intelligent agent-based routing structure for mobile sinks in WSNs," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2310–2316, 2010.

[12] H. Wang, G. Pottie, K. Yao, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proceedings*

of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04), pp. 36–45, ACM Press, April 2004.

[13] T. S. Rappaport, *Wireless Communications: Principles and Practice*, IEEE Press, 1996.

[14] H. Mousannif and I. Khalil, "Cooperative data management in wireless sensor networks," *International Journal of Computational Intelligence Systems*, vol. 5, no. 3, pp. 403–412, 2012.

[15] N. Labraoui, M. Guerroui, M. Aliouat, and T. Zia, "Data aggregation security challenge in wireless sensor networks: a survey," *Ad-Hoc and Sensor Wireless Networks*, vol. 12, no. 3-4, pp. 295–324, 2011.

[16] M. Chen, L. T. Yang, T. Kwon, L. Zhou, and M. Jo, "Itinerary planning for energy-efficient agent communications in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3290–3299, 2011.

[17] Q. Wu, N. S. V. Rao, J. Barhen et al., "On computing mobile agent routes for data fusion in distributed sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 6, pp. 740–753, 2004.

[18] C.-L. Fok, G.-C. Roman, and C. Lu, "Mobile agent middleware for sensor networks: an application case study," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 382–387, April 2005.

[19] A. Boulis, C. Han, and M. B. Srivastava, "Design and implementation of a framework for efficient and programmable sensor networks," in *Proceedings of the 1st international conference on Mobile systems, applications and services*, ACM, San Francisco, Calif, USA, 2003.

[20] P. Levis and D. Culler, "Mate: a tiny virtual machine for sensor networks," *ACM Sigplan Notices*, vol. 37, no. 10, pp. 85–95, 2002.

[21] S. Suenaga, N. Yoshioka, and S. Honiden, "Group migration by mobile agents in wireless sensor networks," *Computer Journal*, vol. 54, no. 3, pp. 345–355, 2011.

[22] P. Bellavista, Y. Cai, and T. Magedanz, "Recent advances in mobile middleware for wireless systems and services," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 267–269, 2011.

[23] Z. Zhirou and C. Wengang, "Model and optimization of mobile agent's migration in grid," in *Proceedings of the 2nd International Conference on Bio-Inspired Computing: Theories and Applications (BICTA '07)*, pp. 127–130, September 2007.

[24] J.-P. Zhang, M. Jun, J. Yang, and L.-L. Cheng, "An improved migration strategy of mobile agent," in *Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '07)*, pp. 577–581, August 2007.

[25] Y. Lee and K. Kim, "Optimal migration path searching using path adjustment and reassignment for mobile agent," in *Proceedings of the 4th International Conference on Networked Computing and Advanced Information Management (NCM '08)*, pp. 564–569, Gyeongju, South Korea, September 2008.

[26] M. Tentori, M. Rodríguez, and J. Favela Vara, "An agent-based middleware for the design of activity-aware applications," *IEEE Intelligent Systems*, vol. 26, no. 3, pp. 15–23, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 834704, 9 pages
http://dx.doi.org/10.1155/2013/834704

Research Article

Uncertainty-Aware Sensor Deployment Strategy in Mixed Wireless Sensor Networks

Yan Chang,[1] Shukui Zhang,[1,2] Yang Zhang,[3] Jianxi Fan,[1] and Jin Wang[1]

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*
[3] *School of Engineering, University of Glasgow, Glasgow, G12 8QQ, UK*

Correspondence should be addressed to Shukui Zhang; zhangsk2000@163.com

Received 13 July 2013; Accepted 22 September 2013

Academic Editor: Zhijie Han

Deployment is a fundamental issue in wireless sensor networks (WSNs), which affects the performance and lifetime of the networks. Usually the sensor locations are precomputed based on "perfect" sensor detection model, whereas sensors may not always provide reliable information, either due to operational tolerance levels or environmental factors. Therefore, it is very important to take into account this uncertainty in the deployment process. In this paper, we address the problem of sensor deployment in a mixed sensor network where the mobile and static nodes work collaboratively to perform deployment optimization task. We consider the Gaussian white noise in the environment and present a centralized algorithm (FABGM for short) which discoveres vacancies by using detection model based on false alarm and moves the mobile nodes according to the method based on bipartite graph matching in this study. In this algorithm, the management node of the WSNs collects the geographical information of all of the static and mobile sensors. Then, the management node executes the algorithm to get the best matches between mobile sensors and coverage holes. Simulation results are presented to demonstrate the effectiveness of the proposed approach.

1. Introduction

Wireless sensor networks have received intensive research interest in recent years due to their potential capability of monitoring real physical environments and collecting data. WSNs have been used in various applications, such as forest monitoring, precision agriculture [1], battlefield surveillance, and target tracking. However, in order to conduct their tasks successfully, it is very important that they be deployed properly.

Sensor deployment is at the initial stages in sensor networks research. It is an important issue which has attracted much attention in recent years [2]. The number and locations of sensors deployed in a Region of Interest (RoI) determine the topology of the network, which will further influence many of its intrinsic properties, such as its coverage, connectivity, cost, and lifetime. Consequently, the performance of a sensor network depends to a large extent on its deployment. A problem which impinges upon the success of any WSNs deployment is the fact that sensory data are marred by the flaw of uncertainty. Indeed, information provided by sensors may not always reliable, either due to environmental factors such as Gaussian white noise or operational tolerance levels. Therefore, it is very important to take into account this uncertainty in the deployment process.

However, WSNs cannot be deployed manually in many working environments, such as remote mountainous regions, battlefields, and regions polluted by poisonous gases. An alternative method is scattering the sensors randomly, but this is affected by many uncontrollable factors, and it is difficult to achieve the desired deployment. In the last decade, researchers have focused on mixed sensor networks, in which the static nodes and mobile nodes work in a collaborative fashion to perform deployment task. Such networks have the advantage of mobility so they can be moved to appropriate positions to enhance the extent of coverage and reduce the number of nodes.

In this work, we explore the problem of uncertainty-aware deployment in mixed wireless sensor networks. The original contributions of this work are the following: first, we

introduce a false alarm based detection model; a model considers the existence of Gaussian white noise. Second, using the detection model we compute joint detection probability to discover the coverage vacancies. And then, we present an approach which is based on bipartite graph matching to determine the position of the mobile sensor nodes. Before a mobile node moves to coverage vacancy, it will determine whether there are static nodes within its sensing range. If there are static nodes within its sensing range, it moves to the coverage vacancy. Otherwise, it remains in its current position. Experimental results are given to demonstrate the efficiency of our approach.

The remainder of this paper is organized as follows. Section 2 explains related works. Section 3 gives an overview of the false alarm based detection model, and we detail our deployment algorithm in Section 4. Section 5 presents our experiments. This paper is concluded in Section 6.

2. Related Prior Works

According to the characteristics of the nodes that comprise WSNs, there are three types of such networks, that is, (1) static WSNs, in which all the nodes are static; (2) mobile WSNs, in which all the sensors are mobile; and (3) mixed WSNs, in which some of the nodes are static and some are mobile.

The greatest weakness of a random static WSN is that there must be significant redundancy among the nodes in order to achieve good coverage. And the deployment of a deterministic static WSN is inefficient. In [3], the authors use a sequential deployment of sensors that is, a limited number of sensors; are deployed in each step until the desired probability of detection of a target is achieved. Sensor placement on two- and three-dimensional grids was formulated as a combinatorial optimization problem and solved using integer linear programming [4]. The main drawbacks of these approaches are that the grid coverage approach relies on "perfect" sensor detection; that is, a sensor is expected to yield a binary yes/no detection outcome in every case. The authors in [5] provide a polynomial-time, greedy, iterative algorithm to determine the best placement of one sensor at a time in a grid based scenario, such that each grid is covered with a minimum confidence level. Uncertainty associated with the predetermined sensor locations is considered in [6]. The authors propose two sensor placement algorithms, where the sensor location is modeled as a random variable with a Gaussian probability distribution. In [7, 8], the authors define an evidence-based coverage model and conceive an uncertainty-aware deployment algorithm, which determines the minimum number of sensors and their locations to ensure full area coverage. The evidence-based sensor coverage model is a generalization of the probabilistic model.

In mobile WSNs, a fundamental issue is the coverage problem. Many techniques have been developed to deal with this issue, such as coverage pattern-based movement [9–12], virtual force-based movement [13, 14], and Voronoi-based movement [15]. A distributed energy-efficient deployment algorithm is proposed by Heo and Varshney [9]. The goal is the formation of an energy-efficient node topology for a longer system lifetime. In order to achieve this goal, they employ a synergistic combination of cluster structuring and a peer-to-peer deployment scheme. Moreover, an energy-efficient deployment algorithm based on Voronoi diagrams is also proposed here. The authors in [13] propose a deployment strategy to enhance coverage after an initial random placement of sensors using virtual forces. A cluster head computes the new locations of all the sensors after an initial deployment that would maximize coverage, and then nodes reposition themselves to the designated locations. Wang et al. [15] use Voronoi diagrams to discover the coverage vacancies and design three movement-assisted sensor deployment protocols, including VEC (vector based), VOR (Voronoi based), and minimax. The greatest weakness of a mobile WSN is its price, which is significantly greater than the price of a static WSN, because the price of mobile sensors is much greater than the price of static sensors.

The mixed wireless sensor networks that are composed of a mixture of mobile and static sensors are the tradeoff between cost and coverage. To provide the required high coverage, the mobile sensors have to move from dense areas to sparse areas. In [16], the authors proposed a collaborative coverage enhancing algorithm (coven) which uses a "Voronoi polygon" to determine the placed positions and the number of estimated holes. However, it is not feasible to apply Voronoi diagrams in WSNs due to their excessive complexity. A grid deployment method is proposed in [17], where the map is divided into multiple individual grids, and the weight of each grid is determined by environmental factors such as predeployed nodes, boundaries, and obstacles. The grid with minimum values is the goal of the mobile node. The authors in [18] proposed a novel, centralized algorithm to deploy a mixture of mobile and static sensors to construct sensor networks which used Delaunay triangulation rather than a Voronoi diagram to detect the coverage holes.

To the best of our knowledge, almost all related works assume either a binary or a probabilistic-based coverage model. The binary coverage model is overly simplistic and does not reflect reality, while the probabilistic coverage model is limited and does not allow the easy integration of some related issues, such as sensor reliability. This paper presents a centralized algorithm in which the management node executed the algorithm to discover vacancies by using detection model based on false alarm and then to get the best matches between mobile sensors and coverage holes. The detection model used in our work considers the environment factor and reflects reality well. Simulation results show the effectiveness of our algorithm.

3. Detection Model

3.1. Assumptions. Our algorithm is based on the following assumptions.

(i) The location of each sensor node is known, which can be obtained at a low cost from Global Positioning System (GPS) or through location discovery algorithms.

(ii) It is assumed that all sensor nodes have identical capability for sensing and communication.

(iii) The mobile nodes have the ability to move and can move to the optimized position accurately.

3.2. Sensor Detection Model. We assume that the WSNs work in an environment with a Gaussian white noise, and each sensor involved in the signal detection transmits a signal with the same energy e_{tr}. Thus, the signal received at a distance of r meters away will have energy of e_{tr}/r^{γ}. Here, a simple geometric path loss model [19] is assumed, and the path loss is proportional to $1/r^{\gamma}$, where γ is the path loss exponent, which is an environment-dependent constant typically between 2 and 4. We assume that the number of sensors which perform signal detection is n. Thus, at sensor i, the observations under the two different hypotheses are given by [20]

$$y_i = \begin{cases} \dfrac{\beta}{D_{ti}^{\gamma/2}} + n_i & H_1,\ i = 1, 2, \ldots, n, \\ n_i & H_2,\ i = 1, 2, \ldots, n, \end{cases} \tag{1}$$

where H_1 denotes the target-present hypothesis, and H_0 is the null hypothesis; y_i is the received signal; n_i is zero-mean, complex Gaussian noise with variance σ^2; β is a scalar defined by $\beta = \sqrt{e_{\text{tr}}/2^{\gamma}}$; and D_{ti} denotes the distance between the target (x_t, y_t) and the sensor (x_i, y_i); that is,

$$D_{ti} = \sqrt{(x_t - x_i)^2 + (y_t - y_i)^2}. \tag{2}$$

It is noted that under hypothesis H_1, the distance that the active sensing signal traverses is given by $r = 2D_{ti}$. Therefore, for the ith sensor, the likelihood (probability density function) under H_1 is given by

$$\Pr(y_i \mid H_1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}\left(y_i - \frac{\beta}{D_{ti}^{\gamma/2}}\right)^2\right\}, \tag{3}$$

and the likelihood under H_0 is

$$\Pr(y_i \mid H_0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{y_i^2}{2\sigma^2}\right\}. \tag{4}$$

Now, let us focus on the Neyman-Pearson criterion. The Neyman-Pearson criterion maximizes the probability of detection P_D subject to a predetermined bound on the probability of false alarm P_F. In other words, the optimal decision rule ϑ according to the Neyman-Pearson criterion is the solution to the following constrained optimization problem [21]:

$$\max_{\vartheta} P_D(\vartheta) \ \text{ subject to } \ P_F(\vartheta) \le \alpha. \tag{5}$$

The Neyman-Pearson optimum test is a likelihood ratio test [13]. From (3) and (4), the likelihood ratio for sensor i can be written as

$$\begin{aligned} L_i(y_i) &= \frac{P_r(y_i \mid H_1)}{P_r(y_i \mid H_0)} \\ &= \exp\left\{\frac{1}{2\sigma^2}\left(\frac{2\beta y_i}{D_{ti}^{\gamma/2}} - \frac{\beta^2}{D_{ti}^{\gamma}}\right)\right\}. \end{aligned} \tag{6}$$

Since the $n_i, i \in [1, n]$, are assumed to be statistically independent, the joint probability of the observations is simply the product of the individual probability densities. Thus, let us define $y = [y_1, \ldots, y_n]$, for the n sensors, the overall likelihood ratio is

$$L(y) = \prod_{i=1}^{n} L_i(y_i). \tag{7}$$

For convenience, we consider the log-likelihood ratio, which is given by

$$\begin{aligned} \ln L(y) &= \sum_{i=1}^{n} \ln L_i(y_i) \\ &= \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(\frac{2\beta y_i}{D_{ti}^{\gamma/2}} - \frac{\beta^2}{D_{ti}^{\gamma}}\right). \end{aligned} \tag{8}$$

Therefore, the likelihood ratio test is given by [21]

$$\begin{aligned} \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(\frac{2\beta y_i}{D_{ti}^{\gamma/2}} - \frac{\beta^2}{D_{ti}^{\gamma}}\right) &\ge \ln\eta \quad H_1, \\ \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(\frac{2\beta y_i}{D_{ti}^{\gamma/2}} - \frac{\beta^2}{D_{ti}^{\gamma}}\right) &< \ln\eta \quad H_0, \end{aligned} \tag{9}$$

where η is uniquely determined by solving $P_F = \alpha$. Equivalently, we can reformulate (9) into

$$\begin{aligned} \underbrace{\sum_{i=1}^{n} \frac{y_i}{D_{ti}^{\gamma/2}}}_{g} &\ge \underbrace{\frac{1}{\beta}\sigma^2 \ln\eta + \frac{1}{2}\sum_{i=1}^{n} \frac{\beta}{D_{ti}^{\gamma}}}_{\tau} \quad H_1, \\ \underbrace{\sum_{i=1}^{n} \frac{y_i}{D_{ti}^{\gamma/2}}}_{g} &< \underbrace{\frac{1}{\beta}\sigma^2 \ln\eta + \frac{1}{2}\sum_{i=1}^{n} \frac{\beta}{D_{ti}^{\gamma}}}_{\tau} \quad H_0, \end{aligned} \tag{10}$$

where we have further defined the test statistics g and the new threshold τ. For a fixed set of sensors, the second part of τ will be fixed and known. The variable g is actually a *sufficient statistic*, and when making a decision, knowing the value of g will be just as good as knowing y. Then, invoking the model for n_i, the hypothesis pair can be written as

$$\begin{aligned} H_0 &: g \sim N\left(0, \sum_{i=1}^{n} \frac{\sigma^2}{D_{ti}^{\gamma}}\right) \ \text{versus} \\ H_1 &: g \sim N\left(\sum_{i=1}^{n} \frac{\beta}{D_{ti}^{\gamma}}, \sum_{i=1}^{n} \frac{\sigma^2}{D_{ti}^{\gamma}}\right). \end{aligned} \tag{11}$$

For notational convenience, let us define

$$\mu_1 = \sum_{i=1}^{n} \frac{\beta}{D_{ti}^{\gamma}}, \qquad \sigma_1^2 = \sum_{i=1}^{n} \frac{\sigma^2}{D_{ti}^{\gamma}}. \tag{12}$$

Thus, the false alarm probability is

$$\begin{aligned} P_F &= \Pr\left(g > \tau \mid H_0\right) \\ &= \Pr\left(\frac{g}{\sigma_1} > \frac{\tau}{\sigma_1} \mid H_0\right) \\ &= 1 - \Phi\left(\frac{\tau}{\sigma_1}\right), \end{aligned} \tag{13}$$

where $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function; that is,

$$\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz. \tag{14}$$

Similarly, the detection probability is given by

$$\begin{aligned} P_D &= \Pr\left(g > \tau \mid H_1\right) \\ &= \Pr\left(\frac{g - u_1}{\sigma_1} > \frac{\tau - u_1}{\sigma_1} \mid H_1\right) \\ &= 1 - \Phi\left(\frac{\tau - \mu_1}{\sigma_1}\right). \end{aligned} \tag{15}$$

It is clearly seen that with the aid of (11), the false alarm probability P_F, the detection probability P_D, and the detection threshold τ (or η) are connected by some one-to-one relations. Suppose that we define the allowed level of false alarm as $P_F = \alpha$, then from (13), we obtain

$$\frac{\tau}{\sigma_1} = \Phi^{-1}(1 - \alpha). \tag{16}$$

By the definitions of μ_1 and σ_1 in (12), we can write

$$\frac{\mu_1}{\sigma_1} = \frac{\beta}{\sigma}\left(\sum_{i=1}^{n} \frac{1}{D_{ti}^{\gamma}}\right)^{1/2}. \tag{17}$$

Therefore, by using (16) and (17), we can reformulate P_D of (15) into

$$P_D = 1 - \Phi\left(\Phi^{-1}(1 - \alpha) - \frac{\beta}{\sigma}\left(\sum_{i=1}^{n} \frac{1}{D_{ti}^{\gamma}}\right)^{1/2}\right). \tag{18}$$

We assume that the target may appear at a random position k in the detection area. By using (18), we can obtain the detection probability of the target at any position k.

Consider

$$C_k(P) = 1 - \Phi\left(\Phi^{-1}(1 - \alpha) - \frac{\beta}{\sigma}\left(\sum_{i=1}^{n} \frac{1}{D_{ki}^{\gamma}}\right)^{1/2}\right), \tag{19}$$

where n denotes the number of nodes which is deployed in the detection area. Here, D_{ki} denotes the distance between the point $k(x_k, y_k)$ and the ith sensor (x_i, y_i); that is, by using (19), we can formulate the detection probability of any point k in the detection area.

4. Deployment Algorithm

In this paper, we evaluate the coverage performance by area coverage rate. We assume that N static nodes and M mobile nodes are deployed in the $L \times L$ area. The $L \times L\,\mathrm{m}^2$ square monitored area is divided into $L \times L$ small uniform square grids. Each grid has the same length of 1 m. For simplicity, here we transform the area coverage problem of WSN into grid coverage problem. We compute joint detection probability $C_k(P)$ of the center point of grid k and use the detection probability $C_k(P)$ to measure whether each grid is covered. The area coverage rate is defined as the ratio between the coverage area $A_{\text{area}}(P)$ of the node set and the total area A_s of the detection region. Thus, the area coverage rate is

$$R_{\text{area}}(P) = \frac{A_{\text{area}}(P)}{A_s} = \frac{\sum_{k=1}^{L\times L} C_k(P)}{L \times L}. \tag{20}$$

4.1. The Discovery of Coverage Vacancies. At first, n sensors (include N static nodes and M mobile nodes) have been scattered randomly in a $L \times L$ area. After the initial random deployment of the sensor nodes, the distribution function of the detection probability of the nodes *approximates* to exponential form. In the two-dimensional detection area, we can obtain the joint detection probability of any grid k in the detection region by the sensor detection model based on false alarm (19). Then, we will search grids at $C_{\min}(P)$ (see Figure 1). The M (the number of mobile nodes) girds which have the lowest detection probability are defined as coverage vacancies. Since the detection probability of each position is continuous, the probability around the point which has lower probability is also low. Thus, moving the mobile nodes to the coverage vacancies can improve the coverage performance of the network. Repeating this process until the iteration times t reaches the preset value (t_{pre}) or $C_{\min}(P)$ achieves satisfactory probability (P_s). In order to avoid that the moving distance of a single node is too large, we deploy virtual nodes at the position of coverage vacancies per iteration. When the iteration process terminates, we move the mobile nodes to the position of virtual nodes.

4.2. The Movement of Mobile Nodes. After the termination of iteration, the position of the virtual nodes is defined as coverage vacancies. For the initial network deployment, Firstly we construct a bipartite graph $G = (V, E)$, $V = V_1 \cup V_2$, where V_1 denotes the set of mobile nodes and V_2 denotes the set of virtual nodes. We take the moving distance as cost to illustrate the construction method of the bipartite graph. The concrete method is given by the following.

(1) Add all the movable nodes into V_1.

(2) Add all the virtual nodes into V_2.

(3) For $\forall u \in V_1$ and $\forall v \in V_2$, if mobile node u can reach v, (the distance between u and v does not exceed the maximum moving distance d_u of u or the remaining energy of u is within a certain range E_{residual}) then add an edge (u, v) into the bipartite graph; the weight of the edge (u, v) which is defined as $w(u, v)$ denotes

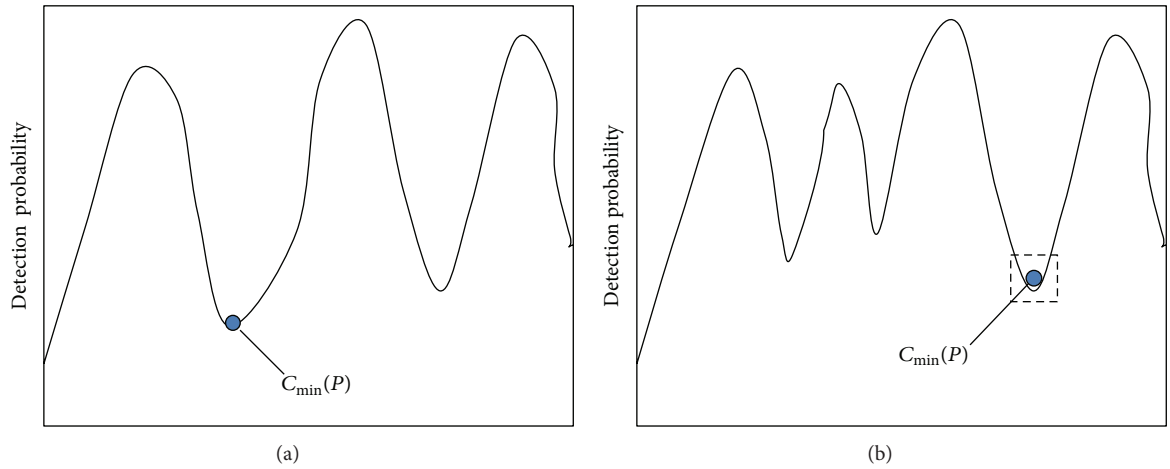


Figure 1: Schematic of joint detection probability. (a) Detection probability before iteration. (b) Detection probability after iteration.

the distance between sensor node u and virtual node v, otherwise, $w(u, v) = 0$.

A set H of independent edges in a graph $G = (V, E)$ is called a matching (see [22]). H is a matching of $U \subseteq V$ if every vertex in U is incident with an edge in H. The vertices in U are then called matched (by H); vertices not incident with any edge of H are unmatched. Because the two ends of any edge of G lie in different set of vertices, the number of edges in H denotes that $|H|$ virtual nodes (that is coverage holes) are covered by $|H|$ mobile nodes. Corresponding cost is given by $C_H = \sum_{(u,v)} w(u, v)$. Thus, maximizing the network coverage and minimizing the moving distance in this condition can be transformed into a problem of finding a maximum matching H_{opt} of bipartite graph G which has a minimum cost, that is, for any matching H of G, $|H| \leq |H_{\text{opt}}|$; If $|H| = |H_{\text{opt}}|$, then $C(H) \geq C(H_{\text{opt}})$. A maximum matching H_{opt} of bipartite graph G which has a minimum cost represents an optimal mobile solution. Deploying the mobile nodes according to the optimal mobile solution, the coverage of the network can be improved while the moving distance is the smallest.

For illustration, an example is given here. We assume that the vertex set of bipartite graph G is $V = V_1 \cup V_2$. The mobile node set $V_1 = \{x_1, x_2, x_3, x_4, x_5\}$, and virtual node set $V_2 = \{y_1, y_2, y_3, y_4, y_5\}$. Here, we just consider movement distance. As introduced above, if the distance between x_i and y_j does not exceed the maximum moving distance d_{x_i} of x_i, it indicates that mobile node x_i can reach y_j, then add an edge (x_i, y_j) into the bipartite graph G. The weight of edge (x_i, y_j) is defined as $w(x_i, y_j)$ which denotes the distance between sensor node x_i and virtual node y_i; otherwise, $w(u, v) = 0$. Otherwise, add an edge whose weight is 0 into the bipartite graph G. For example, if the distance between x_1 and y_1 is 3, it is assumed that the maximum moving distance of x_1 is 10, then add an edge (x_1, y_1) whose weight is 3 into the bipartite graph G. Similarly, we can obtain the weight of other edges of bipartite G, as shown in Figure 2(a). After the construction of the bipartite graph G, then we can obtain a maximum matching H_{opt} of bipartite graph G which has a minimum cost $H_{\text{opt}} = \{x_1y_5, x_2y_3, x_3y_4, x_4y_2, x_5y_1\}$.

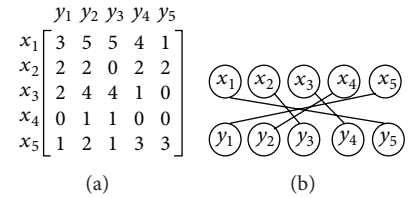


Figure 2: An example of bipartite graph construction (a) Weight matrix of edges of bipartite graph (b) the constructed bipartite graph.

Table 1: Settings of simulation parameter.

Parameter	Value
The area of A_s	$80 \times 80\ \text{m}^2$
The iteration times t	30
Total number of nodes: n	60
The value of γ	2
The value of σ	1
The value of β	7
The value of $P_F = \alpha$	5%

According to H_{opt}, we will move x_1 to y_5, x_3 to y_4, x_4 to y_2, and x_5 to y_1, respectively. For $w(x_2, y_3) = 0$, we do not move node x_2. The corresponding cost is 4. The concrete algorithm is as found in Algorithm 1.

5. Simulations Results

Now we present simulation results regarding the proposed algorithm. All of the simulation results given in the following are obtained by programs in C++ and Matlab. We consider a specific WSNs scenario where the field has a square shape of size $80 \times 80\ \text{m}^2$; 60 sensor nodes (including static and mobile nodes) are randomly distributed over this field. We ran 100 simulations for every result and calculated the average result. The experimental parameters are shown in Table 1.

input: Static nodes $S = \{s_1, s_2, \ldots, s_N\}$, Mobile nodes $V_1 = \{v_1, v_2, \ldots, v_M\}$, E_{residual}, d_u, $u \in V_1$, t_{pre}, P_s
False-alarm probability parameter $\alpha, \beta, \gamma, \sigma, e_{\text{tr}}$
Output:

(1) $t \leftarrow 0$.
(2) Calculate $C_k(P)$ of the target at any position k of area A_s using (19).
(3) **Repeat:**
(4) Search $C_{\min}(P)$, construct set of virtual node $V_2 = \{(x_{E,1}, y_{E,1}), (x_{E,2}, y_{E,2}), \ldots, (x_{E,M}, y_{E,M})\}$.
(5) Update the static node set $S = \{s_1, s_2, \ldots, s_N\} \cup V_2$.
(6) Update iteration times, set t: $t = t + 1$.
(7) Calculate $C_k(P)$ of the target at any position k of area A_s using (19).
(8) **Until** (iteration time t is larger than t_{pre} or $C_{\min}(P)$ is larger than P_s)
(9) Construct bipartite graph $G = (V, E)$, calculate H_{opt}.
(10) Adjust the position of the mobile node v_i, $v_i \in V_1$ according to optimal mobile solution which is in consistent with H_{opt}.

Algorithm 1: Algorithm FABGM.

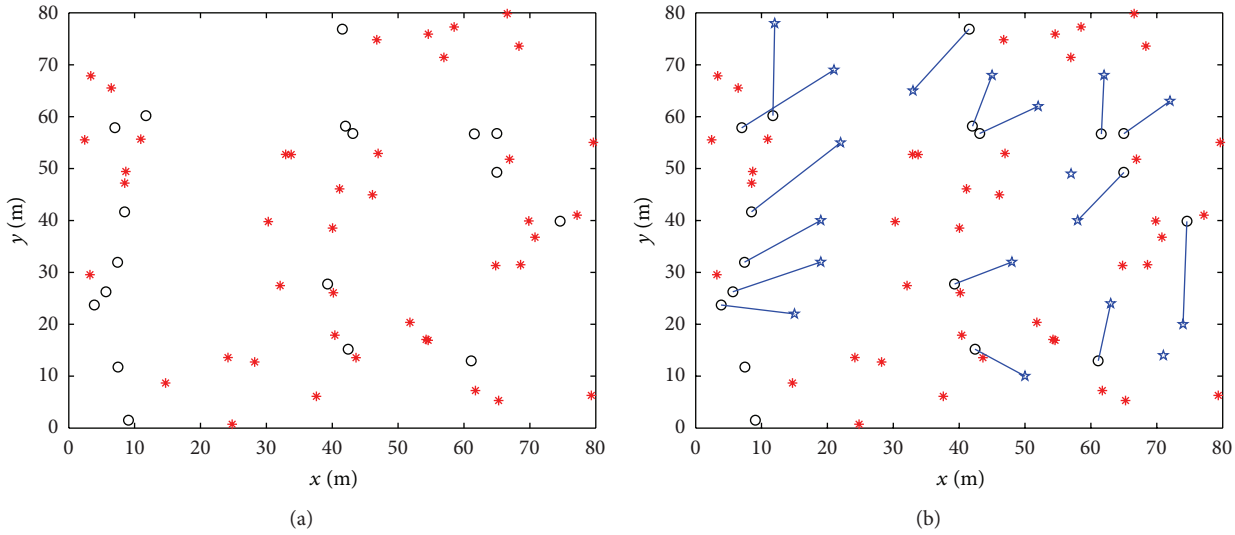


Figure 3: Position of mobile nodes before and after optimization.

Figure 3 shows the schematic diagram of movement condition of the mobile nodes. The circle represents the mobile node and the five-pointed star represents coverage void. If a circle i and a five-pointed star j are connected by a line, it indicates that mobile node represented by circle i can move to the coverage hole represented by the five-pointed star j. As introduced above, if the distance between mobile node u and coverage hole v exceeds the maximum moving distance d_u of u or the remaining energy of u is less than a preset value E_{residual}, we set $w(u, v) = 0$. Thus, according to practical situation, we can set different d_u and E_{residual}, then we can obtain a different weight matrix regarding edges of the constructed bipartite graph and different optimal mobile solutions. Therefore, the proposed algorithm is more flexible and reflects reality well. After performing FABGM algorithm, the sensors are distributed uniformly in the detection area, which helps to increase the system efficiency, improve energy conservation, and decrease the probability of missing an event.

Figure 4 provides the schematic diagram of detection probability distribution comparison between the random method and our deployment method. It is obvious that there are many coverage vacancies and points with low probability in the network after the initial random deployment. From Figure 4(b), we know that the proposed algorithm in this paper can decrease the coverage void and improve the coverage performance of the network. Figure 5 shows the coverage performance of FABGM algorithm when the mobile nodes percentage varies from 0% to 80%. Due to the lack of reference system, we compare the proposed algorithm with the initial deployment. We can see that our deployment can enhance coverage compared to the initial random deployment and that the network coverage quality increases with the number of mobile nodes increase. However, when the mobile node percentage is larger than 50%, network coverage increases slowly along with the increase of the number of mobile nodes.

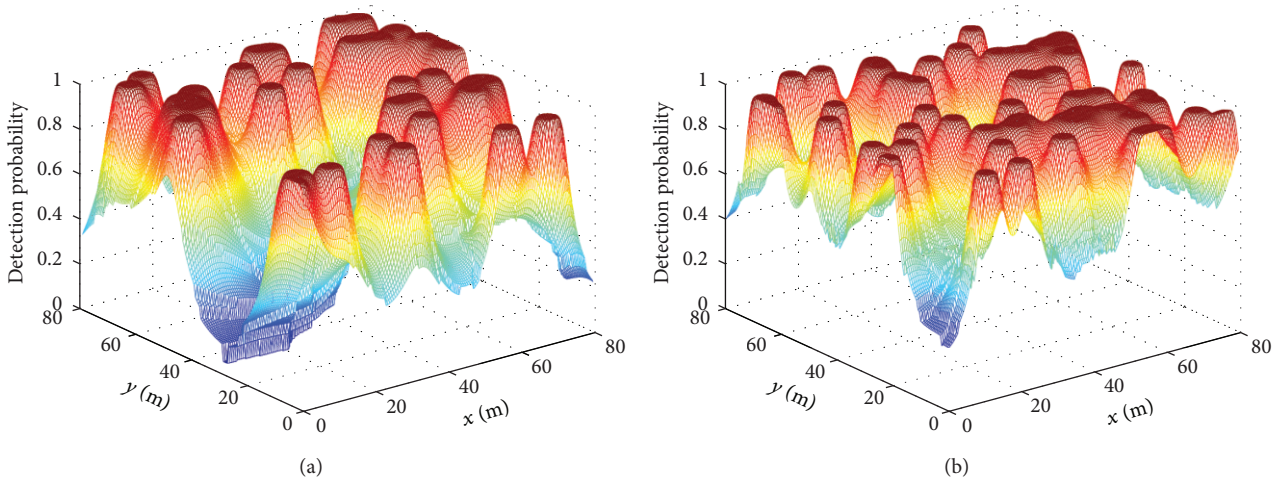


Figure 4: Distribution of joint detection probabilities. (a) Random deployment. (b) Distribution after running FABGM.

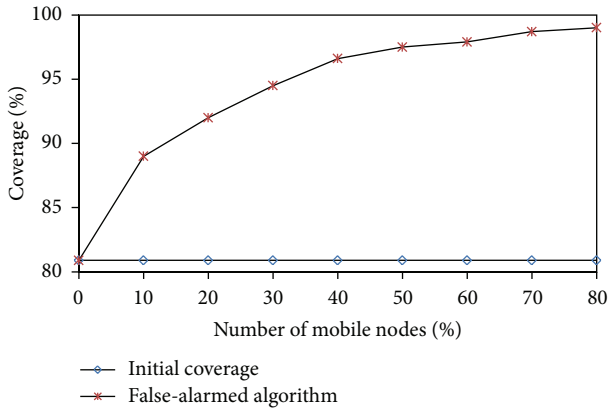


Figure 5: Coverage quality before and after optimization.

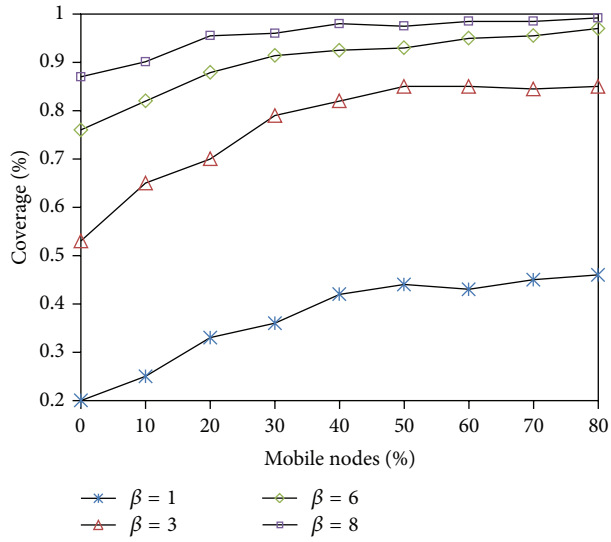


Figure 6: Coverage comparison of different β.

In order to analyze the relationship between coverage performance and the value of β, we discuss the network coverage condition when the value of β is 1, 3, 6, and 8 separately. From Figure 6, we know that when the value of β is fixed, the network coverage improves with the increase of number of mobile nodes. When the value of β is small, the network coverage improves apparently. When the value of β is large (e.g., = 6, 8), increasing the number of mobile nodes will not improve network coverage significantly. That is because the value of β is proportionate to e_{tr} when γ is fixed, when the value of β is large, e_{tr} is also large. It indicates that each sensor node consumes much energy to transmit signal, so its detection range increases. The initial deployed sensors can almost cover the whole area, and so increasing the number of mobile nodes does not enhance network coverage significantly. The value of β can be determined according to practical application.

Figure 7 gives the minimum number of sensors required to reach a certain coverage percent when the percentage of mobile sensors varies from 0% to 80%, in 10% increments. As shown in Figure 7, the sensor nodes required in the mixed WSN are less than that in the static WSN. As the percentage of mobile sensors is 40%, the number of sensors in the mixed WSN is about half the number of sensors in the static WSN. The number of sensors needed decrease with the increase of the number of mobile nodes. When the percentage of mobile sensors increases from 0% to 30%, the number of sensors for 80% coverage decreases from 65 to 35, but when the percentage increases from 50% to 70%, the effect is not as significant as before. Considering that the price of mobile sensors is higher than the price of static sensors, it is necessary to determine proper number of mobile nodes in a network. However, the number of mobile nodes is related to the ratio of the price of mobile sensors to the price of static sensors.

Figure 8 shows the cost of WSN to reach 85% coverage at different ratios of the price of mobile sensors to the price of static sensors. When the price discrepancy of mobile nodes and static nodes is minor, increasing the proportion of mobile

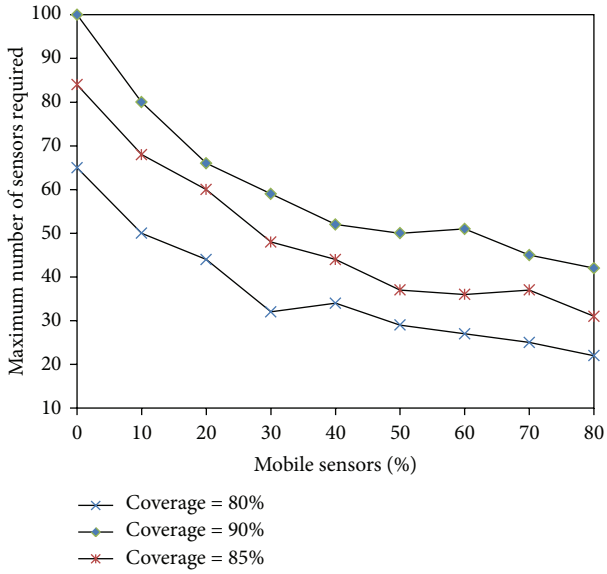


Figure 7: The minimum number of sensors required to reach a coverage percent in different mobile percentages.

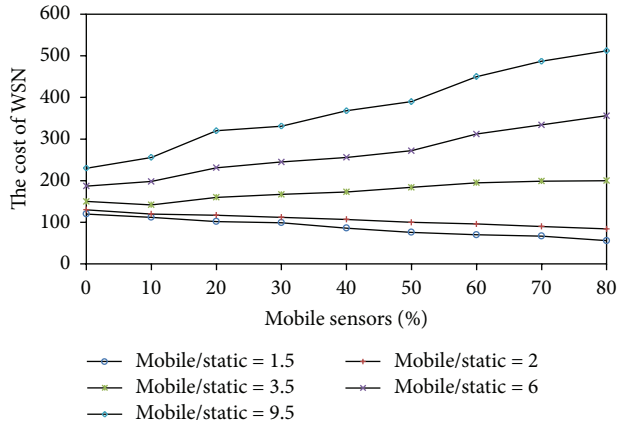


Figure 8: The cost to reach 85% coverage.

sensors can reduce the overall cost of WSN. When the ratio is between 2 and 6, the mixed WSN has the lower cost, and also the proportion of mobile sensors is not very high (e.g., 20% ≤ the percentage of mobile sensors ≤ 50%). But when the ratio is greater than 6, it is better that the nodes in the network are all static nodes. It is apparent that the mixed WSN is a best choice when the ratio between the price of the mobile sensor and the price of the static sensor is 2–6. Under this circumstance, our proposed algorithm is useful and can provide a tradeoff between the cost and the coverage.

6. Conclusion

Sensor deployment is at the initial stages in sensor networks research. Here, we explore the problem of uncertainty-aware WSNs deployment, and we start with a "random" distribution of the nodes (including mobile nodes and static nodes) over the region of interest. In this paper, we propose a false-alarm deployment algorithm to improve an initial deployment of nodes. After going through the proposed algorithm, the area of interest is covered by uniformly nodes. Simulation results demonstrate the effectiveness of the proposed approach.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants No. 61070169 and Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, the Specialized Research Foundation for the Doctoral Program of Higher Education of China No. 20103201110018 and Application Foundation Research of Suzhou of China No. SYG201118, SYG201240.

References

[1] M. Mafuta, M. Zennaro, A. Bagula, G. Ault, H. Gombachika, and T. Chadza, "Successful deployment of a wireless sensor network for precision agriculture in malawi," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 150703, 13 pages, 2013.

[2] B. Wang, *Coverage Control in Sensor Networks*, Springer, New York, NY, USA, 2010.

[3] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for target detection," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 42–48, New York, NY, USA, September 2002.

[4] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448–1453, 2002.

[5] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," in *Proceedings of the 5th International Conference on Information Fusion*, vol. 2, pp. 1581–1587, 2002.

[6] Y. Zou and K. Chakrabarty, "Uncertainty-aware sensor deployment algorithms for surveillance applications," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, pp. 2972–2976, December 2003.

[7] M. R. Senouci, A. Mellouk, L. Oukhellou, and A. Aissani, "Uncertainty-aware sensor network deployment," in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference (GLOBECOM '11)*, December 2011.

[8] M. R. Senouci, A. Mellouk, L. Oukhellou, and A. Aissani, "Efficient uncertainty-aware deployment algorithms for wireless sensor networks," in *Proceedings of the 54th IEEE Wireless Communications and Networking Conference on Mobile and Wireless Networks*, pp. 2163–2167, December 2012.

[9] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 35, no. 1, pp. 78–92, 2005.

[10] P. C. Wang, T. W. Hou, and R. H. Yan, "Maintaining coverage by progressive crystal-lattice permutation in mobile wireless sensor networks," in *Proceedings of the 2nd IEEE International Conference on Systems and Networks Communications (ICSNC '06)*, pp. 1–6, Polynesia, Tahiti, Island, 2006.

[11] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE*

Transactions on Mobile Computing, vol. 7, no. 2, pp. 262–274, 2008.

[12] Y.-C. Wang and Y.-C. Tseng, "Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1280–1294, 2008.

[13] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 1, pp. 61–91, 2004.

[14] J. Li, B. H. Zhang, L. G. Cui, and S. C. Chai, "An extended virtual force-based approach to distributed self-deployment in mobile sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 417307, 14 pages, 2012.

[15] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[16] A. Ghosh, "Estimating coverage holes and enhancing coverage in mixed sensor networks," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 68–76, November 2004.

[17] R. C. Luo and O. Chen, "Mobile sensor node deployment and asynchronous power management for wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 5, pp. 2377–2385, 2012.

[18] Z. J. Zhang, J. S. Fu, and H. C. Chao, "An energy-efficient motion strategy for mobile sensors in mixed wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 813182, 12 pages, 2013.

[19] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Upper Saddle River, NJ, USA, 2th edition, 2002.

[20] Y. Yang and R. S. Blum, "Energy-efficient routing for signal detection under the Neyman-Pearson criterion in wireless sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 303–312, April 2007.

[21] H. V. Poor, *An Introduction To Signal Detection and Estimation*, Springer, New York, NY, USA, 2nd edition, 1994.

[22] R. Diestel, *Graph Theory*, Springer, New York, NY, USA, 3nd edition, 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 127203, 8 pages
http://dx.doi.org/10.1155/2013/127203

Research Article

A New Metric for Modeling the Uneven Sleeping Problem in Coordinated Sensor Node Scheduling

Gaojuan Fan[1,2] and Chongsheng Zhang[1]

[1] *School of Computer and Information Engineering, Henan University, Kaifeng, Henan 475004, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China*

Correspondence should be addressed to Gaojuan Fan; fangaojuan@henu.edu.cn

Received 19 July 2013; Accepted 21 September 2013

Academic Editor: Fu Xiao

Monitoring performance and energy constraint are two conflicting aspects in wireless sensor networks. In order to save battery power in very dense sensor networks, some redundant sensors can be put into the sleep state while other sensor nodes remain active for the sensing and communication tasks. However, if the energy consumptions for some nodes are more than those of the others, these nodes would lose effectiveness earlier than the remaining ones, which subsequently causes the coverage holes. Because of the uneven node scheduling problem in the sensing areas and the partitions of communication network, the coverage holes further influence the sensing and communication qualities and the network life. In this work, we propose the uneven sleeping problem (USP) happening in the coordinated node scheduling in wireless sensor networks. We analyze the key factors that may lead to USP. We design a new metric which can better measure and evaluate USP. This new metric takes the influence of the boundary effects on node schedule into consideration. We experimentally evaluate the performance of our proposal. The results show that our metric can effectively identify the boundary sensor nodes that deserve equal chances of sleep; it thus reduces the number of dead sensors and can help achieve a longer network lifetime than existing methods.

1. Introduction

A wireless sensor network is typically composed of a large number of tiny, low-powered sensor nodes equipped with data processing, sensing, and communication capabilities that communicate with each other through single or multihop wireless links for monitoring the surveillance field. Wireless sensor networks have been extensively used in a variety of domains, for example, environmental observation, military monitoring, healthcare, and commercial applications [1]. The sensor nodes usually operate on an unattended manner and are battery powered. However, it is usually hard to renew or recharge the battery after deployment, due to environmental situations or cost concerns. Therefore, energy efficiency has been a very challenging task in achieving long lifetime in large-scale sensor networks.

Coordinated node scheduling plays a critical role that exploits the redundancy of sensor nodes to minimize the number of active nodes while preserving some properties of the network. The problem of sensor scheduling to maximize the network lifetime while guaranteeing the sensing coverage and network connectivity has been extensively addressed in the literature [2, 3]. Network coverage is the central consideration for the sensor scheduling methods that conserve energy by powering off redundant nodes while still assuring network coverage.

Energy consumption rates in different parts of the network should be even or almost even. Thus all nodes throughout the network area have about the same lifetime. Otherwise, some parts of the network may die much sooner than the others. If some critical parts of the network run out of battery early, it may lead to early dysfunction of the entire network, even if other parts of the network still have a lot of residual energy. In this paper, we propose the uneven sleeping problem and a new metric to be used in coordinated node scheduling for wireless sensor networks.

The remainder of this paper is organized as follows. In Section 2, we present related work on coordinated sensor

scheduling problems. Section 3 describes network model and introduce the uneven sleep problems. In Section 4, we present detailed analysis on uneven sleep problem and give new metric for proposed node scheduling scheme. Experimental results are given in Section 5. Finally, we give a summary and future work in Section 6.

2. Related Work

From perspectives of network configuration, network performance, network parameters, and models for recognizing redundant nodes, we categorize coordinated node scheduling into four types shown in Figure 1.

In the initial steps of WSN deployment, the status of the nodes can be scheduled according to preconfigured node location or distance information, to determine the coverage of the network node density. Coverage redundancy discrimination is a key step in coordinated node scheduling. It aims to identify the coverage redundant nodes and hibernate them alternately. There are generally two ways to detect redundant nodes, which are location-based and location-free techniques. We will address these two types of techniques in the following.

2.1. Location-Based Coordinated Nodes Scheduling. Location-based node scheduling can guarantee full (100%) coverage of the network, meaning that the monitor environment will be covered with all the active nodes. In paper [2], the authors define a "sponsored area" as the largest area a node can be covered by some communication neighbor. If the sensing area of a node can be covered by the combined sponsored areas of all of its communication neighbors, then this node can be shut down. In [3], the neighbors of a node are expanded to an area of a circle of radius $2r$ to further reduce the redundancy. In [4], a distributed and local redundancy detection method is developed according to bounded coverage.

Zhang and Hou [5] adopted a crossed redundancy detection method and proved that the full coverage network must be connected if the radius of the node's communication area is no less than 2 times of the radius of its sensing area. Under the assumption that the location information of each node is available, they proposed an Optimal Geographical Density Control (OGDC) algorithm. Xing et al. [6] extended the above work, and proposed the Coverage Configuration Protocol (CCP) algorithm that maximizes the number of hibernate nodes under the constraint of k-coverage and k-connection. They found that when the communication radius is 2 times of that of the sensing area, a k-coverage network is still k-connected.

All the above methods require the exact location information of the nodes to compute the coverage area and assure the full coverage. This is very challenging when designing an economic WSN of high density. In real-world applications, it is not mandatory to guarantee the complete coverage, and it is acceptable to sacrifice some of the coverage requirements for performance improvement.

In [7], the authors presented a dynamic node scheduling method, given that existing node scheduling methods always require static 1- or k-coverage. This method can adjust the status of network coverage on the fly. Concerning the boundary effects of circle coverage, the authors in [8] developed an improved circle coverage density control algorithm.

Hefeeda and Ahmadi in [9] proposed a new probabilistic coverage protocol, considering that the sensing ability may be influenced by the external application environment. In [10], a probabilistic k-coverage protocol was proposed that used heuristic greedy algorithm to decide the smallest set of active nodes. Shi et al. [11] gave two coverage algorithms based on probabilistic sensing models, for the consideration that the detection of an event is influenced by factors such as the distances between nodes, the physical characteristics of the nodes, the number of neighbors, and so on.

Exact location information is needed by all the above methods. To acquire such location information, people must have GPS, directional antennas infrastructures, or positioning devices. This constraint not only results in high cost, high complexity, and more energy consumption but also reduces the benefits brought by the node scheduling algorithm. Still, it is difficult to apply them to large-scale application for low purity of the current positioning devices.

2.2. Location-Free Coordinated Nodes Scheduling. Considering the problems with location-based scheduling algorithms, location-free coordinated node scheduling algorithms have been developed which are independent of the location information when scheduling and scheduling the nodes. This type of methods is economic and can be applied in applications that do not require strict network coverage rate. For these methods, the number of neighbor nodes or the distance information are usually used to detect redundancy.

Tian and Georganas in [12] first proposed the location-free redundancy detection model. They set a redundancy threshold based on the number of neighbors or the nearest distance within the sensing area. A node will hibernate if the number of nearest neighbors or the distance is smaller than the threshold. This method is simple and easy to implement. But it only guarantees partial coverage. Still, it does not consider the situation whether the neighboring nodes can fully or partly cover the monitoring area when this node is shut down. Consequently, the monitoring quality is not guaranteed.

Gao et al. in [13] analyzed the lower and upper bounds of complete node redundancy and proposed to compute the redundancy probability using the number of neighbors within the sensing area. Lightweight Deployment-Aware Scheduling (LDAS) [14] is an improvement of [13]; a number of nodes will be closed when the number of nearby working nodes reaches the threshold. Besides being location free, this new method can guarantee the network connectivity. The constraints of this method are as follows: the senor nodes should be uniformly distributed and each node can get the number of its neighbors.

Yen and Cheng in [15] proposed Range-Based Sleep Scheduling (RBSS) that determines the redundancy using the distance between the node and its neighbors. It assumes that the density of the sensor nodes is large enough to become

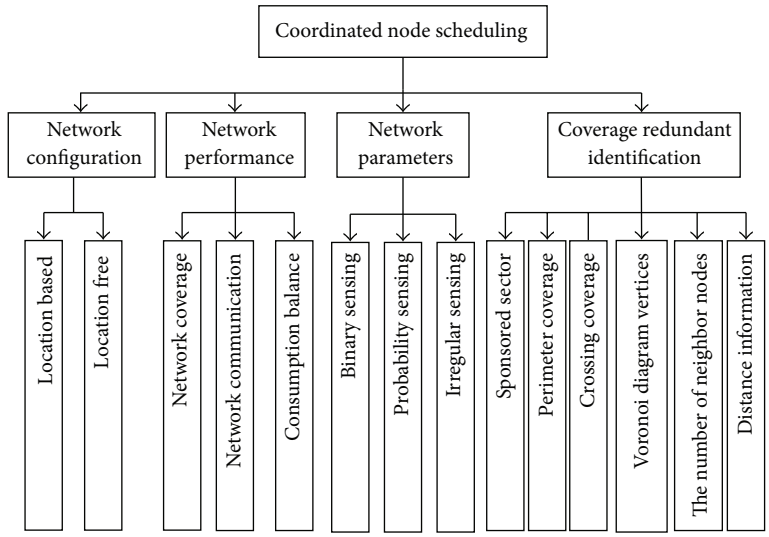


Figure 1: Taxonomy for coordinated node scheduling.

independent of the exact location information. While still assuring the complete network coverage and connectivity, it searches an optimal model from the nodes already deployed to maintain as less working nodes as possible. However, the energy cost of the method is imbalanced for it does not consider the energy consumption during the voting for start and coordinated nodes. Still, the early expiration of boundary nodes reduces the coverage area. In consideration of the imbalanced energy consumption problem, the authors in [16] presented a multistart and energy-adaptive strategy to do coverage control. Their method tackles the above-mentioned problem of the early expiration of boundary nodes.

Younis et al. in [17] designed a location-unaware coverage (LUC) method that utilizes the density between the node and its 2-hop neighbors to detect redundancy.

Location-based coordinated node scheduling can guarantee high coverage rate but requires special devices to obtain this location information, which reduces the benefits from node hibernation, whereas location-free coordinated node scheduling just needs information of the distance between the nodes and the number of nodes, so it cuts the corresponding extra costs.

3. Network Model and Problem Description

In this section, we first describe the network model and introduce the definitions and notations used in the paper. Next, we define the problem to be solved in this paper. Finally, we explain our motivations for this work.

3.1. Network Model. In a typical monitoring application where massive tiny sensor nodes are deployed, let the sensor nodes be uniformly distributed in the monitoring field. Let the sensing area of a senor node be a circle of radius r. Then, each sensor node can only monitor the environment and detect events within its sensing circle. A point is covered if and only if it locates in at least one sensor node's sensing circle.

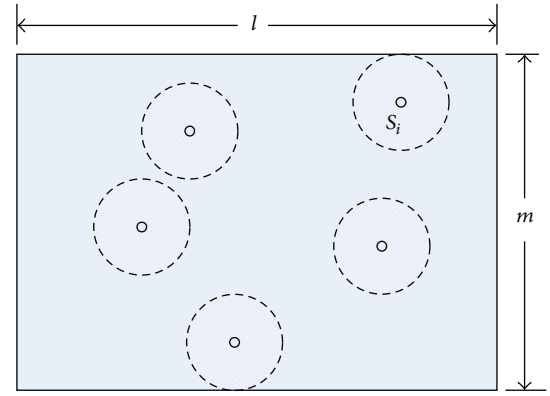


Figure 2: Monitored region.

If and only if all the points in a region are covered, the region is covered.

For simplicity, we assume that there are N homogeneous sensor nodes deployed within the monitored region A, denoted by a set $S = \{S_1, S_2, \ldots, S_N\}$, where A is a 2D $l \times m$ rectangular area, as shown in Figure 2. It should be noted that our method in this work can be easily extended to 2D monitored regions of arbitrary shapes.

Below we give the definitions used in this paper.

Definition 1 (Boundary Nodes). Let $d(i)$ be the Euclidean distance from the boundary of monitored region to any senor node S_i. The boundary nodes of a monitored region A, denoted by Bou(A), contains all of the sensors that satisfy $0 \le d(i) \le r$. Then,

$$\mathrm{Bou}(A) = \{i \in S : 0 \le d(i) < r\}. \quad (1)$$

Definition 2 (Interior Nodes). The interior nodes of monitored region A, denoted by Int(A), includes all of the sensors that satisfy $d(i) > r$. Then,

$$\mathrm{Int}(A) = \{i \in S : d(i) > r\}. \quad (2)$$

Definition 3 (Redundancy Rate). The redundancy rate is the size of the intersection area between the sensing area of node S_i and the total sensor area of its neighbors, divided by the size of the sensing area of S_i. That is,

$$\theta = \frac{\left(\bigcup_{j=1}^{N_1(i)} C\left(S_j\right)\right) \bigcap C\left(S_i\right)}{C\left(S_i\right)}, \quad (3)$$

where $C(S_i)$ denotes the sensing area of node S_i and $N_1(i)$ represents the number of neighbor of node S_i.

Definition 4 (Coverage Redundancy Node (CRN)). A node S_i is coverage redundant node if $\theta_i \geq \theta_{\mathrm{th}}$, where θ_{th} is a redundancy threshold which depends on application requirement. From this, when $\theta = 1$ is the full redundancy node, $\theta \leq 1$ is partial redundancy node.

Definition 5 (Effective Coverage Area (ECA)). The intersection area of sensor nodes' sensing coverage and the monitor region, denoted as $Ef(d(i))$, is the effective coverage area of sensor S_i. Therefore, for the interior nodes, the effective coverage area is equal to the sensing coverage; that is, $Ef(d(i)) = \pi r^2$. For the boundary nodes, it holds that $Ef(d(i)) = C(S_i) \cap C(A) < \pi r^2$.

3.2. Problem Definition. For a densely deployed monitoring application, the coverage redundancy nodes should be switched to sleep state in turn, to prolong the network lifetime while still preserving some properties of the network. The challenge is that, some nodes in Bou(A) or Int(A) have less chance to enter into sleep state, and the redundancy rate of any of these nodes is lower than θ. As a result, these nodes will remain active until their energy runs out, resulting in the coverage holes problem in the local monitoring area. Moreover, these nodes that have no energy may make their neighbors have fewer chances to be in the sleep state. Then these neighbors will again run out energy and cause their respective neighbors to use out the energy quickly. This loop continues, the coverage holes spread from boundary nodes to interior nodes, and the total monitoring area that can be covered by the active nodes shrinks inwardly to the center of the monitor region. We call this phenomenon uneven sleep sroblem (USP). Figure 3 shows the USP occurred in the node scheduling algorithm LDAS [14].

In Figure 3, 100 sensor nodes are randomly deployed in a circular area with radius $r = 50$. We show the node states with network running time being 600 s (Figure 3(a)) and 1000 s (Figure 3(b)). In the figures, the hollow points represent the active nodes and the solid points represent the sleeping nodes, whereas the solid square red points represent the dead nodes that have no energy. As we can see in Figure 3, as the monitoring runs, some boundary nodes and interior nodes run out of their energy due to the lack of chances to enter into sleep state.

3.3. Motivation. The task of node scheduling will be trivial if it is done in a centralized way. Under this assumption, it will be easy to determine which nodes are the boundary nodes and internal nodes and whether their sensing areas are covered by their neighbors or not and then decide which nodes should go to sleep or stay active. However, due to the energy constraints and the energy-cost considerations, this scenario is impractical for most monitoring applications. In the next section, we analyze several factors that influence the uneven sleep problem and derive instructions to stop the wretched cycle of energy holes. Under the analysis, we propose an improved node scheduling scheme which can alleviate the uneven sleep problem.

4. Problem Analysis and Proposed Approach

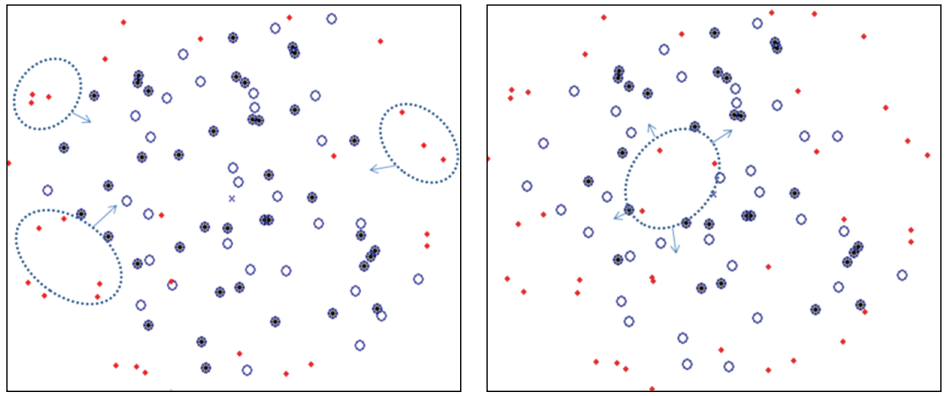
4.1. Problem Analysis. The failure of sensor nodes due to energy exhaustion or physical destruction may lead to the shrinkage of sensor coverage. The coverage holes can hardly be avoided in monitoring application because of the geographic environments or the uneven energy consumption problem even in the high-density deployment. In this section, we first give analysis on why the nodes that are in Bou(A) or Int(A) may encounter the uneven sleep problems.

We divide the monitored region A into three subregions, $\Omega_r(0)$, $\Omega_r(1)$, and $\Omega_r(2)$, as shown in Figure 4. The areas of these three regions are denoted as follows:

$$\begin{aligned} \left\|\Omega_r(0)\right\| &= \frac{(l-2r)(m-2r)}{lm}, \\ \left\|\Omega_r(1)\right\| &= \frac{2r((l-2r)+(m-2r))}{lm}, \\ \left\|\Omega_r(2)\right\| &= \frac{4r^2}{lm}. \end{aligned} \quad (4)$$

In Definition 4, we give ECA to illustrate the relationship of nodes' sensing coverage and the monitored region. For subregion $\Omega_r(0)$, effective coverage area is equal to the sensing coverage; that is, $Ef(d(i)) = \pi r^2$ when nodes are in Int(A). Meanwhile, when nodes are within Bou(A), then $Ef(d(i)) = C(S_i) \cap C(A) < \pi r^2$. This happens due to the border effects; the nodes in Bou(A) have less chance to be covered by their neighbors, so the nodes have little chance of becoming coverage redundancy node so as to enter into sleep state. These nodes will die out of energy, and then the death spreads to the inner region.

The problem illustrated above is the main reason for the uneven sleep problem occurring on the boundary nodes. The sensor nodes may also fail for reasons such as nonuniform deployment or environment influence (such as animals trample, the damage of the rain, etc.). It is clear that the network lifetime will decrease when the uneven sleep problem arises. In the next section, we give a proposed approach to mitigate the uneven sleep problem.



(a) 600 s (b) 1000 s

Figure 3: Uneven sleep problem.

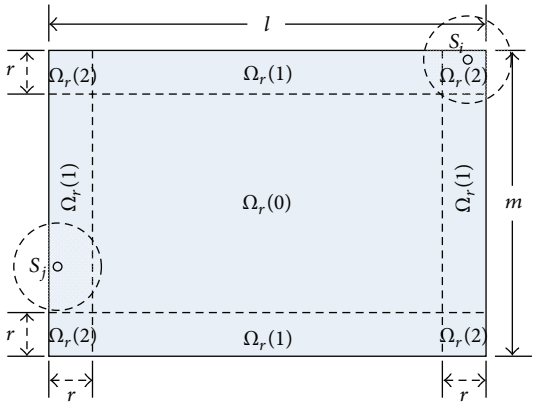


Figure 4: The subregions of monitored area.

4.2. Proposed Approach. In this section, we propose our approach which is an improvement based upon LDAS which schedules nodes without location information [14]. The proposed node scheduling scheme is divided into a few identical rounds, as shown in Figure 5. Each round starts with (i) neighbor discovery, (ii) redundancy evaluation, and (iii) node scheduling where the off-duty nodes will go to sleep and save energy.

4.2.1. Neighbor Discovery Phase. To obtain neighbor node information, each node broadcasts a *hello* message, which contains the node's position information or number of neighbors, to its one-hop neighbors at the beginning of neighbor discovery stage. This way, a sensor node S_i obtains the $N_1(i)$ as follows:

$$N_1(i) = \left\{S_j \in S \mid d\left(S_i, S_j\right) \leq r,\ S_i \in S,\ S_j \neq S_i\right\}, \quad (5)$$

where $d(S_i, S_j)$ represented the distance between the node S_i and the node S_j and r denotes the sensing radius of nodes.

4.2.2. Redundancy Evaluation Phase. In LDAS, redundancy evaluating is based on the number of neighbors. The percentage of a sensor's sensing area that is covered by its n random neighbors is not smaller than

$$1 - n0.609^{n-1} - \left(\frac{n}{6} - 0.109\right)0.109^{n-1}. \quad (6)$$

The expectation area that is not covered by random neighbors is

$$E\,[\text{uncover}] \leq \pi r^2 0.609n + \left(\frac{n\pi r^2}{6} - 0.109\pi r^2\right)0.109^{n-1}. \quad (7)$$

For more detailed proof, the readers can refer to [14].

To determine which nodes are coverage redundancy nodes, it is needed to compute the redundancy rate. A node has more chance to enter sleep state if the redundancy rate satisfies application requirement. For a node with sensing area of radius r, the redundancy rate is defined as

$$\theta = \frac{\pi r^2 - E\,[\text{uncover}]}{\pi r^2}. \quad (8)$$

As stated in Section 4.1, the boundary nodes in monitored area encounter the problem of border effects. To solve the uneven sleep problem in node scheduling, we should take border effects into consideration. As shown in Figure 3, we give the expected coverage area that an arbitrary sensor is located in three sub-regions (see our work published in [18] for further details):

$$\begin{aligned} E_r\,[0] &= \pi r^2, \\ E_r\,[1] &= \left(\pi - \frac{2}{3}\right)r^2, \\ E_r\,[2] &= \left(\pi - \frac{29}{24}\right)r^2. \end{aligned} \quad (9)$$

Then we can get the expected coverage area of the sensor, with sensing range being r and the sensor nodes being uniformly distributed in $l \times m$ rectangular monitored region:

$$E\,[r] = \frac{\pi r^2 lm - (4/3)\,r^3 l - (4/3)\,r^3 m + (1/2)\,r^4}{lm}. \quad (10)$$

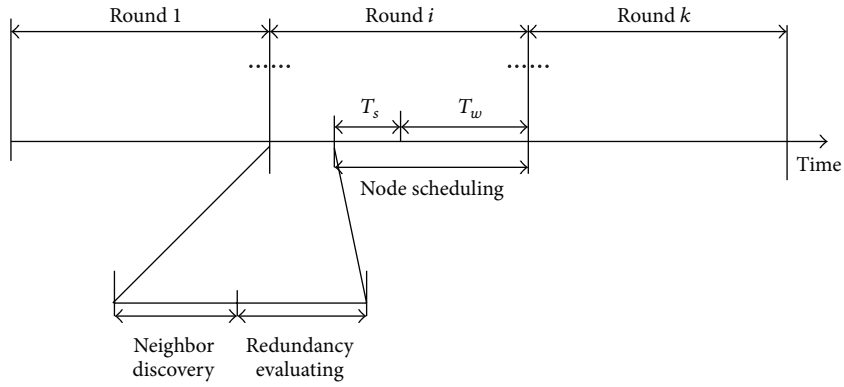


Figure 5: proposed node scheduling.

Taking border effects into consideration, we use the node's expect coverage area to replace the nodes coverage area to decide whether the node is a coverage redundancy node. Finally, we have the following formula:

$$\theta = \frac{\pi r^2 - E\,[\text{uncover}]}{E\,[r]} \geq \theta_{\text{th}}, \tag{11}$$

where variable θ_{th} is the threshold for a node to become a coverage redundancy node.

4.2.3. Node Scheduling Phase. In node scheduling phase, each node has three states: active state, presleeping state, and sleep state.

We assume that all the nodes are in active state in the beginning of the network. Since all the nodes are scheduled round robin, here we only describe the scheduling process in one round. There are 5 steps in one round.

Step 1. If the current node is active, there are two cases to consider. If there is data in the network, process the data; otherwise, go to step 2.

Step 2. Determine whether the current node is a coverage redundancy node using the techniques given in Section 4.1. If yes, go to step 3; otherwise, return to step 1.

Step 3. Send a pre-sleeping message to the neighbors of the current node and start the latency timer T_{backoff}.

Step 4. The current node listens to messages from the neighbors. If within T_{backoff} interval, it can receive more than one pre-sleeping messages from its neighbors; then return to step 1; otherwise, go to step 5.

Step 5. The current node begins to sleep for T_s time.

5. Performance Evaluation

We refer to our proposal as expectation-based LDAS, or ELDAS for short. In order to validate the effectiveness and correctness of ELDAS, we conduct experiments using MATLAB. We measure the performance of ELDAS in terms

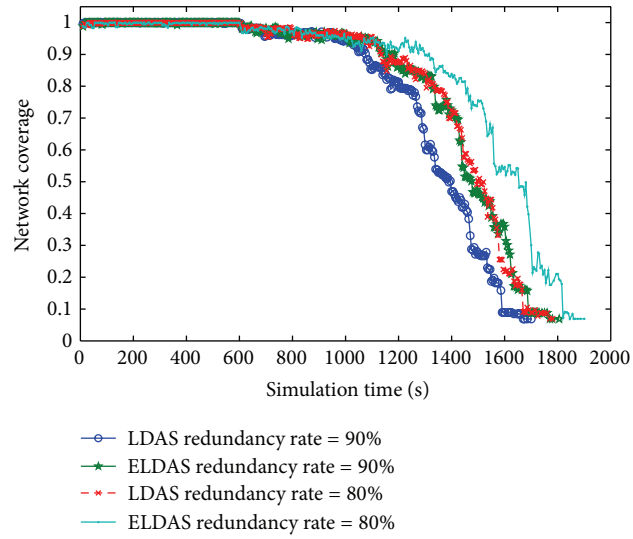


Figure 6: Network coverage with different redundancy rates.

of two aspects: (1) the relationship between the network coverage ratio and network lifetime and (2) showing the network coverage damaged by uneven sleep problem owning to border effects. In the experiments, we randomly deploy 200–300 sensors in two-dimensional terrain with 100 m × 100 m; all sensor nodes have been configured with an sensing area with radius $r = 15$ m. We adopt the energy model in [19] upon which the ratio of energy consumption for a node to transmit or stay idle and asleep is 20 : 4 : 0.01. Let the node sleep time be 10 seconds.

Figure 6 gives the relationship between the simulation time and network coverage.

In Figure 6, there are 200 sensor nodes in the monitoring area. We test the relationship between network lifetime and coverage by setting the node coverage redundancy to be 90% and 80%, respectively. We observe that the coverage ratio of ELDAS is usually better than that of LDAS, for both node coverage redundancy rates.

Figure 7 gives the influence of the number of sensor nodes to network coverage ratio and lifetime, when the node coverage redundancy rate is 90%.

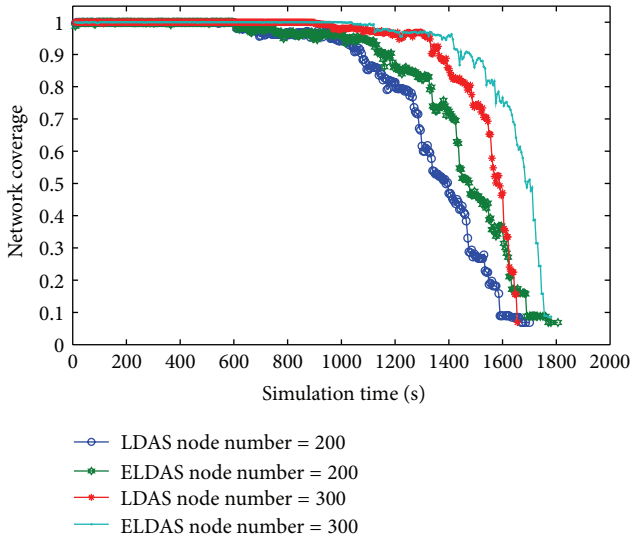


Figure 7: Network coverage with different node numbers.

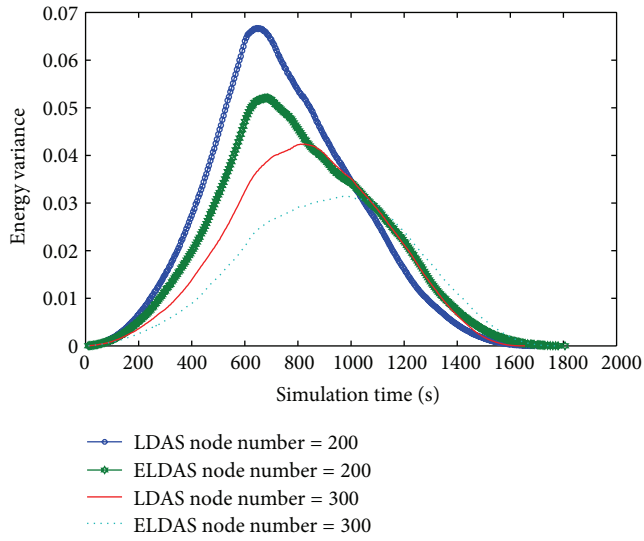


Figure 8: Energy variance.

In order to illustrate the uneven sleep problem in node scheduling, we use the energy consumption variance to measure the energy consumption imbalance.

We use both the mean value and variance of network energy to measure the energy balance status of a node scheduling method at time t:

$$\mathrm{AVG}_E(t) = \frac{1}{N}\sum_{i=1}^{N} E_i(t), \tag{12}$$

$$D_E(t) = \frac{1}{N}\sum_{i=1}^{N}\left(E_i(t) - \mathrm{AVG}_E(t)\right)^2, \tag{13}$$

where N is the number of sensor nodes and $E_i(t)$ is the energy of node S_i at time t.

The energy consumption rate can be measured by the mean energy value, but it is not sufficient because the boundary sensor nodes or center nodes will run out of energy earlier than the rest nodes, whereas the mean energy value remains high. To this end, we also need the energy variance metric to better reflect the energy consumption balance status and validate whether the node scheduling algorithms can handle the coverage holes problem effectively.

Figure 8 compares the network energy consumption variances of ELDAS and LDAS, when the node redundancy ratio is 90%, the radius of the sensing area is $r = 15$ m, and there are 200 and 300 sensor nodes deployed in a monitoring region.

For node energy consumption, the lower the energy consumption variance, the better the network energy consumption balance. In Figure 8, we observe that ELDAS has smaller variance values and thus better energy consumption performance than LDAS. This is because LDAS can lead the boundary nodes to be in a serious uneven sleep problem, whereas ELDAS can effectively alleviate this problem such that the network energy consumption can be balanced, resulting in prolong the network lifetime.

6. Conclusion and Future Work

In this paper, we propose the uneven sleeping problem that happens during coordinated node scheduling in wireless sensor networks. To solve this problem, we analyze the important factors that may cause the uneven sleeping problem. More importantly, we propose a new metric to better model USP and a new node scheduling algorithm. Experiments show that, without influencing the network performance, our proposal can alleviate the uneven sleeping problem, reduce the influence of boundary nodes, and prolong the network lifetime. Although we only test our metric in the node scheduling approach in [14], this new metric applies to other node scheduling algorithms as well.

For the future work, we will research the redefinition of network lifetime. Network lifetime cannot be effectively reflected; people have proposed various such definitions though. In fact, network lifetime is a very comprehensive and systematic issue; yet few people have paid enough attention to this issue. Most node scheduling methods focus on extending the network lifetime, but it is unclear how they define network lifetime and measure the energy-cost balance.

Notations and Definitions

A:	The monitored region
N:	The number of deployed sensor nodes
r:	The sensing radius of sensor
$C(S_i)$:	The circular area centered at given node S_i with radius r
$d(i)$:	The distance to the boundary of monitored region of sensor S_i
$Ef(d(i))$:	The effective coverage area of sensor S_i
$\Omega_r(j)$:	The sub-region j of monitored region.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants no. 61300215, Key Scientific and Technological Project of Henan Province under Grant no. 122102210053, and Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks under Grants no. BM2010577.

References

[1] C.-F. Hsin and M. Liu, "Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 433–442, April 2004.

[2] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 32–41, September 2002.

[3] Y. Shen, X. Yin, H. Chen, W. Qi, and H. Dai, "A density control algorithm for surveillance sensor networks," in *Proceedings of the Wireless Algorithms, Systems, and Applications Conference*, pp. 199–205, 2006.

[4] A. Boukerche, X. Fei, and R. B. Araujo, "An optimal coverage-preserving scheme for wireless sensor networks based on local information exchange," *Computer Communications*, vol. 30, no. 14-15, pp. 2708–2720, 2007.

[5] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 1, no. 1-2, pp. 89–124, 2005.

[6] G. Xing, X. Wang, Y. Zhang et al., "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 1, article 72, 2005.

[7] H. Ling and T. Znati, "Energy efficient adaptive sensing for dynamic coverage in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '09)*, pp. 1–6, April 2009.

[8] S.-F. Jiang, M.-H. Yang, H.-T. Song, and J.-M. Wang, "An enhanced perimeter coverage based density control algorithm for wireless sensor network," in *Proceedings of the 3rd International Conference on Wireless and Mobile Communications (ICWMC '07)*, p. 79, IEEE Computer Society, March 2007.

[9] M. Hefeeda and H. Ahmadi, "A probabilistic coverage protocol for wireless sensor networks," in *Proceedings of the 15th IEEE International Conference on Network Protocols (ICNP '07)*, pp. 41–50, October 2007.

[10] J.-P. Sheu and H.-F. Lin, "Probabilistic coverage preserving protocol with energy efficiency in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 2633–2638, March 2007.

[11] W.-R. Shi, J.-Y. Yuan, and L.-N. Lei, "Research on wireless sensor network coverage control algorithm," *Acta Automatica Sinica*, vol. 35, no. 5, pp. 540–545, 2009.

[12] D. Tian and N. D. Georganas, "Location and calculation-free node-scheduling schemes in large wireless sensor networks," *Ad Hoc Networks*, vol. 2, no. 1, pp. 65–85, 2004.

[13] Y. Gao, K. Wu, and F. Li, "Analysis on the redundancy of wireless sensor networks," in *Proceedings of the 2nd ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '03)*, pp. 108–114, ACM, San Diego, Calif, USA, September 2003.

[14] K. Wu, Y. Gao, F. Li, and Y. Xiao, "Light weight deployment-aware scheduling for wireless sensor networks," *Mobile Networks and Applications*, vol. 10, no. 6, pp. 837–852, 2005.

[15] L.-H. Yen and Y.-M. Cheng, "Range-Based Sleep Scheduling (RBSS) for wireless sensor networks," *Wireless Personal Communications*, vol. 48, no. 3, pp. 411–423, 2009.

[16] C. Ma, H. Shi, G. Yan, and T. Hu, "EeRCCA: an energy efficient range based coverage control algorithm in wireless sensor networks," *Chinese Journal of Sensors and Actuators*, vol. 22, no. 11, pp. 1639–1644, 2009.

[17] O. Younis, M. Krunz, and S. Ramasubramanian, "Location-unaware coverage in wireless sensor networks," *Ad Hoc Networks*, vol. 6, no. 7, pp. 1078–1097, 2008.

[18] G. Fan, R. Wang, H. Huang, L. Sun, and C. Sha, "Coverage-guaranteed sensor node deployment strategies for wireless sensor networks," *Sensors*, vol. 10, no. 3, pp. 2064–2087, 2010.

[19] J. Hill, R. Szewcayk, A. Woo et al., "System architecture directions for networked sensors," *Architectural Support for Programming Languages and Operating Systems*, vol. 35, no. 11, pp. 93–104, 2000.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 987961, 6 pages
http://dx.doi.org/10.1155/2013/987961

Research Article

A Sociability-Based Spray and Forward Scheme for Opportunistic Network

Guoxia Sun,[1] Fu Xiao,[1,2,3] Lingyun Jiang,[1,2,3] Jia Xu,[1,2] and Ruchuan Wang[1,2,3]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China*
[3] *Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education of Jiangsu Province, Nanjing, Jiangsu 210003, China*

Correspondence should be addressed to Fu Xiao; xiaof@njupt.edu.cn

Received 25 June 2013; Accepted 21 September 2013

Academic Editor: Chongsheng Zhang

In opportunistic network, data transmission relies mainly on node mobility. However, the moving patterns of nodes in diverse application scenarios are different. Thus, nodes moving patterns should be considered when designing data transmission strategy. Based on the analysis of node moving characteristics in opportunistic social networks, two parameters, called social rang and social frequency, are introduced and sociability-based Spray and Forward (SSF) scheme is proposed by applying them to the data dissemination and forwarding phase, respectively. Simulation results show that SSF can effectively improve data delivery ratio and reduce latency under the given model.

1. Introduction

As a new type of wireless ad hoc networks, opportunistic network [1] has been mushrooming in recent years. Compared with traditional ad hoc networks, opportunistic network depends mainly on node moving to achieve data transmission. As one kind of opportunistic networks, opportunistic social network [2] is mainly composed of human-carried mobile devices (such as mobile phone, PDA) and achieves data communication by the human movements in their social activities.

Data transmission in opportunistic network is challenging due to the fact that the network is intermittently connected. Practically, the application scenarios of opportunistic network are diverse, for example, wildlife tracking network [3], vehicle network [4], and pocket switched network [5], which results in the fact that moving patterns of nodes are quite different for different scenarios. So node's mobility pattern should be explored when designing data transmission scheme. Intuitively, in human society, people with wide range of social activities can encounter more different people and some people can meet each other regularly because they are socially connected. On network level, we use social range and social frequency to describe the two features of node mobility. Given that there is no significant linear correlation between the two parameters, this paper introduces social range and social frequency in different stages of data forwarding, and proposes a sociability-based Spray and Forward data transmission scheme based on the improvement of Spray and Wait [6].

2. Related Work

In terms of the number of message copies in network, existing data transmission schemes can be categorized into single-copy scheme and multicopy scheme. For single-copy scheme, there is only one message copy for each message at any time instance. Message carrier deletes the message from its memory once it forwards the message to another node. Typical single-copy scheme includes Direct Delivery (DD) [7] and First Contact (FC) [8]. The basic idea of DD is that source node will always carry the message until it delivers the message to the destination it can reduce network overload and energy consumption, but its performance on

delivery ratio and latency are very poor especially in the scenario where source node cannot meet with destination node directly. For FC, by keeping the ID of node it has passed by, message is only forwarded to the node that did not carrier it before. Compared with DD, FC can improve data delivery ratio and shorten delay. Single-copy scheme can reduce traffic redundancy, but its performance on delivery ratio and latency may be very poor.

On the other hand, by increasing the number of message copies in network, multicopy scheme can achieve high delivery ratio. Epidemic [9] is one of the most typical multicopy schemes. Messages are forwarded in a way like database replication. In theory, if network resource is unlimited, with nodes continuous encounters, messages can reach every node. However, owing to the limitation of node storage, message *TTL*, and network bandwidth, the performance of epidemic is very poor in practice. To make a balance between overhead ratio and delivery ratio, researchers proposed Spray and Wait which can reduce the network overhead by presetting the upper bound of message copies as a constant L. Spray and Wait includes two phases: Spray phase and Wait phase. There are two types of forwarding patterns in Spray phase; one type is that message is only sprayed by the source node. In this type, message copies are forwarded to L relay nodes by the source node; the other type is that message copies can be sprayed by both source node and relay nodes. For this type, message carrier sprays half of the message copies to another relay node. Once the number of message copies drops to 1, data transmission steps into Wait phase. In this phase, message is forwarded in DD mode. Spray and Wait can effectively reduce network overload, but message carrier sprays messages to relay nodes without selection in Spray phase, and in Wait phase it has the flaw of DD, which results in its bad performance on delivery ratio especially for opportunistic social network where the mobility patterns of nodes are very different.

In consideration of node mobility, this paper applies social range and social frequency to improve Spray and Wait and proposes a sociability-based Spray and Forward scheme (SSF). SSF includes Spray phase and Forward phase. In Spray phase, message relay nodes are selected based on their social range, in Forward phase message is forwarded to the nodes whose social frequency with destination node is bigger than that of the current carrier.

3. SSF Scheme

In human society, people move based on their social activities, so we cannot simply use RWP (random waypoint) [10] mobile pattern to model nodes mobility. Intuitively, in people's daily life, some people have a wide range of social activities; they can meet a variety of people; some people meet each other regularly because they are socially connected for work, friendship, and so on. Figure 1 illustrates the social features (normalization value) of some nodes in pmtr [11] dataset; we can see that there is a significant difference between the two parameters, so instead of integrating them into a single value, we use them in different phase to forward message, which is different from existing work [12].

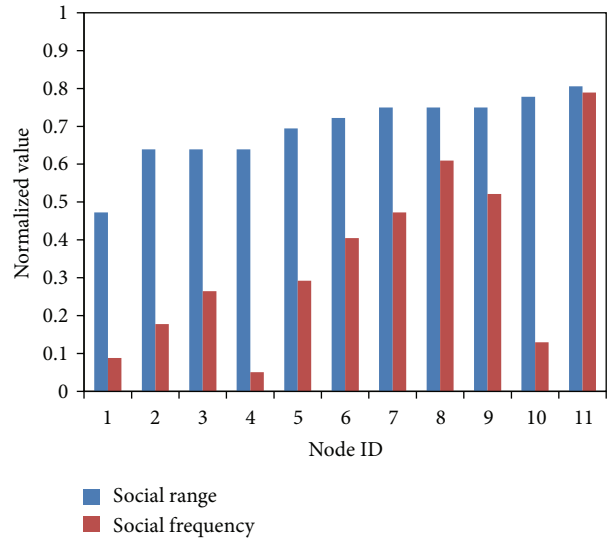


Figure 1: Social features of nodes in pmtr dataset.

3.1. System Model and Assumptions. In human daily life, people's activities are based on geocommunity, such as a classroom, a lab, or a dormitory for a student. Based on the roles people play, people can engage in various communities, so we give the following network model and related definition.

(1) The mobility of a node is based on the mobility of the user who carries the node.

(2) A node can engage in one or more communities; that is, for some inert nodes, they only appear in limited communities, but for some active nodes, they can be active in more communities.

(3) In this paper, the area of a geocommunity is small (like a lab, a classroom); individuals in the same community can contact directly in a face-to-face way, so the source and the destination of messages generally belong to different communities.

(4) Nodes belonging to a single community move according to the Cluster movement model.

(5) Nodes belonging to multiple communities move according to the Map Route movement model.

A. Social Range. Social range reflects the size of people's circle of friendship in social life, and we define the social range of a node as the number of nodes it can directly encounter. Suppose that the collection of nodes that A can directly meet is $M(A)$ and its size is $|M(A)|$; then the social range of A is defined as the following:

$$\mathrm{SR}(A) = |M(A)|. \quad (1)$$

B. Social Frequency. Social frequency refers to the frequency of encounter between two nodes. The more they meet each

other, the better opportunity they have to communicate with each other. Assuming that the number of times node A encounters B is $N(A, B)$, the social frequency between A and B, $\mathrm{SF}(A, B)$ can be defined as the following:

$$\mathrm{SF}(A, B) = N(A, B). \quad (2)$$

3.2. SSF Strategy. Based on the above definitions and Spray and Wait scheme, this paper puts forward an improved data transmission scheme, called SSF (sociability-based Spray and Forward), for opportunistic social network. SSF makes forwarding decision based on the local encounter information of nodes. The main idea of SSF is to spread out the message from local area as far as possible by spraying message to nodes which have a wider social range in Spray phase and to achieve a rapid-delivery of message it forwards message to nodes which have a higher social frequency with destination. A detailed exposition of SSF is given below.

A. Spray Phase

(1) When source node S generates a message M whose destination is D, the related number of message copies is set as L in the network.

(2) Assume that S meets node C. If the social range of C is bigger than that of S; that is, $\mathrm{SR}(C) > \mathrm{SR}(S)$, S will spray message M to C, and the related number of message copies in C is set as k, k can be set as $(\mathrm{SR}(C)/(\mathrm{SR}(C) + \mathrm{SR}(S)) * L$. For simplicity, k is set as 1 in this paper. Otherwise, S will not spray message to C.

(3) Message carrier repeats Step (2) until there is only one message copy left in its buffer; data transmission steps into Forward phase.

B. Forward Phase

(4) Suppose that the current message carrier C meets node E; if there is $\mathrm{SF}(E, D) > \mathrm{SF}(C, D)$; that is, the social frequency between E and D is higher than that of C and D, C will forward M to E and delete the message from its own buffer. Otherwise, C will not forward message M.

(5) Repeat Step (4) on E and the following relay nodes until M is delivered to destination D or deleted from the message buffer due to TTL or buffer overflow.

Compared with Spray and Wait, the advantages of SSF can be concluded as follows.

(1) As is well known, the friend circle of people in the same community largely overlaps. Similarly, the encounters of nodes located in the same geocommunity also largely overlap during a period of time; Spray and Wait makes node continuously spray message copies to its encounter nodes without selection which makes its Spray phase inefficient. Given that nodes with large social range will have a big chance to carry message to different communities, SSF will select the nodes whose social range is larger as message carrier in Spray phase which can avoid invalid spray to some extent.

(2) In Forward phase, SSF chooses nodes that have a better social frequency with destination to forward message, so it can achieve rapid delivery and avoid the weakness of Spray and Wait in which message carrier will wait to encounter destination to delivery message.

Table 1: Simulation parameters.

Parameter	Value
Transmit speed	250 kBps
Transmit range	5 m
Buffer size	5 MB
Node mobile speed	0.5 m–1 m
Wait time	10 s–30 s
Message TTL	∞
Message size	50 kB–100 kB
Cluster range	40 m
Number of message copies	4
Spray mode	Nonbinary

4. Simulation and Evaluation

4.1. Simulation Scenarios and Parameter Settings. Simulation is made on ONE1.4.1 [13] which is a typical simulation platform for DTN (Delay and Tolerant Network). There are 6 geocommunities in this simulation, and nodes move among these communities. The activity area of some nodes is limited; they can only access few communities and even move in a single community. For other nodes, their activity areas can be very wide; they can access more communities. The above activity patterns are also common in human real social life. Moreover, the geocommunity in this paper refers to some small venues like a lab, a students' dormitory, or a classroom in school, rather than some wide areas like an overall campus and its public areas. Nodes move according to Custer movement model within a community and move according to Map Route movement model among communities. Related simulation parameters are shown in Table 1.

4.2. Simulation Results and Analysis

4.2.1. Performance of Each Protocol under Varying Simulation Time. In our experiment, there are two message generating events, one generates a message every 5 min and the other generates a message every 10 min. Figure 2 shows that, compared with the other three forwarding algorithms, SSF achieves the highest delivery ratio all the time. As is well known, the rational usage of multicopy mechanism can help to improve the delivery ratio; SSF not only uses multicopy mechanism but also makes better use of nodes social features in message transmission, which makes its delivery ratio much higher than the other schemes. FC is one kind of single-copy forwarding schemes, but it forwards message to all the nodes which have not ever carried the message; thus, it can make message go through almost all of the nodes; meanwhile,

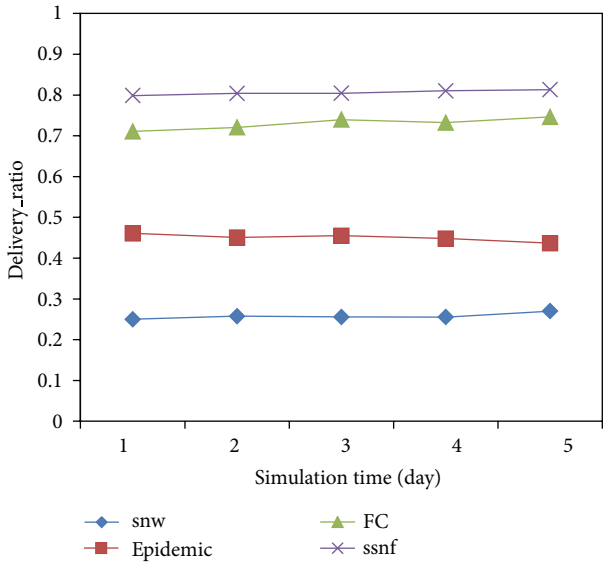


Figure 2: Delivery ratio versus simulation time.

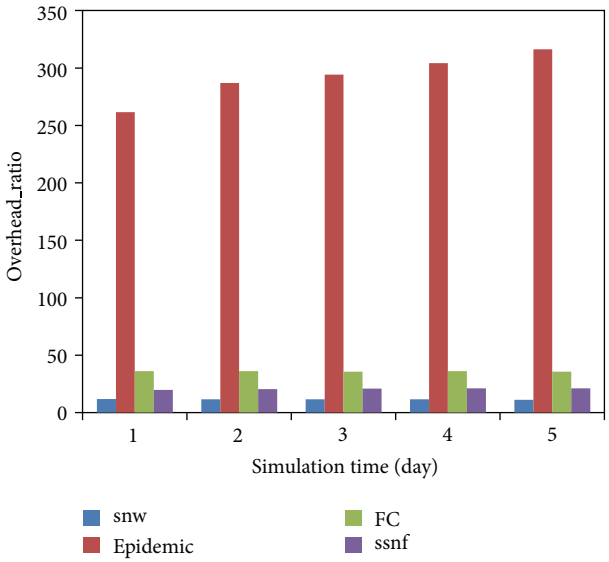


Figure 3: Overhead ratio versus simulation time.

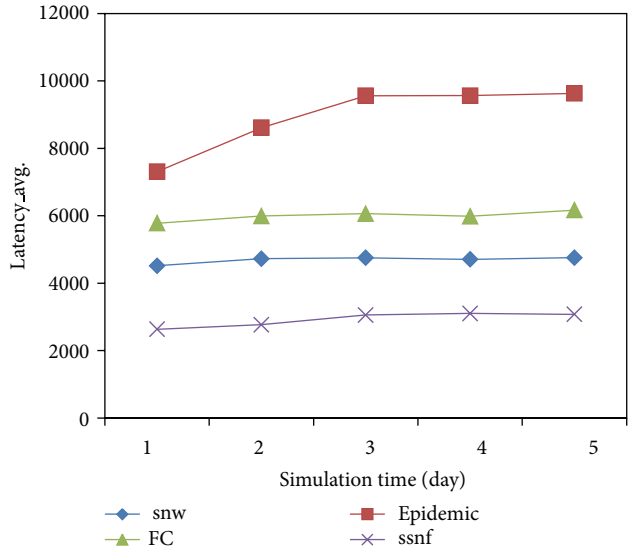


Figure 4: Average latency versus simulation time.

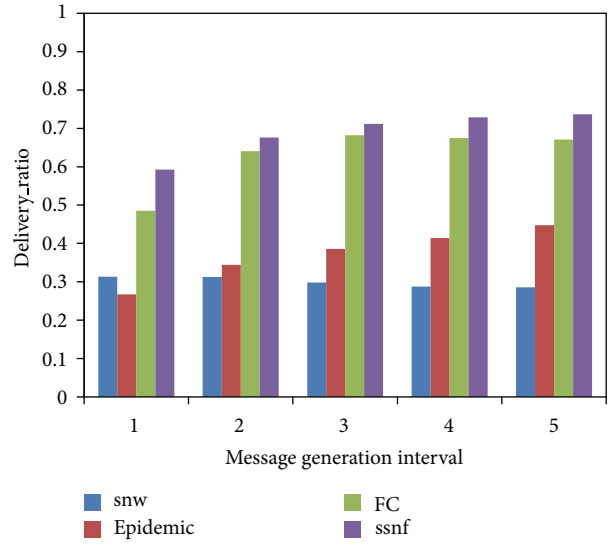


Figure 5: Delivery ratio versus message generation interval.

because of the unlimited message *TTL*, it can also achieve high delivery ratio. For epidemic, because node forwards message in a flooding way, the delivery ratio will be very low when message buffer is limited. Although Spray and Wait limits the number of message copies in the network, its blind spray and passive waiting for destination node make its performance on delivery ratio low.

We also compared overhead ratio in Figure 3. It shows that, under the default settings, the overhead ratio of SSF is the lowest besides Spray and Wait. Because the message forwarding opportunity is very small in Wait phase, the overhead ratio of Spray and Wait is lower than that of SSF. Compared with the above two algorithms, FC uses a single-copy mechanism to forward data, but it enables messages to traverse most of the nodes, so its overhead ratio is higher than the above two algorithms. Due to its flooding strategy, epidemic has the highest overhead ratio.

In addition, we also observed the average latency of every algorithm over time. Message delivery delay includes queuing scheduling delay, sending delay, and transmission delay. Figure 4 shows that the performance of SSF on average latency is the best, that is, because of its wise usage of nodes sociability on message forwarding.

4.2.2. Performance of Each Protocol under Varying Message Generation Interval. In this experiment, the simulation time is set as 2 days.

In Figure 5, we can see that the delivery ratio of SSF is always the highest when the message generation rate changes. Except for Spray and Wait, the delivery ratio of each protocol

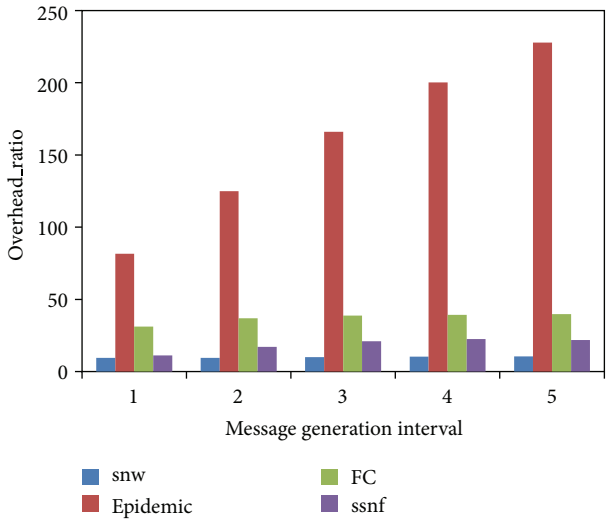


Figure 6: Overhead ratio versus message generation interval.

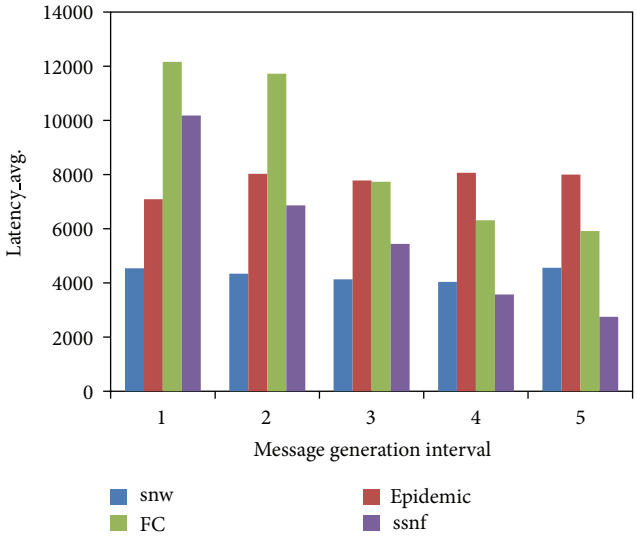


Figure 7: Average latency versus message generation interval.

increases as the message generation interval rises. It is because that the more available buffer there is in the network, the fewer messages are dropped. In Spray and Wait, there are only few messages that can be forwarded, so its delivery ratio keeps stable and low.

From Figure 6, we can clearly observe that the overhead ratio of epidemic is also the heaviest. Spray and Wait has very low overhead ratio that is also because of its slim message forwarding opportunity. Although FC is one kind of single-copy forwarding schemes, it can forward more messages when the message *TTL* is unlimited, so its overhead ratio is higher than that of SSF. SSF can deliver more messages than Spray and Wait, so its overhead ratio is higher than that of Spray and Wait.

We also plot out the average latency of each protocol in Figure 7. It shows that the average latency of FC and SSF changes significantly as the message generation interval changes while Spray and Wait and epidemic keep stable. In general, the average latency of SSF is lower than epidemic and FC.

5. Conclusion

Aiming to improve the message delivery ratio and reduce the latency, this paper proposes a sociability-based Spray and Forward scheme for data transmission based on the node's social characteristics. First, we analyze the node's social characteristics in opportunistic social network and by applying them to improve Spray and Wait propose SSF. Experimental results show that SSF can significantly increase the message delivery ratio and reduce the delivery latency. However, this paper uses a relatively simple approach to analyze node's social characteristics; in the future, we can deeply study node's social characteristics and further improve SSF to make it more practical and intelligent.

Acknowledgments

This work is sponsored by the National Natural Science Foundation of China (61373137, 61373017, 61171053, and 61100199), the Scientific & Technological Support Project of Jiangsu (BE2012755), the Natural Science Foundation of Jiangsu (BK2011755), and the Scientific Research & Industry Promotion Project for Higher Education Institutions (JHB2012-7).

References

[1] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella, "Human mobility models for opportunistic networks," *IEEE Communications Magazine*, vol. 49, no. 12, pp. 157–165, 2011.

[2] A. K. Pietilänen and C. Diot, "Dissemination in opportunistic social networks: the role of temporal communities," in *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, pp. 165–174, ACM, 2012.

[3] S. Ehsan, M. Brugger, K. Bradford, B. Hamdaoui, and Y. Kovchegov, "Sufficient node density conditions on delay-tolerant sensor networks for wildlife tracking and monitoring," in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference: Energizing Global Communications (GLOBECOM '11)*, pp. 1–6, December 2011.

[4] U. G. Acer, P. Giaccone, D. Hay, G. Neglia, and S. Tarapiah, "Timely data delivery in a realistic bus network," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1251–1265, 2012.

[5] S. Wang, M. Liu, X. Cheng, and M. Song, "Routing in pocket switched networks," *IEEE Wireless Communications*, vol. 19, no. 1, pp. 67–73, 2012.

[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the Conference on Computer Communications (SIGCOMM '05)*, pp. 252–259, ACM, August 2005.

[7] Y. Wang, H. Dang, and H. Wu, "A survey on analytic studies of Delay-Tolerant mobile sensor networks," *Wireless Communications and Mobile Computing*, vol. 7, no. 10, pp. 1197–1208, 2007.

[8] K. Massri, A. Vernata, and A. Vitaletti, "Routing protocols for delay tolerant networks: a quantitative evaluation," in *Proceedings of the 7th ACM workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, pp. 107–114, ACM, 2012.

[9] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Duke University, Durham, NC, USA, 2000.

[10] A. Ahuja, K. Venkateswarlu, and P. V. Krishna, "Stochastic characteristics and simulation of the random waypoint mobility model," http://arxiv.org/abs/1203.3920. In press.

[11] M. Paolo and G. Sabrina, "Crawdad metadata: unimi/pmtr [EB/OL]," 2008, http://www.crawdad.org/meta.php?name=unimi/pmtr.

[12] R. Verdone and F. Fabri, "Sociability based routing for environmental opportunistic networks," *Advances in Electronics and Telecommunications*, vol. 1, no. 1, pp. 98–103, 2010.

[13] A. Keranen, J. Ott, and T. Karkkainen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools '09)*, vol. 55, p. 10, Brussels, Belgium, 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 217180, 7 pages
http://dx.doi.org/10.1155/2013/217180

Research Article

Fast Endmember Extraction for Massive Hyperspectral Sensor Data on GPUs

Zebin Wu,[1,2] Shun Ye,[1] Jie Wei,[1] Zhihui Wei,[1,3] Le Sun,[1] and Jianjun Liu[1]

[1] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*
[2] *Lianyungang Research Institute of NJUST, Lianyungang 222006, China*
[3] *Jiangsu Key Lab of Spectral Imaging and Intelligent Sensing, Nanjing 210094, China*

Correspondence should be addressed to Zebin Wu; zebin.wu@gmail.com

Received 14 July 2013; Accepted 12 September 2013

Academic Editor: Zhijie Han

Hyperspectral imaging sensor becomes increasingly important in multisensor collaborative observation. The spectral mixture problem seriously influences the efficiency of hyperspectral data exploitation, and endmember extraction is one of the key issues. Due to the high computational cost of algorithm and massive quantity of the hyperspectral sensor data, high-performance computing is extremely demanded for those scenarios requiring real-time response. A method of parallel optimization for the well-known N-FINDR algorithm on graphics processing units (NFINDR-GPU) is proposed to realize fast endmember extraction for massive hyperspectral sensor data in this paper. The implements of the proposed method are described and evaluated using compute unified device architecture (CUDA) based on NVIDA Quadra 600 and Telsa C2050. Experimental results show the effectiveness of NFINDR-GPU. The parallel algorithm is stable for different image sizes, and the average speedup is over thirty times on Telsa C2050, which satisfies the real-time processing requirements.

1. Introduction

Multisensor image data fusion in remote sensing is a kind of collaborative image processing technology for sensor networks, which utilize the consistency and complementarity of different sensors' image data to assess accurately [1]. Hyperspectral imaging sensor becomes increasingly important in multisensor collaborative observation. There are tens or hundreds of contiguous bands of high spectral resolution in hyperspectral image, covering the visible, near-infrared, and shortwave infrared spectral bands [2]. It can get the spectral signatures and enable identification of the materials that make up a scanned target, which greatly improve the ability of target recognition and detection in sensor networks.

Spatial resolution of hyperspectral sensor data is often relative low to meters or tens of meters, and several different materials jointly occupy a single pixel. Therefore, most of the pixels of hyperspectral data, which called mixed pixels, contain more than one material (called endmember) [3]. The mixture problem will seriously influence the efficiency of hyperspectral data exploitation. Many researchers focus on the study of hyperspectral unmixing. Endmember extraction is one of the key issues. Many algorithms have been developed to solve this issue, and the N-FINDR [4] is one of the most widely used methods for automatically determining endmembers in hyperspectral data without using a priori information, which has been successfully applied for over ten years.

There are two factors which cause the limitation of N-FINDR's applications in multisensor collaborative observation. First, it is quite expensive in computational terms due to the high algorithm complexity. Second, the quantity of hyperspectral sensor data is massive due to the extremely high dimensionality of hyperspectral data cube. For example, the airborne visible-infrared imaging spectrometer (AVIRIS) is able to record the visible and near-infrared spectrum (wavelength region 400–2500 nm) of reflected light in an area, 2–12 km wide and several kilometers long, using 224 spectral bands. The resulting multidimensional data cube typically comprises several GBs per flight [5]. High-performance computing is extremely demanded for those scenarios requiring real-time response, such as military detection, monitoring of chemical contamination, and wildfire tracking.

Several parallel computing technologies, like supercomputers, clusters, distributed computing, multicore CPUs, field-programmable gate arrays (FPGAs), and graphics processing units (GPUs), are used to accelerate hyperspectral data processing algorithms [6, 7]. GPUs are quickly evolving as a standardized architecture in hyperspectral processing due to their compactness, low cost, portability, low weight, and high computational power [8]. Wu et al. [9] presented an improved GPU implementation of the PPI algorithm which provides real-time performance. Plaza et al. [10] developed three new GPU-based implementations of endmember extraction algorithms: the pixel purity index (PPI), a kernel version of the PPI (KPPI), and the automatic morphological endmember extraction (AMEE) algorithm, and they provided a GPU-based implementation of the fully constrained linear spectral unmixing algorithm. Barberis et al. [6] proposed a new parallel implementation of the vertex component analysis (VCA) algorithm for spectral unmixing of remotely sensed hyperspectral data on commodity GPUs. Although there are several parallel implements of the N-FINDR algorithm existing in literature [5, 11–13], the speedup of them is less than 30 times, which cannot meet the requirements of real-time applications well.

In this paper, we propose a method of parallel optimization for N-FINDR on graphics processing units (NFINDR-GPU) to realize real-time endmember extraction for massive hyperspectral sensor data. The implements of the proposed methods using compute unified device architecture (CUDA) are described and evaluated. The computation time of the parallel implementation on GPUs is compared with the serial implementation on central processing units (CPUs).

2. Endmember Extraction for Hyperspectral Sensor Data

There are two models to unmix the hyperspectral sensor data, named linear and nonlinear. The linear mixture model identifies a collection of spectrally pure constituent spectra (endmembers) and expresses the measured spectrum of each mixed pixel as a linear combination of endmembers weighted by fractional abundances that indicate the proportion of each endmember present in the pixel [3]. It assumes minimal secondary reflections and multiple scattering effects in the data collection procedure, and hence the measured spectra can be expressed as a linear combination of the spectral signatures of materials present in the mixed pixel. The linear mixture model is formulated as follows:

$$y = M\alpha + n, \tag{1}$$

where y denotes a L-by-1 spectrum vector of one pixel of the observed hyperspectral data; L denotes the number of bands, $M = [m_1, m_2, \ldots, m_q]$ denotes a L-by-q mixing matrix with endmembers as columns and is usually of full column rank, q denotes the number of endmembers; $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_q]^T$ denotes a q-by-1 vector containing the respective fractional abundances of the endmembers, α_k is the abundance fraction of the kth endmember, with $k = 1, 2, \ldots, q$, and the notation $(\cdot)^T$ stands for vector transposed; n denotes an additive L-by-1 noise vector collecting the errors affecting the measurements of the pixel at each spectral band. Endmember extraction of hyperspectral data aims at obtaining a good estimation of the mixing matrix M.

Several methods have been used to perform endmember extraction, including geometrical, statistical, and sparse regression-based approaches [3]. A successful and widely used algorithm in the first category has been the N-FINDR.

N-FINDR algorithm is an iterative optimization procedure that maximizes the volume of a simplex containing all hyperspectral image pixels in feature space and automatically extracts the endmembers in the hyperspectral scene [4]. It relies on the assumption that, when the noise vector n is negligible, all the spectrum vectors of hyperspectral pixels are contained in a convex set (named simplex) of high-dimensional space, and the endmembers are vertices of the simplex. Thus, the problem of endmember extraction is transformed to solving the vertices of the simplex.

The N-FINDR looks for the set of pixels with the largest possible volume by inflating a simplex inside the data. The mathematical definition of the volume V of a simplex is formulated as

$$V(E) = \frac{1}{(n-1)!}\operatorname{abs}(|E|), \tag{2}$$

where E is the matrix of endmembers augmented with a row of ones, $E = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e_1 & e_2 & \cdots & e_n \end{bmatrix}$; e_i is a column vector containing the spectra of endmember i; $\operatorname{abs}(x)$ is the absolute value of x; $|\cdot|$ denotes the determinant of matrix; $(n-1)$ is the number of dimensions occupied by the data.

The determinant is only defined in the case where the number of features is $q - 1$; q is the number of desired endmembers. Since $n \gg p$ typically in hyperspectral data, we adopt principal component analysis (PCA) to reduce the dimensionality of the input hyperspectral data [5] and use the virtual dimensionality (VD) method [14] to estimate the number of endmembers, for the preprocessing of N-FINDR. After the preprocessing, the standard N-FINDR algorithm can be summarized as in Algorithm 1.

3. Fast Endmember Extraction Based on GPUs

It could be noted that there are three factors which lead to high computational overhead of the N-FINDR algorithm. Firstly, the algorithm is an iterative procedure, and every pixel in the data set must be evaluated to refine the estimate of endmembers, looking for the set of pixels that maximizes the volume of the simplex defined by the selected endmembers [5]. Secondly, in step (4), the computation is done for every single element in the input data set, and the replacement step is repeated for all the pixel vectors in the dataset. Thirdly, in step (3), the calculation of the determinants is particularly time consuming.

Aiming at solving these problems, the endmember extraction based on N-FINDR is optimized using compute unified device architecture (CUDA) on GPUs as shown in Algorithm 2.

The NFINDR-GPU is graphically illustrated by a flowchart in Figure 1.

(1) Randomly initialize the set of q endmembers $E^{(0)}$ from the dimensionality reduced hyperspectral data. $E^{(0)} = \left\{e_1^{(0)}, e_2^{(0)}, \ldots, e_q^{(0)}\right\}$, and $k = 0$;
(2) **repeat**
(3) Calculate the volume of the current endmember set $E^{(k)}$

$$V(E^{(k)}) = \frac{1}{(q-1)!}\text{abs}\left(\left|\begin{bmatrix} 1 & 1 & \cdots & 1 \\ e_1^{(k)} & e_2^{(k)} & \cdots & e_q^{(k)} \end{bmatrix}\right|\right);$$

(4) For every pixel spectrum vector x_i ($1 \le i \le n$, n is the number of pixels) in the dimensionality reduced data set, test this pixel in all the q endmember positions in the $E^{(k)}$ set and recalculate the volume $V(E^{(k)})$;
(5) If the new volume is larger than the previous one, then update the $E^{(k)}$ set to $E^{(k+1)}$, and $k = k + 1$, then go to step (3). If the new volume is not larger than the previous one, then no replacement is required;
(6) **until** the endmember set converges and no new replacement takes place.

Algorithm 1: N-FINDR.

(1) Randomly initialize the set of q endmembers $E^{(0)}$ from the dimensionality reduced hyperspectral data. Set $E^{(0)} = \left\{e_1^{(0)}, e_2^{(0)}, \ldots, e_q^{(0)}\right\}$, and $k = 0$;
(2) Calculate the volume of the endmember set $E^{(0)}$

$$V(E^{(0)}) = \frac{1}{(q-1)!}\text{abs}\left(\left|\begin{bmatrix} 1 & 1 & \cdots & 1 \\ e_1^{(0)} & e_2^{(0)} & \cdots & e_q^{(0)} \end{bmatrix}\right|\right);$$

(3) **repeat**
(4) Transmit the data set and the set of q endmembers $E^{(k)}$ to the global memory of GPU;
(5) Calculation and Replacement: create n threads (n is the number of pixels in data set), each thread test a pixel in all the q endmember positions in the $E^{(k)}$ and recalculate the volume $V(E^{(k)})$. Details of this step are listed below:
(5.1) set $i = 1$;
(5.2) **repeat**
(5.3) Create n threads for all the pixels;
(5.4) Each thread reads $E^{(k)}$ from the global memory of GPU to their own registers respectively, and replace $e_i^{(k)}$ in $E^{(k)}$ with the corresponding pixel vector of the thread (GPU Kernel);
(5.5) Calculate the volume of the current endmember set $E'^{(k)}$, and store $V(E'^{(k)})$ in the global memory (GPU Kernel);
(5.6) copy the volumes of all threads to internal memory, and find out the largest volume $V_{\max}(E'^{(k)})$;
(5.7) if $V_{\max}(E'^{(k)}) > V(E^{(k)})$, update $k = k + 1$, $V(E^{(k)}) = V_{\max}(E'^{(k)})$, and $E^{(k)} = E'^{(k)}$;
(5.8) $i = i + 1$;
(5.9) **until** $i = q$;
(6) **until** the endmember set converges and no new replacement takes place.

Algorithm 2: Fast endmember extraction based on N-FINDR on GPUs (NFINDR-GPU).

4. Experiment

A well-known real hyperspectral scene labeled as f970619t01p02_r02, collected by the airborne visible infrared imaging spectrometer (AVIRIS) over the Cuprite mining district in Nevada [15], is used to evaluate the performance of NFINDR-GPU. In order to test the performance of the algorithm on different magnitude, the portions used in experiments correspond to 5 subsets of the scene, including a 250 × 191-pixel subset, a 300 × 300-pixel subset, a 350 × 350-pixel subset (see Figure 2), a 512 × 614-pixel subset, and a 2206 × 614-pixel subset. The scene comprises 224 spectral bands between 0.4 and 2.5 μm, with nominal spectral resolution of 10 nm. Prior to the analysis, bands 1–3, 105–115, 150–170, and 221–224 were removed due to water absorption and low SNR in those bands. The site is well understood mineralogically and has several exposed minerals of interest including alunite, buddingtonite, calcite, kaolinite, and muscovite [5]. Reference ground signatures of the above minerals are available in the form of a U.S. geological survey library (USGS) [16]. The number of endmember q was estimated to be 16 after calculating the VD of the AVIRIS Cuprite data.

The GPU platforms used to test our parallel algorithm are the NVidia Quadra 600 and Tesla C2050. The former features 96 processor cores, total dedicated memory of 1 GB, and memory bandwidth of 25.6 GB/s. The latter features

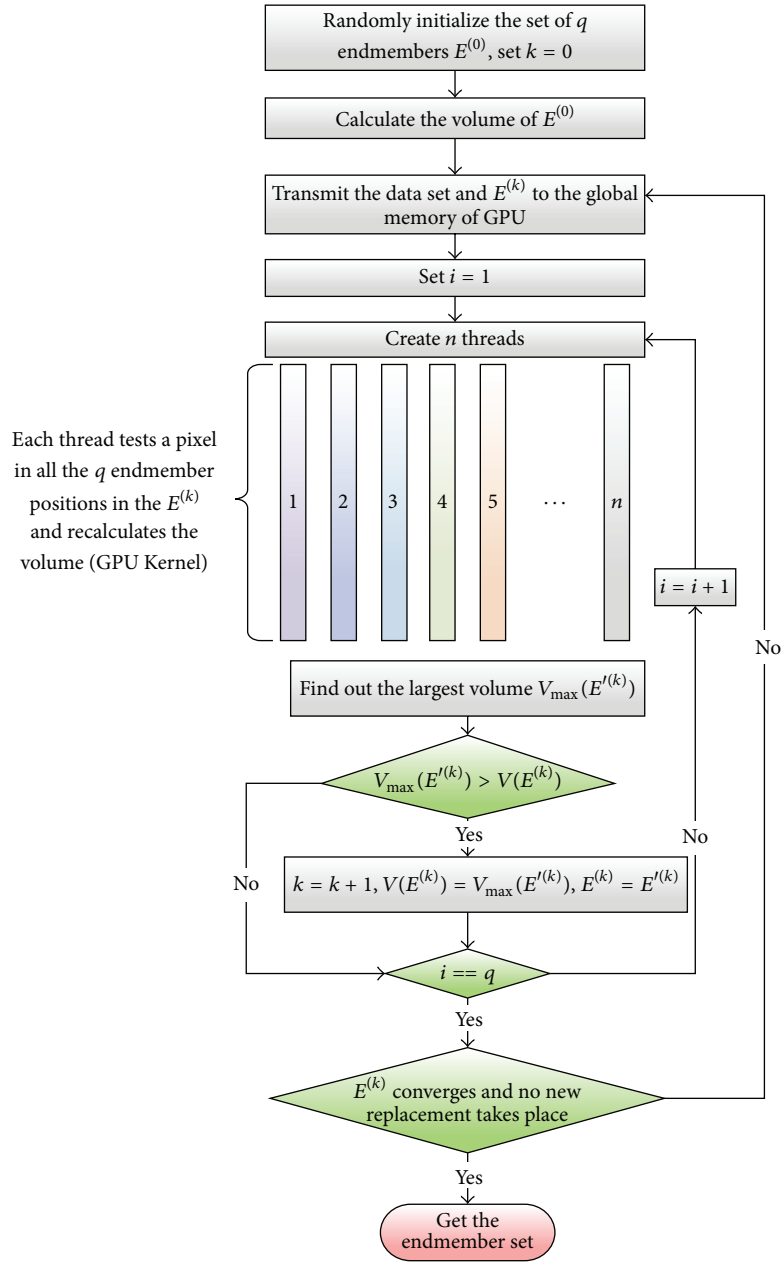


Figure 1: Flowchart of NFINDR-GPU.

448 processor cores operating at 1.15 GHz, with single precision floating point performance of 1030 Gflops, double precision floating point performance of 515 Gflops, total dedicated memory of 3 GB, 1.5 GHz memory (with 384-bit GDDR5 interface), and memory bandwidth of 144 GB/s. Both of the GPUs are connected to an Intel Xeon E5603 CPU at 1.6 GHz with 4 cores, with 8 GB RAM. The tests were performed on 64 b Microsoft Windows 7 operating system.

In order to achieve a fair benchmark in terms of execution time with respect to the GPU version, we firstly realize a *C* serial implementation of the N-FINDR algorithm as a basis for the subsequent parallel implementation. The serial algorithm is executed in one of the available CPU cores, and the parallel time is measured in the considered GPU platform. For each test, 60 runs are performed and the mean values are reported. The experimental results indicate that the endmember extraction results of both parallel and serial versions of N-FINDR correspond to the published ground truth very well. While using the same initialization endmember set $E^{(0)}$, the serial and parallel algorithms reach the same maximum volume with the same iteration times. The main difference between them is the time they need to complete their calculations. The average run time per iteration and the speedup of the serial and parallel implementations are summarized in Table 1 and Figure 3.

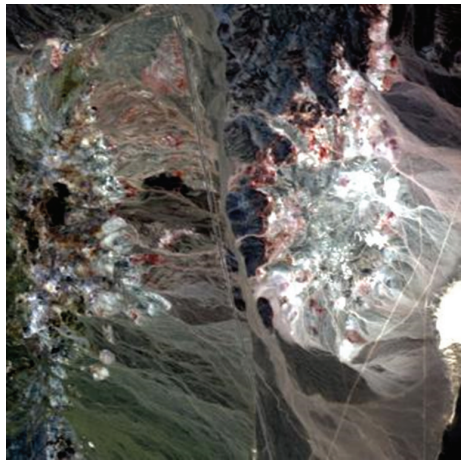


Figure 2: A 350 × 350-pixel subset of the AVIRIS Cuprite scene.

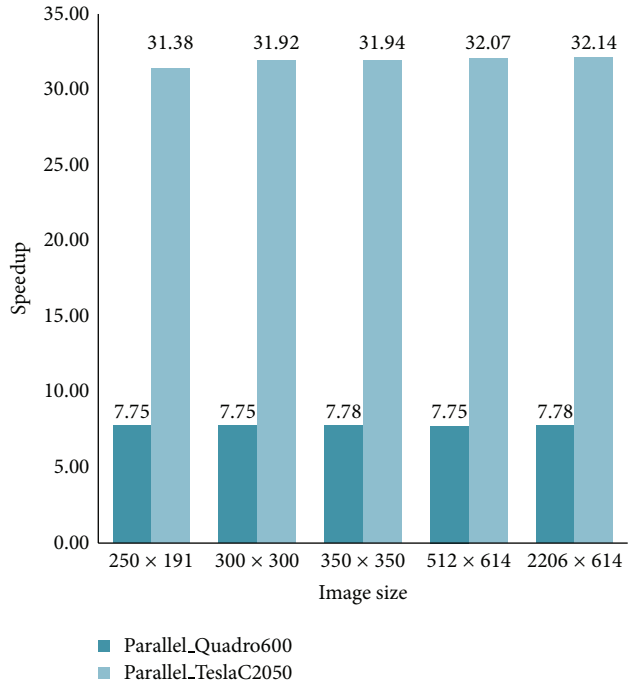


Figure 3: The average speedup of the parallel implementations of N-FINDR.

Table 1: Average run time and speedup of the serial and parallel implementations of N-FINDR.

Data size	Data: f970619t01p02_r02_sc04.rfl				
	Serial	Parallel_Quadro600		Parallel_TeslaC2050	
	Time (sec)	Time (sec)	Speedup (X)	Time (sec)	Speedup (X)
250 × 191	21.184	2.732	7.75	0.675	31.38
300 × 300	39.735	5.124	7.75	1.245	31.92
350 × 350	54.229	6.969	7.78	1.698	31.94
512 × 614	138.152	17.827	7.75	4.308	32.07
2206 × 614	596.901	76.735	7.78	18.572	32.14

We can conclude that the parallel implementations are stable for different image sizes, even for the big size of 2206 × 614 pixels, and NFINDR-GPU on Tesla C2050 achieved a significant speedup of greater than 30 times with regard to the CPU-based serial version of the N-FINDR algorithm. The proposed method shows better performance than the methods proposed in the existing literature. As [9] reports, while the AVIRIS scanning rate is 12 Hz, more recent satellite hyperspectral sensors such as Hyperion feature 220 Hz cross-line scanning rates, which means that a hyperspectral sensor data like the AVIRIS Cuprite scene (a typical AVIRIS data cube with 614 × 512 pixels and 224 spectral bands) could be collected in about 5 s. The achieved processing time on the considered GPU architecture satisfies the real-time processing requirements.

5. Conclusions

Hyperspectral sensor can get the spectral signatures and enable identification of the materials that make up a scanned target, which will greatly improve the ability of target detection and recognition in sensor networks. Endmember extraction is one of the key issues for hyperspectral application in multisensor collaborative observation.

The N-FINDR is one of the most widely used and successfully applied methods for endmember extraction. But it suffers from long execution time due to its high algorithm complexity and the massive quantity of hyperspectral sensor data. High-performance computing is extremely demanded for those scenarios requiring real-time response, such as military detection, monitoring of chemical contamination, and wildfire tracking.

Improved parallel optimization of N-FINDR using the compute device unified architecture (CUDA) on GPUs (NFINDR-GPU) is proposed for fast endmember extraction of hyperspectral sensor data. The algorithm is implemented on both NVIDIA Quadra 600 and Telsa C2050, achieving significant improvements when compared with the previous GPU-based implementations of N-FINDR. Experimental results, based on the hyperspectral data collected by AVIRIS hyperspectral imaging sensor, show the effectiveness of NFINDR-GPU. The parallel implementation is stable for different image sizes, and the NFINDR-GPU on NVIDA Tesla C2050 achieved a significant speedup of greater than 30 times with regard to the CPU-based serial version of N-FINDR, which satisfies the real-time processing requirements. Future work will focus on the comparison with other parallel methods.

Acknowledgments

Financial support for this work, provided by the National Natural Science Foundation of China (Grants nos. 61101194, 61071146), the Jiangsu Provincial Natural Science Foundation of China (Grant no. BK2011701), the Research Fund for the Doctoral Program of Higher Education of China (Grant no. 20113219120024), the Jiangsu Province Six Top Talents project of China (Grant no. WLW-011), the CAST Innovation Foundation (Grant no. CAST201227), and the Project of China Geological Survey (Grant no. 1212011120227), is gratefully acknowledged.

References

[1] P. Lamborn and P. J. Williams, "Data fusion cm a distributed heterogeneous sensor network," in *Proceedings of the International Society for Optical Engineering (SPIE '06)*, April 2006.

[2] R. M. Cavalli, L. Fusilli, S. Pascucci, S. Pignatti, and F. Santini, "Hyperspectral sensor data capability for retrieving complex urban land cover in comparison with multispectral data: venice city case study," *Sensors*, vol. 8, no. 5, pp. 3299–3320, 2008.

[3] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente et al., "Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–3379, 2012.

[4] M. E. Winter, "N-FINDR: an algorithm for fast autonomous spectral endmember determination in hyperspectral data," in *Proceedings of the Imaging Spectrometry V (SPIE '99)*, M. R. Descour and S. S. Shen, Eds., vol. 3753, pp. 266–275, 1999.

[5] S. Sanchez, R. Ramalho, L. Sousa, and A. Plaza, "Real-time implementation of remotely sensed hyperspectral image unmixing on GPUs," *Journal of Real-Time Image Processing*, 2012.

[6] A. Barberis, G. Danese, F. Leporati, A. Plaza, and E. Torti, "Real-time implementation of the vertex component analysis algorithm on GPUs," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 2, pp. 251–255, 2013.

[7] C. A. Lee, S. D. Gasster, A. Plaza, C.-I. Chang, and B. Huang, "Recent developments in high performance computing for remote sensing: a review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 3, pp. 508–527, 2011.

[8] A. Plaza, Q. Du, Y.-L. Chang, and R. L. King, "High performance computing for hyperspectral remote sensing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 3, pp. 528–544, 2011.

[9] X. Wu, B. Huang, A. Plaza, Y. Li, and C. Wu, "Real-time implementation of the pixel purity index algorithm for endmember identification on GPUs," *IEEE Geoscience and Remote Sensing Letters*. In press.

[10] A. Plaza, J. Plaza, and S. Sánchez, "Parallel implementation of endmember extraction algorithms using nvidia graphical processing units," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '09)*, vol. 5, pp. V208–V211, July 2009.

[11] W. Luo, "Parallel implementation of N-FINDR algorithm for hyperspectral imagery on hybrid multiple-core CPU and GPU parallel platform," in *Proceedings of the Remote Sensing Image Processing, Geographic Information Systems, and Other Applications (SPIE '11)*, November 2011.

[12] S. Sánchez, G. Martín, and A. Plaza, "Parallel implementation of the N-FINDR endmember extraction algorithm on commodity graphics processing units," in *Proceedings of the 30th IEEE International Geoscience and Remote Sensing Symposium (GARSS '10)*, pp. 955–958, July 2010.

[13] M. ElMaghrbay, R. Ammar, and S. Rajasekaran, "Fast GPU algorithms for endmenber extraction from hyperspectral images," in *Proceedings of the 2012 IEEE Symposium on Computers and Communications (ISCC '12)*, pp. 631–636, Cappadocia, 2012.

[14] C.-I. Chang and Q. Du, "Estimation of number of spectrally distinct signal sources in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 3, pp. 608–619, 2004.

[15] NASA Jet Propulsion Laboratory, Free AVIRIS Standard Data Products, 2013, http://aviris.jpl.nasa.gov/data/.

[16] U.S. Geological Survey, USGS Digital Spectral Library, 2013, http://speclab.cr.usgs.gov/spectral-lib.html.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 719640, 9 pages
http://dx.doi.org/10.1155/2013/719640

Research Article

A Novel Image Mosaicking Algorithm for Wireless Multimedia Sensor Networks

Zhiyuan Li, Weiting Kong, Yongzhao Zhan, and Junlei Bi

School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013, China

Correspondence should be addressed to Zhiyuan Li; lizhiyuan81@126.com

Received 18 July 2013; Accepted 27 August 2013

Academic Editor: Fu Xiao

The wisdom tourism is an important application of internet of things industry, and many cities in China have paid attention to the development of wisdom tourism, such as the national historic city of Zhenjiang. How to take advantage of the cooperation among sensor nodes to obtain the panoramic information of scenic spots is a challenging issue for the wisdom tourism. However, the existing image mosaic algorithms are not suitable for wireless multimedia sensor networks (WMSNs) due to the resource-constrained multimedia sensor nodes, such as energy, computing, and storage space. And hence an image mosaic algorithm based on phase correlation and weighted average (IMBPW) is proposed in this paper. The IMBPW algorithm uses the phase correlation based on Fourier transform to achieve the registration of translation, rotation, and scaling images. After the image registration, the adaptive weighted average algorithm is proposed to do the image fusion. The simulation experiments show that compared with homogeneous algorithms, the IMBPW algorithm has higher real-time and fast convergence speed. Furthermore, the simulation results also show that the proposed algorithm can improve the accuracy of image registration, reduce the complexity of the image mosaic, and prolong the network lifetime while providing better image quality.

1. Introduction

In 2010, the national historic city of Zhenjiang firstly put forward the concept of wisdom tourism. The wisdom tourism makes comprehensive use of the cloud computing, the Internet of Things [1], and other wireless broadband communication technologies, and it can provide the abundant scenic spots information for tourists at anytime and anyplace. The government has already deployed lots of wireless multimedia sensors at each scenic spot of Zhenjiang city, and then the engineers use image mosaic technology to put the local images together to form the panoramic image of scenic spots. The image mosaic technology for wireless multimedia sensor networks (WMSNs) [2, 3] has played an important role in panorama images processing for the scenic spots however, most of the existing image mosaic schemes are not suitable for WMSNs. Hence, it is necessary to study this technology.

WMSNs are a network of spatially distributed smart camera sensors capable of processing and fusing images of a scene from a variety of viewpoints into some form more useful than the individual images. In WMSNs, the sensors are small and have limited resources, such as computing, transmission bandwidth, and power resources. However, multimedia applications require higher bandwidth, greater information processing capabilities, and higher energy consumption. And hence, a good image mosaic algorithm design for WMSNs has to take into consideration the tradeoff between the complexity and the accuracy of the image mosaic.

This paper proposes an image mosaic algorithm based on phase correlation and weighted average (IMBPW). The IMBPW algorithm runs on the top of a structured network. The advantage of the structured network is that fewer nodes may be deployed with lower network maintenance and management overhead. The IMBPW algorithm firstly uses the phase correlation based on Fourier transform to achieve the registration of translation, rotation, and scaling images. After the image registration, the adaptive weighted average algorithm is proposed to fulfill the task of image fusion. The IMBPW algorithm is implemented on the platform of CMUcam camera sensors. The experimental results show that the proposed algorithm can achieve the tradeoff between the image mosaic convergence speed and the quality of

Table 1: Advantages and disadvantages of image registration methods.

Types	Advantages	Disadvantages
Gray matching [4]	Simple to understand, easy to implement	Weak registration accuracy, excessive dependence on the gray value, large computation
Feature matching [3, 5–10]	Medium registration accuracy and wide scope of applications	Excessive dependence on feature extraction, and prone to mismatch
Frequency matching [11, 12]	High registration accuracy, independency of other system, and strong anti-interference	The extra overhead incurred by the coordinate transformation

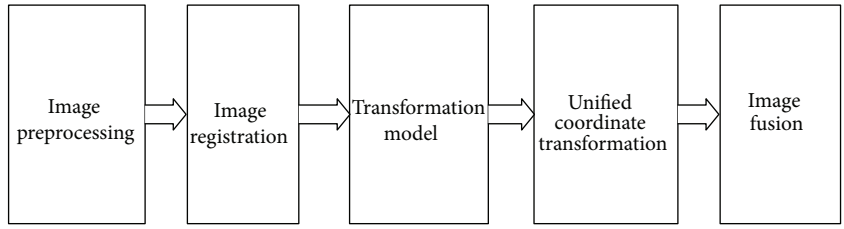


Figure 1: Image mosaic process.

image. Moreover, the IMBPW algorithm has a positive effect on controlling the energy consumption of networks and prolonging the network lifetime.

The rest of the paper is organized as follows: Section 2 introduces an overview of existing related works and motivation. Section 3 presents the image mosaic algorithm based on phase correlation and weighted average. Section 4 provides the experimental simulation and analysis. Finally, Section 5 concludes the paper.

2. Related Works and Motivation

Image mosaic is a technology which combines two or more partial images into a large seamless high-resolution image. It is usually made up of five steps, such as image preprocessing, image registration, the establishment of transformation model, unified coordinate transformation and image fusion. Among these steps, both the image registration and image fusion are two key issues. Figure 1 shows the process of image mosaic.

2.1. Image Registration. Image registration is the process of transforming different sets of data into one coordinate system, and the data sets are from different multimedia sensor nodes. Registration is necessary to integrate the data obtained from different sensors.

The image registration algorithms can be classified into three categories.

(1) Gray information-based image registration: gray information-based image registration is a kind of mathematical analysis methods, which includes the crosscorrelation method (also known as template matching), the sequential similarity detection method and the mutual information method. The gray level is referred to as the brightness of pixels. This method does not require complex pre-processing, but use the statistics of the gray information of images to measure the similarity between two images. Its main characteristic is simple; however, it has the narrow scope of applications, and it cannot be directly used to correct the nonlinear distortion of the images. Moreover, it needs a mount of computation [4].

(2) Feature-based image registration: feature-based image registration method requires image preprocessing, such as the image segmentation and the image feature extraction, and then uses the extracted features to complete the match between the two images [3]. At present, there are many image features, that results in a variety of feature-based approaches, including the edge point extraction methods, such as LOG operator, Canny operator, and wavelet transform- based algorithm, and the corner detection methods, such as SUSAN corner detection, Harris corner detection and other methods [5–10].

(3) Frequency domain-based image registration [11]: the most common image registration based on the frequency domain is the phase correlation algorithm [12]. According to the phase information of images, the phase correlation algorithm can calculate the cross power spectrum of two images and then get the impulse function by inversing Fourier transform. It is worth to note that the Cartesian coordinate needs to be transformed into the log-polar coordinate for the scaling and rotation transformation images.

The advantages and disadvantages of the three types of image registration methods are analyzed in Table 1.

2.2. Image Fusion. Multisensor image fusion is the process of combining relevant information from two or more images into a single panoramic image. The panoramic image will show more informative than any other partial images. Image

Table 2: Advantages and disadvantages of image fusion algorithms.

Types	Advantages	Disadvantages
Median filtering [13]	Simple operation, fast running speed, and preserving the edges while removing noise	The weak quality of the image fusion
Wavelet transform [14, 15]	High-quality fused image	Complex, large amounts of calculation, and lack of real time
Weighted average [16, 17]	Simple operation, fast running speed, and improving the SNR of fused images	Making the edges and contours blurry

fusion methods can be broadly classified into three classes, such as the median filtering method, the wavelet transform method and the weighted average method.

(1) Median filtering method [13]: the median filter is a nonlinear digital filtering technique, often used to remove noise. Median filtering is also widely used in digital image processing because it still can preserve edges while removing noise. The principle of the median filter is described as below. Firstly, a filter window is set. Next, the filter window is moved until all pixel points in the image are traversed. If the gray value of a pixel is very different from that of the adjacent pixels, the pixel is replaced by the average gray value of the surrounding pixels. The advantage of the median filter method is the simple numerical operation and fast running speed, and its disadvantage is the weak quality of the image fusion.

(2) Wavelet transform method [14, 15]: wavelet transform is a time-frequency localization method in which the size of its window is fixed, but its shape is variable. The continuous wavelet transform of square integral function $f(t) \in L^2(R)$ is defined as

$$\mathrm{WT}_f(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t)\, \psi^* \left(\frac{t-b}{a} \right) dt, \quad a \neq 0, \tag{1}$$

where $\psi_{a,b}(t) = (1/\sqrt{|a|})\psi^*((t-b)/a)$ is the wavelet generated by the mother wavelet, a denotes the scale parameter, and b denotes the translation parameter. This image fusion algorithm based on 2D wavelet transform firstly decomposes the image into low pass in horizontal and low pass in vertical (LL), high pass in horizontal and low pass in vertical (HL), low pass in horizontal and high pass in vertical (LH) and high pass in horizontal and high pass in vertical (HH) bands. The resolution of image becomes half of the original image. The wavelet transform is further applied in the temporal domain in the LL band only. The resolution of image becomes a quarter of the original image. The decomposition coefficients after decomposition are processed to highlight the contours of the image and weaken its details. Image fusion coefficient is worked out by adding the decomposition coefficients of each layer in wavelet transform region. Finally, we can reconstruct the fusion coefficient to get the image mosaic.

(3) Weighted average method [16, 17]: the weighted average fusion algorithm firstly calculates the weights of images and then superimposes the images through algebra and linear mathematical operation. The method is easy implemented and has fast convergence speed. Furthermore, it can efficiently fuse the images under different illuminations and improve the signal-to-noise ratio (SNR) of image fusion.

The advantages and disadvantages of the image fusion methods mentioned above are analyzed in Table 2. In conclusion, we make a decision to use the phase correlation and the weighted average methods to achieve the image mosaic for WMSNs.

3. Image Mosaic Algorithm Based on Phase Correlation and Weighted Average

3.1. Network Architecture. We divide the network into uneven clusters using our proposed protocol, called UCBCPNS [18], where each cluster is deployed with heterogeneous sensors (camera, audio, and scalar sensors) that communicate directly in a certain schedule with a cluster head and relay their sensed data and images to it. Moreover, these heterogeneous sensor nodes have the same radio interface and propagation range. A cluster head has more resources, and it is able to perform intensive and complex data processing. These powerful nodes and cluster heads are nonuniformly deployed in the network, and they are wirelessly connected with the base station either directly (in case of 1st-level cluster heads) or through other cluster heads in multihop mode. The graphical depiction of the nonuniform clustering network architecture is shown in Figure 2. Our algorithm runs on the top of the nonuniform clustering network topology.

3.2. Image Registration Based on Phase Correlation. Frequency-domain methods find the transformation parameters for registration of the images while working in the transform domain. Such methods work for simple transformations, such as translation, rotation, and scaling. Applying the phase correlation method to a pair of images produces a third image which contains a single peak. The location of this peak corresponds to the relative translation between the images. Unlike many spatial-domain algorithms, the phase correlation method is resilient to noise, occlusions, and other defects typical of medical or satellite images. Additionally, the phase correlation uses the fast Fourier transform to compute

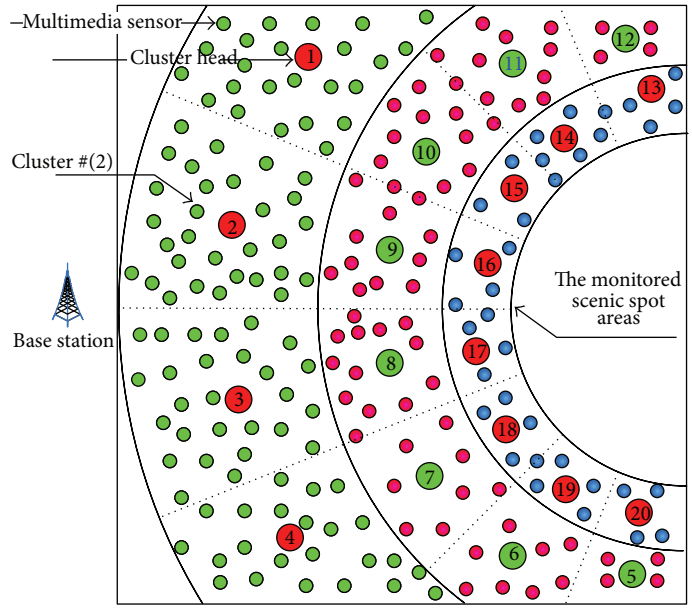


Figure 2: Network architecture.

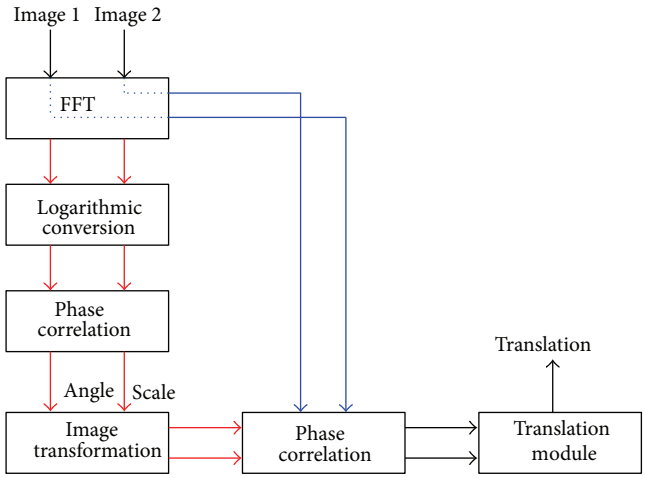


Figure 3: Overview of the registration of images.

the cross-correlation between the two images, generally resulting in large performance gains.

There are two types of registration of images. One is the registration of images only with translation transform which is shown in Section 3.2.1. The other is the registration of images with translation, rotation, and scaling transforms which is shown in Section 3.2.2. Their processing procedure is depicted in Figure 3, where the blue solid line indicates the procedure of Section 3.2.1, and the red solid line indicates the procedure of Section 3.2.2.

3.2.1. Registration of Images with Translation Transform. The phase correlation with translation transform method depends on the translation property of the Fourier transform, namely, the Fourier shift theorem, which is shown in (2). The shift theorem can guarantee that the phase of cross-power spectrum is equivalent to the phase difference between the images. Consider

$$f(x - x_0, y - y_0) \equiv F(\mu, \nu) \exp\left[-j2\pi(\mu x_0 + \nu y_0)\right]. \tag{2}$$

The images registration with translation transforms is described as below.

Let f_1 and f_2 be the two images that differ only by a displacement (x_0, y_0), which is shown as

$$f_2(x, y) = f_1(x - x_0, y - y_0). \tag{3}$$

Their corresponding Fourier transforms F_1 and F_2 will be related by

$$F_2(\mu, \nu) = e^{-j2\pi(\mu x_0 + \nu y_0)} F_1(\mu, \nu). \tag{4}$$

The correlation between the images is calculated by the inner product of Fourier spectrum instead of the convolution due to the large amount calculation of convolution. Then the cross-power spectrum of two images f_1 and f_2 with Fourier transforms F_1 and F_2 is defined as

$$\frac{F_1(\mu, \nu) F_2^*(\mu, \nu)}{\left|F_1(\mu, \nu) F_2^*(\mu, \nu)\right|} = e^{-j2\pi(\mu x_0 + \nu y_0)}, \tag{5}$$

where $F_2^*(\mu, \nu)$ is the complex conjugate of F_2. By taking inverse Fourier transform of (5) in the frequency domain, we can get a impulse function which is shown in (6). It is approximately zero everywhere except at the displacement that is needed to optimally register the two images. And hence, we can use a way of solving the location of the maximum of impulse function to determine the translation parameters (x_0, y_0) between image f_1 and image f_2. Consider

$$\delta(x - x_0, y - y_0) = F^{-1}\left[\frac{F_1(\mu, \nu) F_2^*(\mu, \nu)}{\left|F_1(\mu, \nu) F_2^*(\mu, \nu)\right|}\right]. \tag{6}$$

The specific solving steps are described as follows:

(1) input two images f_1 and f_2 which have the same dimensions, such as $M \times M$. If the images are color images, we have to turn them into 2D gray images;

(2) use the two-dimensional fast Fourier transform (FFT) on the two images to get F_1 and F_2;

(3) use (5) to compute the cross-power spectrum to get the phase difference matrix;

(4) the impulse function is obtained by taking the inverse FFT of the phase difference matrix, and then the displacement (x_0, y_0) can be solved by (6).

3.2.2. Registration of Images with Translation, Rotation, and Scaling Transforms. The phase correlation with translation transform can accurately detect the displacement between images, but it is very sensitive to rotation and scaling transform. And hence, we have to use an improved image registration algorithm to deal with the registration of images with rotation and scaling transforms. The principle of the improved image registration scheme can be described as below.

We firstly combine the log polar with the phase correlation. And then according to the distance invariance and angle invariance in log-polar transformation, the image rotation and scaling transforms are converted into the translation of amplitude spectrum in log-polar coordinate system. Let us elaborate the scheme by the formal method as below.

If $f_2(x, y)$ is a translated, scaled, and rotated replica of $f_1(x, y)$ with translation (x_0, y_0), scale factor a, and rotation θ_0, then we can get (7)

$$f_2(x, y) = f_1(x, y)\left[\frac{1}{\alpha}\begin{pmatrix}\cos\theta_0 & \sin\theta_0\\ -\sin\theta_0 & \cos\theta_0\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} - \begin{pmatrix}x_0\\ y_0\end{pmatrix}\right]. \tag{7}$$

According to the Fourier translation property and the Fourier rotation property, transforms of f_1 and f_2 in (7) are related by (8)

$$F_2(\mu, \nu) = \alpha^2 F_1\left[\alpha(\mu\nu)\begin{pmatrix}\cos\theta_0 & -\sin\theta_0\\ \sin\theta_0 & \cos\theta_0\end{pmatrix}\right] \times e^{-j\alpha(\mu,\nu)}\begin{pmatrix}\cos\theta_0 & -\sin\theta_0\\ \sin\theta_0 & \cos\theta_0\end{pmatrix}\begin{pmatrix}x_0\\ y_0\end{pmatrix}. \tag{8}$$

Let $M_1(\mu, \nu)$ and $M_2(\mu, \nu)$ be the magnitudes of $F_1(\mu, \nu)$ and $F_2(\mu, \nu)$ from (8), and then we can get

$$M_2(\mu, \nu) = \alpha^2 M_1(\mu, \nu)\begin{pmatrix}\cos\theta_0 & -\sin\theta_0\\ \sin\theta_0 & \cos\theta_0\end{pmatrix}. \tag{9}$$

After the Cartesian coordinate system is converted into the log polar coordinate system, $M_1(\mu, \nu)$ and $M_2(\mu, \nu)$ are converted into $M_1(\rho, \theta)$ and $M_2(\rho, \theta)$ simultaneously. The (9) is equivalent to

$$M_2(\lg\rho, \theta) = \alpha^2 M_1(\lg\rho + \lg\alpha, \theta + \theta_0). \tag{10}$$

Then let $\gamma = \lg\rho$ and $\gamma_0 = \lg\alpha$; then, we can get

$$M_2(\gamma, \theta) = \alpha^2 M_1(\gamma + \gamma_0, \theta + \theta_0), \tag{11}$$

where γ is the log-polar radius. Equation (11) coverts the rotation and scaling transforms in spatial domain into the translation between $M_1(\gamma, \theta)$ and $M_2(\gamma, \theta)$ in frequency domain.

The cross-power spectrum between $F_{m1}(\mu, \nu)$ and $F_{m2}(\mu, \nu)$ is calculated by

$$\frac{F_{m1}(\mu, \nu) F_{m2}^*(\mu, \nu)}{\left|F_{m1}(\mu, \nu) F_{m2}^*(\mu, \nu)\right|} = e^{-j(\mu\gamma_0 + \nu\theta_0)}, \tag{12}$$

where both $F_{m1}(\mu, \nu)$ and $F_{m2}(\mu, \nu)$ are the Fourier transforms of $M_1(\gamma, \theta)$ and $M_2(\gamma, \theta)$, respectively.

We can obtain an impulse function by taking inverse Fourier transform of (12), and then the phase correlation scheme is used to solve the scale value a and the angle value θ_0. The specific solving algorithms are described as follows.

(1) Obtain two original images f_1 and f_2 which have the same dimensions, such as $M \times M$. If the images are the color images, we have to convert them into 2D gray images.

(2) The two-dimensional fast Fourier transform (FFT) of each image is taken to get F_1 and F_2 respectively.

(3) Solve the magnitudes of F_1 and F_2, and convert the Cartesian coordinate system into the log-polar coordinate system.

(4) The phase difference matrix is derived by forming the cross-power spectrum computed by (12).

(5) Take the inverse FFT of the phase difference matrix to obtain the impulse function and calculate the scale and the angle information.

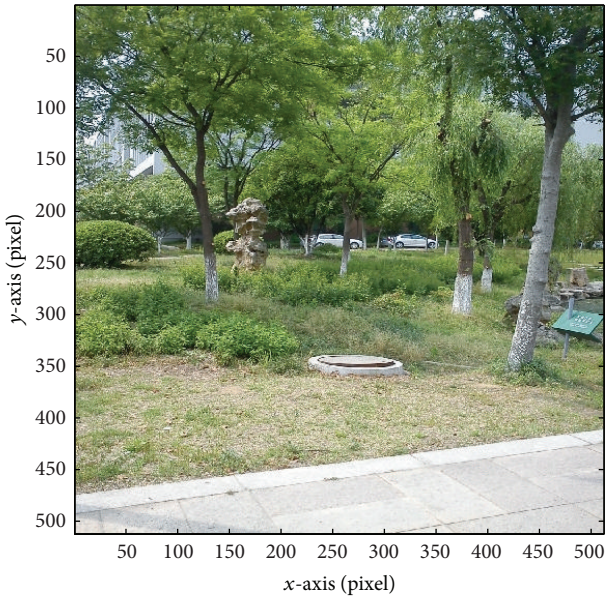
(6) Once the scale a and the angle θ_0 are obtained, the image f_2 is scaled and rotated by amounts a and θ_0, respectively. And then the mount of translational movement (x_0, y_0) is found out using the images registration algorithm in Section 3.2.1.

3.3. Image Fusion Based on Weighted Average. Assume that ω_1 and ω_2 are the weights of pixels of the overlapping area between the image f_1 and the image f_2, respectively. The image f fused by the image f_1 and the image f_2 is denoted as

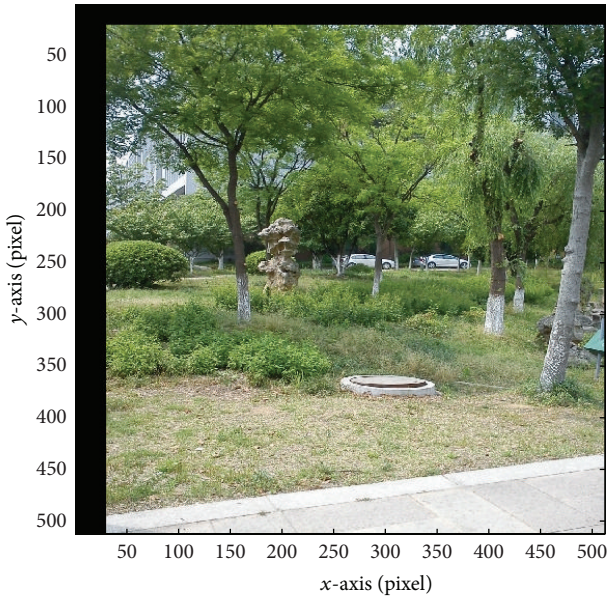
$$f(x, y) = \begin{cases} f_1(x, y) & (x, y) \in f_1 \\ \omega_1 f_1(x, y) + \omega_2 f_2(x, y), & (x, y) \in f_1 \cap f_2 \\ f_2(x, y) & (x, y) \in f_2, \end{cases} \tag{13}$$

where $\omega_1 + \omega_2 = 1$, $0 < \omega_1 < 1$, $0 < \omega_2 < 1$. Because ω_1 varies from one to zero and ω_2 varies from one to zero, the fused image has a smooth transition from the image f_1 to the image f_2 in the overlapping region. The weights of pixels ω_1 and ω_2 are denoted as (14)

$$\omega_1 = \frac{x_r - x_i}{x_r - x_l}, \qquad \omega_2 = 1 - \omega_1, \tag{14}$$



(a) The reference scenic spot image



(b) The translated scenic spot image

(c) The rotated and scaled scenic spot image

Figure 4: The reference image and examined image.

where x_i denotes the abscissa of the current pixel, x_l denotes the abscissa of the left edge of the overlapping area, and x_r is the abscissa of the right edge of the area.

4. Experimental Simulation and Analysis

4.1. Simulation Environment Settings. In this part, we simulate our proposal using CMUcam camera sensors. The network size is 400 m × 400 m deployed with 400 camera nodes for duration of 1200 time rounds. We use the first-order radio model as the energy consumption model in the paper, and assume that initial energy of a node is 0.5 J.

In the simulations, we focus on measuring the performance on different sets of the images, such as the images with white Gaussian noise and no noise. We choose (a), (b), and (c) in Figure 4 as the test images of this simulation experiments. Figure 4(a) is the reference scenic spot image. Figure 4(b) is a translated replica of Figure 4(a) with translation (30, 20), which means that the image's offset on the x-axis direction is 30 pixels and its offset on the y-axis direction is 20 pixels. Figure 4(c) is a rotated and scaled replica of Figure 4(a) with scale factor 1.2 and rotation 22°.

4.2. Simulation Results Analysis. The application of the proposed IMBPW algorithm results in a sharp peak at the point of registration. Theoretically, if the two images are the same, the peak value on the phase correlation surface should be equal to 1.0. However, both the presence of the difference

Table 3: Comparison of the running time of ABS, SIFT, and IMBPW algorithms.

Reference image	Running time/Sec		
	ABS	SIFT	Our algorithm
Translated Figure 4(a)	131.344	20.552	2.3181
Scaled Figure 4(a)	332.568	100.398	10.041
Rotated Figure 4(a)	227.973	80.820	7.704

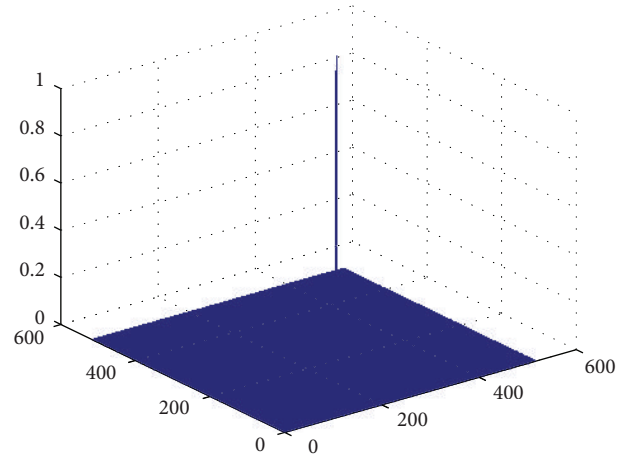


Figure 5: Phase correlation surface between Figures 4(a) and 4(b).

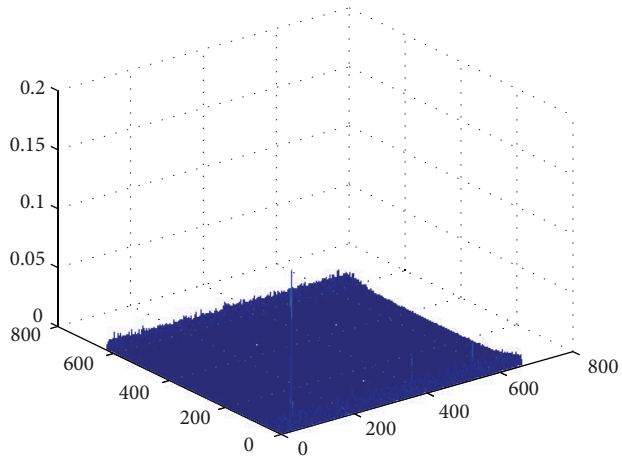


Figure 6: Phase correlation surface between Figures 4(a) and 4(c).

between the images and the images with white Gaussian noise reduce the peak value. Figure 5 shows the phase correlation between Figures 4(a) and 4(b). In the figure, the peak value is 0.92457, and its corresponding position which may be obtained by the way of inverse Fourier transform of the cross-power spectrum lies in the coordinate (30, 20). The translation parameter calculated by the IMBPW algorithm is consistent with the pre-set translation parameter.

According to the registration algorithm mentioned in Section 3.2.2, the rotation and scaling parameters can be calculated by inverse Fourier transform of the cross-power spectrum, which is shown in Figure 6. In the figure, the phase correlation surface is characterized by a sharp peak value which corresponds with the parameters of rotation and scaling in log-polar plane. We obtain the numerical result that the rotation angle is 21.622°, and magnification is 1.1977. Moreover, their corresponding computing errors are 0.378° and 0.0023 respectively. The computing errors are induced by the coordinate transformation.

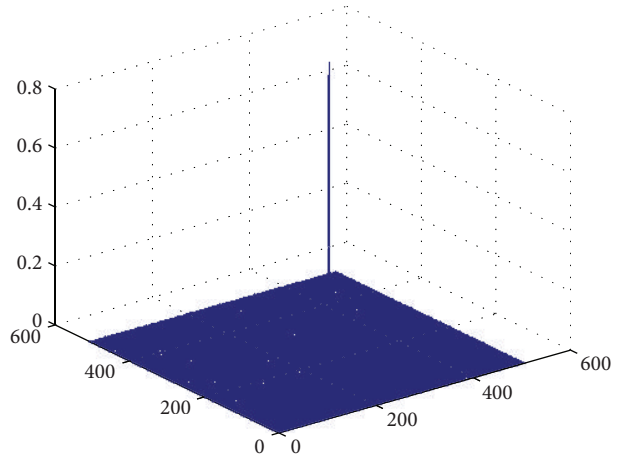


Figure 7: Phase correlation surface between the noisy Figure 4(a) and the noisy Figure 4(b).

In order to verify the proposed registration algorithm for the noisy image, the white Gaussian noise with the mean value 0.2 and the variance value 0.008 is inserted into Figure 4(a). The phase correlation surface between the noisy Figure 4(a) and the noisy Figure 4(b) is shown in Figure 7. In the figure, although the peak value decreases to 0.74842, the location corresponding with the peak value still lies in the coordinate (30, 20).

The experimental results mentioned above show that our proposed image registration algorithm has a high registration accuracy and robust.

Tables 3 and 4 are the comparison of two classical image registration algorithms with our algorithm in terms of the running speed and the registration accuracy. And the two classical registration algorithms are ABS (the abbreviation of Absolute Balance Search) and SIFT (the abbreviation of Scale Invariant Feature Transform) respectively.

As we can see from Tables 3 and 4, ABS has high registration accuracy, but its running time is too long to be suitable for WMSNs; the running time of SIFT is much lower than ABS, but its registration accuracy is the worst; our algorithm has a high registration accuracy and its running time is the best among the three algorithms. And hence, our

Table 4: Comparison of registration accuracy for translated Figure 4(a).

Reference displacement	Translation parameter solved by		
	ABS	SIFT	IMBPW
(73, 97)	(73, 97)	(72, 98)	(73, 97)
(31, 51)	(31, 51)	(31, 51)	(31, 51)
(10, 5)	(10, 5)	(10, 5)	(10, 5)

(a) Scenic spot image 1

(b) Scenic spot image 2

Figure 8: Original images to be stitched.

Figure 9: Images mosaic result using the IMBPW.

algorithm has more advantages for the resource-constrained WMSNs. It is more suitable for WMSNs environments.

There are two scenic spot images to be stitched in Figure 8.

And then we use IMBPW algorithm to achieve the image mosaic. The images mosaic result is shown in Figure 9. The panoramic image of the scenic spot is smooth and has high-quality image.

The energy consumed for transmission is a critical factor for battery-operated sensor nodes. We assume that all the sensor nodes have the same residual energy and that the chosen sensor nodes are responsible for transmitting 10 frames at each request. We evaluate the network lifetime of our proposed scheme (IMBPW), the nonoverlapping panoramic mosaic (NOPM) [3], and the scheme without image mosaic (Conventional). This simulation is repeated 1000 times to calculate the average of the network lifetime

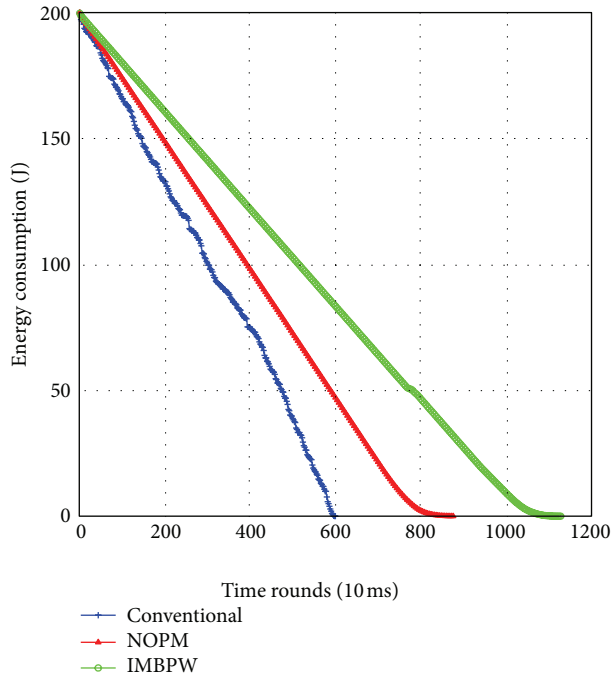


Figure 10: Network lifetime performance.

performance. The simulation results is shown in Figure 10. For the conventional scheme, the last node depletion time is at 600 rounds. For the NOPM scheme, the last node depletion time is at 872 rounds. For our proposed scheme, the last node depletion time is at 1120 rounds. Obviously, our proposed scheme significantly improves the network lifetime.

This increase is due to the method used to reduce the amount of image data transmitted.

5. Conclusions

This paper presented IMBPW, an image mosaic algorithm based on phase correlation and weighted average for WMSNs, which aims at intelligently providing the abundant scenic spots information for tourists at anytime and anyplace. The innovation of our proposed algorithm lies in the combined use of the phase correlation based on Fourier transform along with the adaptive weighted average algorithm. The proposed image mosaicking algorithm can process not only the images with noise but also the images with being not sensitive to the varying energy in the frequency domain.

In this way, the tradeoff between the better quality of image mosaic and lower computational and energy consumption overhead can be achieved.

Extended simulation tests performed showed that the proposed algorithm can improve the accuracy of image registration, reduce the complexity, and increase the network lifetime. These advantages of IMBPW enhance the belief that this scheme is indeed capable of achieving real-time and high quality image mosaicking in real applications for the wisdom tourism.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (61202474, 61201160 and 61170126), by the Natural Science Foundation of Jiangsu Province (BK20130528), and by the Senior Professional Scientific Research Foundation of Jiangsu University (12JDG049).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] I. T. Almalkawi, M. G. Zapata, J. N. Al-Karaki, and J. Morillo-Pozo, "Wireless multimedia sensor networks: current trends and future directions," *Sensors*, vol. 10, no. 7, pp. 6662–6717, 2010.

[3] B. W. Kim, J.-H. Jung, B. G. Kim, and D.-J. Park, "Construction of a non-overlapping panoramic mosaic in wireless multimedia sensor networks," *IET Image Processing*, vol. 6, no. 1, pp. 53–59, 2012.

[4] K. Wu, J. Hao, and C. Wang, "The research based on multi-view image registration," *Communications in Computer and Information Science*, vol. 227, no. 4, pp. 381–387, 2011.

[5] J. Chu and C.-M. Nie, "Multi-view point clouds registration and stitching based on SIFT feature," in *Proceedings of the 3rd International Conference on Computer Research and Development (ICCRD '11)*, pp. 274–278, Shanghai, China, March 2011.

[6] B. Fan, C. Huo, and C. Pan, "Registration of optical and SAR satellite images by exploring the spatial relationship of the improved SIFT," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 4, pp. 657–661, 2013.

[7] B. Li, W. Wang, and H. Ye, "Multi-sensor image registration based on algebraic projective invariants," *Optics Express*, vol. 21, no. 8, pp. 9824–9838, 2013.

[8] S. Cao, J. Jiang, and G. Zhang, "An edge-based scale- and affine-invariant algorithm for remote sensing image registration," *International Journal of Remote Sensing*, vol. 34, no. 7, pp. 2301–2326, 2013.

[9] A. Boukerche, Y. Huang, and J. Feng, "Progressive image mosaicking in wireless image sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobility Management and Wireless Access (MobiWac '08)*, pp. 103–110, Vancouver, Canada, 2008.

[10] R. W. Pazzi, A. Boukerche, J. Feng, and Y. Huang, "A novel image mosaicking technique for enlarging the field of view of images transmitted over wireless image sensor networks," *Mobile Networks and Applications*, vol. 15, no. 4, pp. 589–606, 2010.

[11] L. R. Jorge, V. M. Rafael, and M. S. Juan, "A fourier domain framework for variational image registration," *Journal of Mathematical Imaging and Vision*, vol. 32, no. 1, pp. 57–72, 2008.

[12] J. Zhang, C.-S. Wang, and W.-L. Liao, "An image mosaics algorithm based on improved phase correlation," in *Proceedings of the International Conference on Environmental Science and Information Application Technology (ESIAT '09)*, pp. 383–386, Wuhan, China, July 2009.

[13] S. Li and X. Kang, "Fast multi-exposure image fusion with median filter and recursive filter," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 626–632, 2012.

[14] H. Ren, Y. Lan, and Y. Zhang, "Research of multi-focus image fusion based on M-band multi-wavelet transformation," in *Proceedings of the 4th International Workshop on Advanced Computational Intelligence (IWACI '11)*, pp. 395–398, Wuhan, China, October 2011.

[15] H. Wu and Y. Xing, "Pixel-based image fusion using wavelet transform for SPOT and ETM+ image," in *Proceedings of the 1st IEEE International Conference on Progress in Informatics and Computing (PIC '10)*, vol. 2, pp. 936–940, Shanghai, China, December 2010.

[16] X. Wen, "Image fusion based on improved IHS transform with weighted average," in *Proceedings of the International Conference on Computational and Information Sciences (ICCIS '11)*, pp. 111–113, Chengdu, China, October 2011.

[17] C. He, Y. Qin, and G. Cao, "Medical image fusion using guided filtering and pixel screening based weight averaging scheme," *Journal of Software Engineering*, vol. 7, no. 2, pp. 77–85, 2013.

[18] Z.-Y. Li and R.-C. Wang, "Secure coverage-preserving node scheduling scheme using energy prediction for wireless sensor networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 17, no. 5, pp. 100–108, 2010.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 464703, 7 pages
http://dx.doi.org/10.1155/2013/464703

Research Article

Hybrid Ant Algorithm Based Query Processing with Multiagents in Sensor Networks

Jianping Yu,[1,2] Lianming Zhang,[3] Ming Chen,[1] and Xingting Liu[1]

[1] *College of Mathematics and Computer Science, Key Laboratory of High Performance Computing and Stochastic Information Processing, Ministry of Education of China, Hunan Normal University, Changsha 410081, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China*
[3] *College of Physics and Information Science, Hunan Normal University, Changsha 410081, China*

Correspondence should be addressed to Jianping Yu; jianpinghn@163.com and Lianming Zhang; lianmingzhang@gmail.com

Received 19 July 2013; Accepted 3 September 2013

Academic Editor: Fu Xiao

The wireless sensor networks are usually deployed in various application-specific contexts, which can be treated as distributed databases with big data. The event-involved query responses can be obtained by issuing query requests to this kind of database. However, the constraints of the energy and delay have had a great impact on the operation of wireless sensor networks. How to design the query-involved network model and the corresponding query processing algorithms is extremely challenging. This work investigates query processing problem in resource-constrained wireless sensor networks with the two-tier architecture and multiple query agents, where the multiple nodes of query agents are configured in the networks and the corresponding source cluster-heads send collected events to only one optimum query agent. To reduce the energy consumption and shorten the delivery delay, an efficient query processing algorithm inspired by the swarm intelligence of ants is proposed, which takes advantage of the beneficial clustering and routing emerging in a hybrid self-organized way from the positive interaction of ants. The experimental results demonstrated that the proposed algorithm can deliver collected events to the optimum query agents efficiently. Not only is the energy cost reduced but also the delivery delay is shortened significantly when transmitting the named events to the appropriate query agents.

1. Introduction

The technologies of wireless sensor networks (WSN) have been developed rapidly in the most recent years. Because WSN can cover large areas and extract localized features, the applications involved are from a wide variety of areas such as environment monitoring and security [1–3].

As WSNs are characterized by data-centric routing under most circumstances, any application involved requires data processing technologies, especially the query processing [4]. Because of the severe constraints on energy and computation that are characteristics of WSNs, the centralized querying approach is often infeasible for large-scale networks, and it is crucial to propose energy-efficient distributed in-network querying protocols while providing an acceptable quality of information [5]. A decentralized infrastructure for supporting querying in WSNs was introduced in [6], which utilizes sensor's spatial and semantic characteristics. The ACQUIRE mechanism in [7] provides superior query optimization for responding to particular kinds of queries. However, a one-size-fits-all approach is unlikely to provide efficient solutions for all types of queries [7]. In this paper, we focus on the solution for query processing with both the energy and the delay constraints in WSNs. To the best of our knowledge, when considering the multiple QoS constraints such as the query delay and the minimal energy required, the multiconstrained querying problem is NP-hard.

For a special application in WSNs, the event types are limited. When the events involved occur somewhere, the sensors nearby will collect the information and transmit it to the local cluster head. Each of the cluster-heads can identify the limited kinds of the events and transmit the event data to the queriers. Therefore, the process of issuing queries can

be omitted, and the routing problem of how can the cluster-head find the optimum querier and deliver the events to it becomes the focused topic in this paper. To promote the query performances, the query model with multiagents is presented in this paper, in which certain number of nodes are selected as the query agents and once the named events are delivered to one of the agents, the query routing is then completed. As the query agents is selected randomly, the location of them is unknown beforehand; how to search the optimum query agent when collecting the interested events is a challenge.

A distributed multiple ant colonies any cast algorithm was proposed in [8]. Though aiming at finding replicated service, it is revelatory to search replicated query agents in WSNs.

In this paper, we propose a distributed hybrid ant algorithm based query processing algorithm with multiple agents (HAAQP), which intends to overcome the inefficiency of those previous querying algorithms by reducing the expected search cost and shortening the query delay.

The remainder of the paper is organized as follows. Section 2 describes the preliminaries and the problem formulation. The novel HAAQP algorithm is proposed in Section 3. In Section 4, the proposed algorithm is evaluated by simulations. Finally, Section 5 concludes the paper.

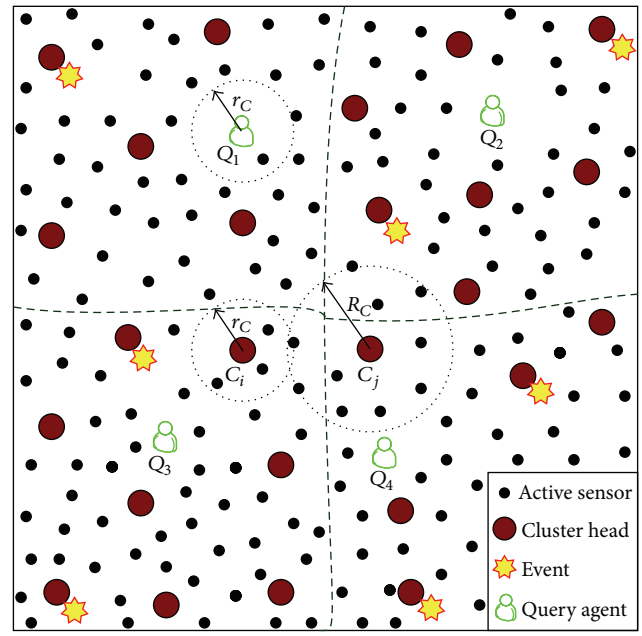


Figure 1: The network environment of the proposed HAAQP algorithm.

2. Preliminaries and Problem Formulation

2.1. Ant-Based Routing Model. Swarm intelligence is a relatively new approach to problem solving that takes inspiration from the social behaviors of insects. In particular, ants have inspired lots of methods and techniques among which the most successful is ant colony optimization (ACO). The large majority of the applications of ACO are to NP-hard problem. ACO was applied to routing in various networks [9, 10]. Several improved ant-based routing algorithms for WSNs were proposed in [11].

2.2. Ant-Based Clustering Model. A special kind of ants divides their eggs based on the size. An ant-based clustering algorithm was studied [12] inspired by such behavior. Ant-based clustering is characterized by a probabilistic approach, where clustering is repeatedly realized by ants, and stochastically selected eggs are picked up or dropped. By the combination of the two ant-based models, we propose a hybrid ant-based HAAQP algorithm to solve the multiconstrained querying problem for WSNs.

2.3. Network Model of WSN. As the WSNs considered in this paper are large-scale networks, the utilization of cluster can improve the scalability. A clustering protocol capable of providing uniformly distribution of cluster-heads was proposed in [13]. We adopt the similar approach for low-tier of the WSN. The cluster radius r_C and the number of cluster-heads N_C can be calculated according to [14].

To reduce the energy cost, only part of the sensors in each cluster stays alive. There are N_Q query agents in the network ($N_Q \leq N_C$). When completing clustering in the low-tier of the WSN, the up-tier of the network emerges composed of the cluster-heads including the query agents. Figure 1 shows the network environment of the problem.

In Figure 1, each cluster-head maintains several active sensors, and the sleeping sensors are omitted. The query agents Q_i are the special cluster-heads merely receiving the event packets, which are the destinations for all events. Other cluster-heads C_i act as both aggregation nodes and routers. Once an event occurrs, the sensors collect the event and transmit it to the corresponding cluster-head. If the cluster-head is a query agent, the event is received successfully. Otherwise, it will be transmitted to another cluster-head till it reaches one of the query agents or it is dropped because of the constraints on it. To enhance the scalability, it is necessary to cluster on the up-tier of WSN, which makes each cluster-head C_i belong to one optimum query agent Q_i. An ant-based clustering method is adopted to work on the up-tier of the network.

2.4. Generation and Aggregation of Events. For specific application in WSNs, suppose that the maximum number of event types and the event attributes are M_0 and k, respectively. The event attributes are denoted as $(A_1, \ldots, A_i, \ldots, A_k)$ and $1 < i < k$. Each type of events has its special ranges of attributes. For example, the range of A_i is denoted as $|A_i| \in [a_i, b_i)$. The types of events are loaded into each sensor in advance.

The events are uniformly generated during the deployment period in WSN. To reduce the storage of sensors efficiently, the aggregation method is adopted according to the spatiotemporal attributes and sensing attributes of events [14].

When an event involved occurrs, any sensor obtaining all the attribute values can infer the event type by matching the event list inserted into the node in advance.

2.5. Problem Formulation. As the WSN is treated as a two-tier network, constrained query routing among the cluster-heads on the up-tier has become the focus to be considered. There are two constraints when delivering the events to the corresponding query agent. One is energy constraint *ENE_C* of nodes and the other is query delay constraint *DUR_C* of events. Their values are different according to the different event types. When an intermediate cluster-head C_i receives an event E_r with the $ENE\_C(E_r)$ and $DUR\_C(E_r)$, the event will be forwarded only if the current query delay of the event is smaller than $DUR\_C(E_r)$ and there exist a neighbor of C_i whose energy is more than $ENE\_C(E_r)$. The up-tier of the WSN can be viewed as a connected graph $G' = (V', E')$, where $V' = \sum C_i \cup \sum Q_j$ and E' is the set of the links between cluster-heads. How to transmit the events with different constraints from local C_i to one of the query agents Q_j energy efficiently and quickly is the focused problem. To solve the NP-hard problem, an ant-based HAAQP protocol is proposed as follows.

3. Hybrid Ant-Based Query Processing for WSNs

3.1. Management of Ants. Three kinds of ants are to be considered in HAAQP, namely, the forward ants (FAs), the backward ants (BAs) and the clustering ants (CAs). The FAs can be viewed as events to be transmitted. Once an event E_r occurrs, the local cluster-head C_j completes aggregation and generates a $FA_{C_S}(ID_F, E_r, T_F, ID_{F.Cur}, VL_F)$ where the $E_r = (DT_E, T_E, ENE\_C(E_r), DUR\_C(E_r))$ and C_S is the source cluster-head. ID_F is the identifier of the FA, and T_F is the beginning time of the FA. $ID_{F.Cur}$ denotes the current sensor node which the ant arrives at. The VL_F denotes the cluster-head list ant visited. The DT_E denotes the event data, and the T_E denotes the time E_r generated. The $ENE\_C(E_r)$ and $DUR\_C(E_r)$ denote the energy constraint and delay constraint of the E_r respectively. The generation rate of the initial FAs in one cluster-head C_j is determined mainly by the event rate. We consider that the values of the two constraints are uniformly distributed. Each ant nest corresponds to a source cluster-head, and each ant in an ant colony corresponds to an event packet. Each FA will generate a new FA when it makes a move forward. Each current FA assigns its next hop as the $ID_{F.Cur}$ of the new FA then puts the $ID_{F.Cur}$ into the VL_F of the new FA. Assume that the delay of the current C_i to its next hop C_j is delay(C_i, C_j); current ant will add delay(C_i, C_j) to its T_F and assign it to the T_F of the new FA.

Both the FAs and the BAs can update the pheromone. There are two kinds of pheromones in HAAQP, namely, the search pheromone τ and the clustering pheromone π. The BAs update both of the τ and π when they come back. Each cluster-head maintains a buffer of ants including the FAs and BAs. The ants that arrive to it are arrayed in the buffer ordered by the T_F. There are special ants CAs in each cluster-head C_i, which determine the membership of C_i according to the clustering pheromone to different query agents. The CAs issue from the C_i only communicate with the neighbors of the C_i.

3.2. Initiation of HAAQP. Suppose that the energy cost for querying is to be proportional to the number of transmissions. The query delay is the end-to-end delay between the source cluster-head and the optimum query agent. The initial values of the τ and π should be initiated beforehand. In the initial phase, each query agent Q_j assigns the initial values of $\tau_{C_i}(Q_j) = \tau_{\max}$ and $\pi_{C_i}(Q_j) = \pi_{\max}$ on the directly connected links, where the C_i is the neighbor of Q_j, $1 < i \le N_C$ and $1 < j \le N_Q$. Meanwhile, the C_i declares that it is a member of Q_j. The initial values of pheromone on the other links are set as $\tau_{C_j}(C_i, Q_{C_i}) = \tau_{\min}$ and $\pi_{C_j}(Q_{C_i}) = 0$, where the C_j is the neighbor of C_i belonging to Q_{C_j} and $\tau_{\min}$ is a small positive value.

3.3. Path Construction Policy of HAAQP. When dealing with the initial or intermediate FAs, how to select the next hop of the cluster-head is important. Because the global information of the large-scale sensor network is unknown, only the local information of the neighbors can be used during the selection. As the ordinary sensors have been ignored on the up-tier of the network, the neighbors are sure to be the cluster-heads or query agents, which can improve the computational efficiency because the number of candidates to be selected is shrunk sharply. We first calculate the weight of the current C_i to all its possible neighbors by the search pheromone, residual energy of the neighbor, and link cost then select the next hop C_j according to the following equation:

$$j = \begin{cases} \arg\max \left\{ \tau_{C_i}\left(C_j, Q_{C_j}\right) + \alpha \cdot \lambda\left(C_i, C_j\right) \right. \\ \qquad \left. + \beta \cdot \mu\left(C_i, C_j\right) \right\} & \text{if } q \le q_0, \\ J & \text{else,} \end{cases} \tag{1}$$

where $\tau_{C_i}(C_j, Q_{C_j})$ is the search pheromone of the link (C_i, C_j), $\lambda(C_i, C_j) = 1/c(C_i, C_j)$, and $c(C_i, C_j)$ is the cost of link (C_i, C_j); $\mu(C_i, C_j)$ is the residual energy of the C_j; α and β are parameters controlling the relative importance of the link cost and residual energy, respectively. The parameter $q_0 = 0.8$ and $q \in (0, 1]$. The value of J is a stochastic variable, which is determined by the probability of $P_k(C_i, C_j)$. The $P_k(C_i, C_j)$ for the kth FA in C_i to the neighbor C_j is defined as follows:

$$P_k\left(C_i, C_j\right) = \frac{\tau_{C_i}\left(C_j, Q_{C_j}\right) + \alpha \cdot \lambda\left(C_i, C_j\right) + \beta \cdot \mu\left(C_i, C_j\right)}{\sum_{d \in L^k(C_i)} \left[\tau_{C_i}\left(d, Q_d\right) + \alpha \cdot \lambda\left(C_i, d\right) + \beta \cdot \mu\left(C_i, d\right)\right]}, \tag{2}$$

where $L^k(C_i)$ is the set of the optional neighbors to the cluster-head C_i. From (1) and (2), we can see that FAs tend to select the links with smaller cost, more available energy of neighbor, and stronger pheromone. If an ant FA finds that the neighbor

is in its path, it discards the neighbor to prevent loop. To judge whether the path satisfies the constraints, the energy level of the neighbors needs to be recorded timely, as well as the T_F of the ants. If $\mu(C_i, C_j) < ENE\_C(E_r)$ or $(T_F - T_E) > DUR\_C(E_r)$, apparently the event packet is unable to reach the query agent via C_j. If none of the neighbors satisfies the constraints, the ant dies. The illustration is shown in Figure 2.

3.4. Pheromone Update Rules of HAAQP. Two updating rules for search pheromone are introduced herein: the global updating and the local updating. Ants will intensify the pheromone of the link when moving forward each step, so that the ants sent from the same nest can tend to select the same path. Because each ant has different constraints according to different events, they perhaps select another path. Suppose that FA at C_i select C_j as its next hop, the formula of updating local pheromone is as follows.

$$\tau_{C_i}\left(C_j, Q_{C_j}\right) \longleftarrow (1-\rho_1)\cdot\tau_{C_i}\left(C_j, Q_{C_j}\right) + \Phi\cdot\Delta\tau^k_{C_i}\left(C_j, Q_{C_j}\right). \tag{3}$$

Note that $\tau_{C_i}(C_j, Q_{C_j})$ is the search pheromone of the link (C_i, C_j) to the query agent Q_{C_j}; ρ_1 is the factor of the pheromone volatilization ($0 < \rho_1 < 1$). The Φ is an adjustable parameter. We consider that $\Delta\tau^k_{C_i}(C_j, Q_{C_j}) = 1/\text{delay}(C_i, C_j)$, where the $\text{delay}(C_i, C_j)$ is the delay of link (C_i, C_j). The pheromone can be adjusted dynamically based on the delay. Accordingly, the subsequent ants will not select the link with larger delay.

When an ant finds an eligible path, it intensifies the pheromone of the path and evaporates the pheromone of the involved links that are not in the path meanwhile. The global pheromone updating is defined as follows:

$$\tau_{C_i}\left(C_j, Q_{C_j}\right) \longleftarrow \begin{cases} (1-\rho_2)\cdot\tau_{C_i}\left(C_j, Q_{C_j}\right) \\ \quad +\Delta\widehat{\tau}_{C_i}\left(C_j, Q_{C_j}\right), & \text{if } \left(C_i, C_j\right) \in p, \\ (1-\rho_2)\cdot\tau_{C_i}\left(C_j, Q_{C_j}\right) & \text{else,} \end{cases} \tag{4}$$

where p is the path from source cluster-head C_S to one of the query agents. We consider that $\Delta\widehat{\tau}_{C_i}(C_j, Q_{C_j}) = (\mu_{\min})^{\sigma}/\theta$, where $\mu_{\min}$ is the bottle-neck energy level on the path p, θ is an adjustable parameter, σ is the response degree of pheromone to energy, and ρ_2 is the pheromone volatilization factor ($0 < \rho_2 < 1$) and it also simulates the volatilization of search pheromone.

In order to determine the membership of each cluster-head C_i dynamically in terms of the clustering pheromone $\pi_{C_i}(Q_j)$, the clustering ants CAs should be launched from each C_i to their neighbors in fixed interval ΔT_{CA} to update the values of the $\pi_{C_i}(Q_j)$, which reflects the attractiveness of the query agent Q_j. The rule of updating the $\pi_{C_i}(Q_j)$ is as follows:

$$\pi_{C_i}\left(Q_j\right) = \frac{\sum_{k\in\nabla_{C_i}(Q_j)}\pi_k\left(Q_j\right) + \overline{\tau}_{C_i}\left(Q_j\right)}{\left|\nabla_{C_i}\left(Q_j\right)\right| + 1}, \tag{5}$$

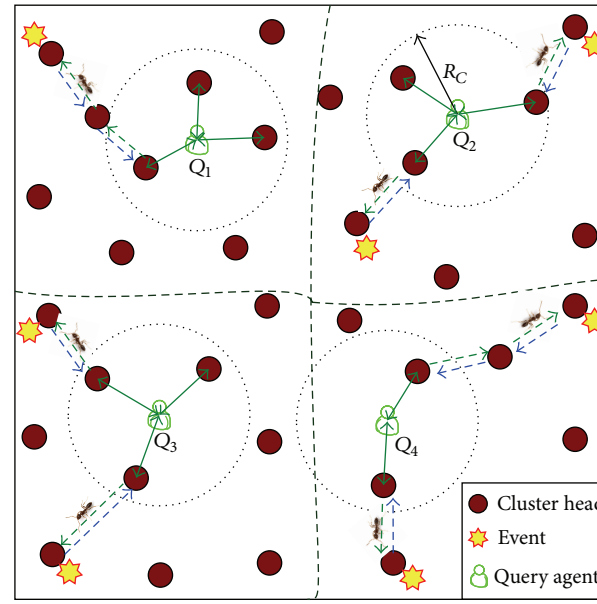


Figure 2: The path construction process of the proposed HAAQP algorithm.

where the $\nabla_{C_i}(Q_j)$ denotes the set of the cluster-heads which are neighbors of C_i that belong to the members of the query agent Q_j and the $|\nabla_{C_i}(Q_j)|$ denotes the size of the $\nabla_{C_i}(Q_j)$; $\overline{\tau}_{C_i}(Q_j)$ is the average value of the search pheromone in the outgoing links of the C_i, which is represented as follows:

$$\overline{\tau}_{C_i}\left(Q_j\right) = \frac{\sum_{k\in\nabla_{C_i}(Q_j)}\tau_{C_i}\left(k, Q_j\right)}{\left|\nabla_{C_i}\left(Q_j\right)\right|}. \tag{6}$$

From (5), and (6), we can see that the search pheromone affects the clustering pheromone to some degree. If an optimum path to the query agent Q_j is explored, the $\pi_{C_x}(Q_j)$ of the cluster-heads C_x nearby the path will increase with the $\overline{\tau}_{C_i}(Q_j)$. The events that occurred near the optimum path can be attracted to the path efficiently. When there are no explored optimum paths near the C_x of the events, FAs are launched to search the optimum neighbor till an optimum path or a query agent is found. If more optimum paths are explored, the search efficiency of FAs to one of the query agents can be improved significantly.

3.5. Approaches of HAAQP Algorithm. The HAAQP algorithm for the WSNs is described as follows.

Step 1. Initialize the values of the search pheromone on each link on the up-tier when completing the two-tier structure of the WSN by existing clustering protocol. The values of the clustering pheromone on each cluster-head C_i are also initialized beforehand.

Step 2. Each source cluster-head generates events with the query constraints of the *ENE_C* and *DUR_C* according to Poisson distribution. These events are buffered in the queue of ant-manager ordered by the generating time T_F of the FAs.

Step 3. In each buffer of the cluster-heads, the ant-manager deals with the intermediate FAs in order. When dealing with initial FAs, new intermediate FAs are generated and inserted into the buffer queue according to the T_F of the new FAs.

Step 4. Ant-manager goes on dealing with subsequent ants. They may be initial FAs, new intermediate FAs, or BAs, which is up to the T_F or T_B of the ants.

Step 5. Before dealing with ants of current cluster-head, whether the neighbors satisfying the constraints of the packets exist or not must be judged. If there is no such neighbor node, the ant dies; otherwise, we judge the type of ants and deal with them, respectively.

(1) For the initial or intermediate FAs, the ants first determine whether the current cluster-head is a query agent or a neighbor of query agents. If yes, the events are accepted directly or through one hop. Meanwhile, the BAs are triggered to update the search pheromone on the corresponding paths. Otherwise, the ants select next hop according to the probability of each possible neighbor sensor and prevent loop. Ants update the local information on the selected link. When completing the FAs, the new FAs are generated. Judge whether the new FAs are satisfied with the constraints. If no, they will be discarded; otherwise, they are inserted into the queue according to their beginning time T_F.

(2) For the BAs, they are adopted to output the paths by which events are delivered to the query agents and update the energy on the path and search pheromone on all the links.

(3) For the CAs, they are launched to collect the value of the neighbors' clustering pheromone in a fixed interval and determine the membership of the current cluster-head.

Step 6. If there are no ants to be processed, we count the total energy cost and the query delay in the simulation.

4. Simulations

We have simulated the HAAQP and evaluated the performance by comparing with the other query algorithms derived from [15–17], namely, the SARP-based query processing (SARPQP), the TBA-based query processing (TBAQP), and the ABS-based query processing (ABSQP). A topology generator is adopted to obtain various sensor networks. The number of sensor nodes N varies from 1000 to 6000, which is deployed in a square area with $\|A\| = 500\,\text{m} \times 500\,\text{m}$ and clustered into N_C clusters with N_Q query agents uniformly ($2 \le N_Q \le 12$). The simulation time $T = 3000$ units divided into T/w segments, where w is the event interval and can vary between 50 and 300 time units. The small the value w is, the heavier the network load is and vice versa. The values of the T and w determined the number of events together. There

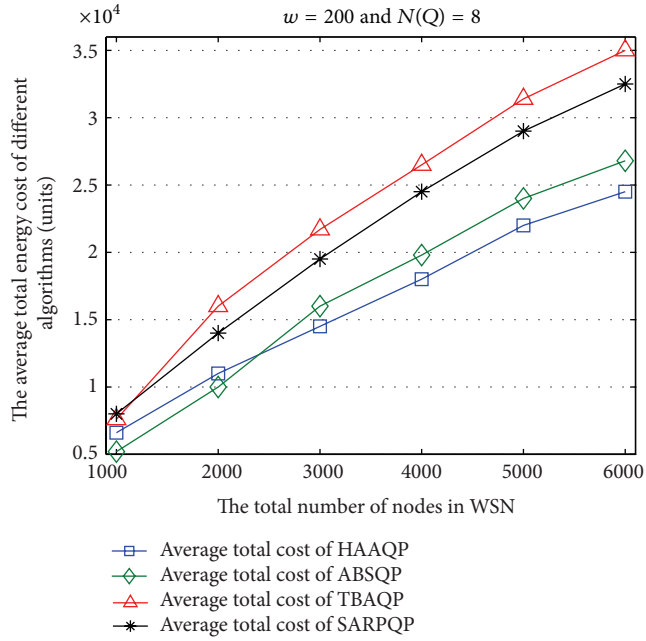


FIGURE 3: The average total energy cost of HAAQP and other algorithms.

are about 5% of the cluster-heads as sources. About 10 events issue from each source in each segment according to Poisson distribution.

The initial energy value of each sensor is 400 units. The values of *ENE_C* are uniformly distributed between 20 and 80 units and that of the *DUR_C* are between 60 and 100 time units. The link cost between cluster-heads is set $c(C_i, C_j) \in [2, 5]$, and the initial link delay is assigned between 2 and 8 time units. The parameter values of the $\alpha = 100$, $\beta = 0.5$, $\rho_1 = \rho_2 = 0.1$, $\sigma = 1$, $\theta = 5$, $\tau_{\min} = 5$, and $\tau_{\max} = \pi_{\max} = 100$. The metrics we investigated are the energy cost and the query delay. For each different configuration of the sensor networks, the algorithm is carried out 200 times, and the average values of the metrics are obtained. The simulation results are shown as in Figure 3 to Figure 5.

Figure 3 shows the comparison of average total energy cost. It indicates that the ant-based HAAQP is more energy efficient than the other algorithms in most cases. The reason is that the ant-based search can provide more intelligence, especially for large-scale wireless sensor networks. The ABSQP algorithm is more energy efficient than both SARPQP and TBAQP for its heuristic design.

Figure 4 plots the curve of the energy cost of HAAQP with the values of the w and N_Q. From Figure 4, we can see that the energy cost is reduced with the increase of w and N_Q. This is because the paths can be shorter with bigger value of N_Q, and the number of events is reduced with the increase of w.

Figure 5 shows the comparison of the average query delay. It indicates that the delay of HAAQP is smaller than that of other algorithms. This is because that the ant-based querying with positive feedback can find short enough paths under multiconstraints.

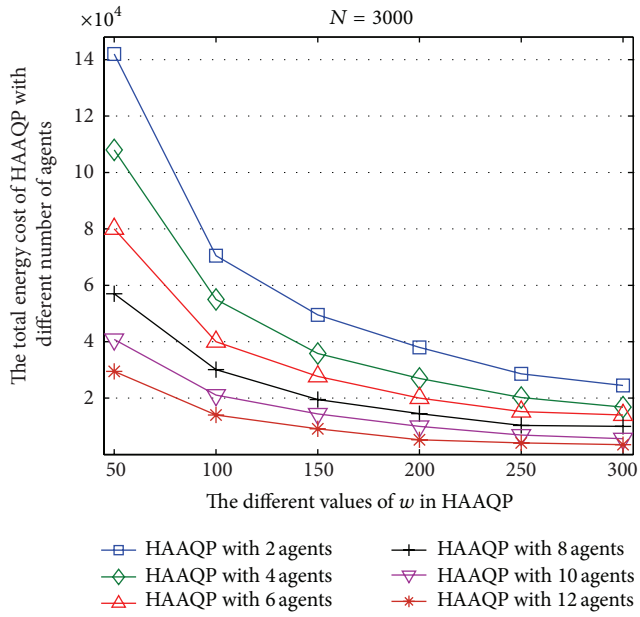


FIGURE 4: The total energy cost of HAAQP with a different number of agents.

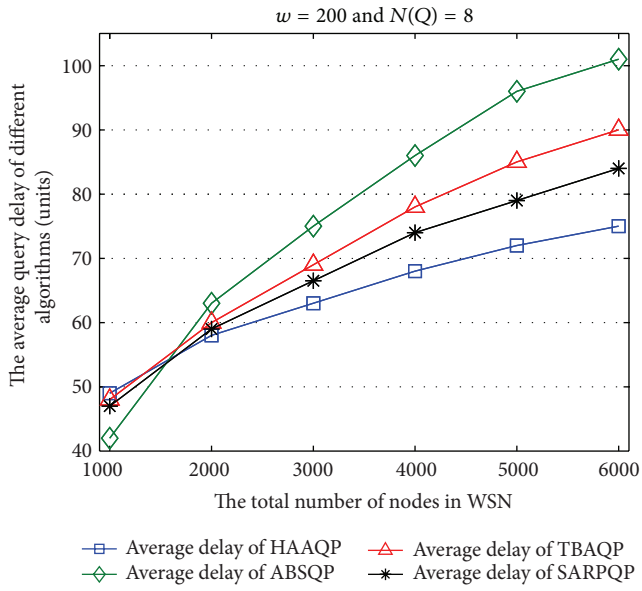


FIGURE 5: The average query delay of HAAQP and other algorithms.

5. Conclusions

The multi-constrained query processing is one of the most challenging problems for resource-constrained WSNs with multiple query agents. Considering the constraints of the node energy and query delay of different applications, an ant-based query processing algorithm HAAQP with positive feedback is proposed for WSNs in this paper. The simulation results show that significant energy gains are achieved by HAAQP, as well as the query delay. Our future work is to optimize the parameters to achieve better performance.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (Grant nos. 60903168, 60973129, 61379115, and 61100215), the Hunan Provincial Natural Science Foundation of China (Grant no. 12JJ6063), the Scientific Research Fund of Hunan Provincial Education Department of China (Grant no. 10B062), the Program for Excellent Talents in Hunan Normal University (Grant no. ET51102), the Foundation of Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks (Grant no. WSNLK201102), and the Foundation for Young Teachers in College of Mathematics and Computer Science of Hunan Normal University. The authors would like to thank the editor and the reviewers.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] J. Jia, X. Wu, J. Chen, and X. Wang, "Exploiting sensor redistribution for eliminating the energy hole problem in mobile sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, article 68, 2012.

[3] X. W. Zhang, H. P. Dai, L. J. Xu, and G. H. Chen, "Mobility-Assisted data gathering strategies in WSNs," *Journal of Software*, vol. 24, no. 2, pp. 198–214, 2013.

[4] J.-Z. Li, J.-B. Li, and S.-F. Shi, "Concepts, issues and advance of sensor networks and data management of sensor networks," *Journal of Software*, vol. 14, no. 10, pp. 1717–1727, 2003.

[5] Y. Yao and J. Gehrke, "Query processing for sensor networks," in *Proceedings of the 1st Biennial Conference in Innovative Data Systems Research*, pp. 21–32, 2003.

[6] R. Rosemark and W.-C. Lee, "Decentralizing query processing in sensor networks," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05)*, pp. 270–280, July 2005.

[7] N. Sadagopan, B. Krishnamachari, and A. Helmy, "Active query forwarding in sensor networks," *Ad Hoc Networks Journal*, vol. 3, no. 1, pp. 91–113, 2005.

[8] J. P. Yu, Y. P. Lin, M. Lin, and Y. Q. Yi, "Multi-constrained anycast routing based on Ant algorithm," *Chinese Journal of Electronics*, vol. 15, no. 1, pp. 133–137, 2006.

[9] R. Schoonderwoerd, O. E. Holland, J. L. Bruten, and L. J. M. Rothkrantz, "Ant-based load balancing in telecommunications networks," *Adaptive Behavior*, vol. 5, no. 2, pp. 169–207, 1996.

[10] G. di Caro, F. Ducatelle, and L. M. Gambardella, "AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 443–455, 2005.

[11] Y. Zhang, L. D. Kuhn, and M. P. Fromherz, "Improvements on ant routing for sensor networks," in *Proceedings of the 4th International Workshop on Ant Colony Optimization and Swarm Intelligence*, vol. 3172 of *Lecture Notes in Computer Science*, pp. 154–165, 2004.

[12] A. L. Vizine, L. N. de Castro, E. R. Hruschka, and R. R. Gudwin, "Towards improving clustering ants: an adaptive ant clustering algorithm," *Informatica Journal*, vol. 29, no. 2, pp. 143–154, 2005.

[13] M. Liu, J.-N. Cao, G.-H. Chen, L.-J. Chen, X.-M. Wang, and H.-G. Gong, "EADEEG: an energy-aware data gathering protocol

for wireless sensor networks," *Journal of Software*, vol. 18, no. 5, pp. 1092–1109, 2007.

[14] J.-P. Yu and Y.-P. Lin, "Real-time query processing for sensor networks based on ant algorithm," *Journal of Software*, vol. 21, no. 3, pp. 473–489, 2010.

[15] C. Intanagonwiwat and D. D. Lucia, "The sink-based anycast routing protocol for ad hoc wireless sensor networks," Tech. Rep. 99-698, Computer Science Department, University of Southern California, Los Angeles, Calif, USA, 1999.

[16] W. Hu, N. Bulusu, and S. Jha, "A communication paradigm for hybrid sensor/actuator networks," in *Proceedings of the IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '04)*, pp. 201–205, September 2004.

[17] Y. T. Hou, Y. Shi, and H. D. Sherali, "Optimal base station selection for anycast routing in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 3, pp. 813–821, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 209318, 10 pages
http://dx.doi.org/10.1155/2013/209318

Research Article

A Coding and Postprocessing Framework of Multiview Distributed Video for Wireless Video Sensor Networks

Chong Han,[1] Lijuan Sun,[1,2,3] and Jian Guo[1,2,3]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China*
[3] *Key Lab of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education Jiangsu Province, Nanjing 210003, China*

Correspondence should be addressed to Lijuan Sun; sunlj@njupt.edu.cn

Received 14 July 2013; Accepted 22 August 2013

Academic Editor: Zhijie Han

In many surveillance application scenarios of wireless video sensor networks (WVSNs), a number of video sensors are deployed, and multidimension monitored the visual information in a region of interest, forming multiview videos. Since the power, computing capability, and bandwidth are very limited in WVSNs, the conventional multiview video coding method is no longer applicable. So multiview distributed video coding (MDVC) emerged and developed rapidly. In this paper, we propose a new multiview video coding and postprocessing framework for multiview videos. First, in coding scheme, motion intense regions (MIRs) and nonmotion intense regions (NMIRs) based on sum of absolute difference (SAD) criteria are distinguished. For the MIR, the side information (SI) is generated by fusion temporal SI and interview spatial SI at the pixel level. But for the NMIR, the temporal SI is directly use as the ultimate SI. Then, to further improve the quality of the decoded image, an image postprocessing scheme is designed by using deblocking and deringing artifact filters on decoded image. Finally, a set of experimental results show that the proposed fusion SI approach can bring improvements up to 0.2–0.5 dB when compares with only temporal SI used. The subsequent decoded videos postprocessing simulation proves that the proposed postprocessing scheme can provide an additional improvement of about 0.1 dB to the decoded video sequences.

1. Introduction

Nowadays, with the rapid development of wireless communication and microelectronics technologies, a series of devices and systems relevant with wireless network and video techniques, such as wireless video sensor networks (WVSNs), wireless IP camera, mobile video phone, dense camera arrays, satellite communication systems, and television systems for multiview virtual meeting, have wide application prospects. These kind of video equipment or systems have a common feature, that is, the computing, data storage, and power consumption capacity are all limited, but all abilities in receiving ends are very powerful. Meanwhile, along with the demand for high-quality multimedia/video surveillance, environmental monitoring, and industrial process control, and more and more video sensors are used in WVSNs. In WVSNs, a number of video sensor nodes are deployed and multidimension shot visual information in a region of interest. As a result, the visual information retrieved from adjacent video senor nodes is called multiview video, which usually exhibit high levels of correlation and give rise to considerable data redundancy in the network.

In fact, multiview video is a kind of video, which has interactive manipulation functions and stereoscopic impression. It is the video record acquired from video sensors placed under a particular array in a same scene and provides the capability of scene roaming and viewpoint selection for users. Since all video sensors shoot the same scene from multiple views, the video network contains a large number of interview statistical dependencies, high-quality images/videos could collect by combining intraview (motion estimation), and interview prediction (disparity estimation). Although the multiview video technology provides a richer viewing experience than traditional video technology, it

needs to deal with a huge amount of data, and this brings new challenges to the data compression. Traditional video coding standards, such as MPEG-x and H.26x [1], mainly rely on the hybrid architecture, and encoder uses motion estimation to fully exploit the video sequences of time and spatial correlation information. Since the heavy computing burden of the motion estimation and compensation task in these video compression standards, the encoder is overwhelmingly more complex than the decoder; for example, the H.264/AVC encoder/decoder complexity ratio is in the order of 10 for basic configurations and can grow up to 2 orders of magnitude for complex ones [2, 3]. So traditional standardized video coding technologies are difficult to meet the low-complexity coding of the new video application in WVSNs.

To address this above requirement, distributed video coding (DVC) which first practical schemes proposed in [4, 5] is a solution because it is based on Slepian-Wolf [6] and Wyner-Ziv theories [7] and relies on a new statistical framework, instead of the deterministic approach of conventional coding techniques. DVC allows shifting the complexity from the encoder to the decoder and encoding with very low complexity then gives the decoder the task to exploit the source statistics to achieve efficient compression. DVC is also well suited for camera/video sensor networks, where the correlation across multiple views can be exploited at the decoder, without communications between the cameras/video sensor nodes [8]. This kind of coding scheme for multiview distributed videos is called multiview distributed video coding (MDVC).

No matter the conventional DVC or MDVC is, the side information (SI) generation is the key link in coding framework. But different from traditional monoview DVC, the SI generation of MDVC can be computed not only from previous and next decoded frames in the same view but also from frames in other spatially proximal views. Meanwhile, DVC and MDVC as source compression and coding scheme transform coding have over the years emerged as the dominating compression strategy. Transformations are decomposition and representation of the image information. Energy of transformed image focused on the transform domain determines the object of quantization coding, which is the core part of image and video coding. However, the transformation itself will not bring about distortion and losses of image information. The distortion segment of image and video coding is quantization process. In traditional image and video coding, standards such as JPEG and JPEG 2000 and video compression standards such as H.26x, the discrete cosine transformation (DCT), and quantization based on divided image blocks all lead to coding effect. So in this paper, we study the MDVC coding and propose a new MDVC framework, which is constituted by video coding scheme and image postprocessing scheme in WVSNs. Our main contributions include the following.

(1) Based on the video sensor characteristic of WVSNs, we present a new MDVC scheme, which not only contains the temporal SI fusion but also includes the spatial SI fusion method.

(2) For distortion issue caused by quantization processing of image and video coding, to further improve the quality of decoded video, image postprocessing based on spatial, using deblock effect and dering effect on the decoded video is designed.

The remainder of this paper is organized as follows. Section 2 discusses some previous works on DVC and MDVC which motivated our work. Section 3 introduces the proposed multiview distributed video coding scheme. Section 4 describes the image postprocessing scheme in the decoder. The performance evaluations of the proposed framework are presented in Section 5. Finally, conclusions and future work are derived in Section 6.

2. Related Works

DVC is the important application of distributed source coding (DSC) for video coding. The theoretical basis of DSC is Slepian-Wolf theorem [6] and Wyner-Ziv theorem [7]. Theory of Slepian-Wolf shows that even if correlated sources are encoded without getting information from each other, coding performance can be as good as dependent encoding if the compressed signals can be jointly decoded. Wyner-Ziv theory extends this conclusion to the lossy source coding with side information. The Slepian-Wolf and Wyner-Ziv theorems suggest that it is possible to compress two statistically dependent signals in a distributed way (separate encoding, joint decoding), approaching the coding efficiency of conventional predictive coding schemes (joint encoding and decoding). Based on these theorems, DVC has emerged and became a hot research topic rapidly [4, 5, 9–12]. The typical DVC solutions are Berkeley WZ video codec [4, 11] and Stanford WZ video coding architecture [5, 10]. The Berkeley WZ video coding solution is mainly characterized by block-based coding with decoder motion estimation, works at block level, and does not require a feedback channel; the Stanford architecture is mainly characterized by frame-based Slepian-Wolf coding, typically using turbo codes, and a feedback channel to perform rate control at the decoder.

Along with the rise of distributed camera/video network and the development of Multiview video coding, the architecture of DVC is considered using in multiview video coding [13–17]. In [13], DVC strategy is first extended to multiview video coding, and a more flexible side information generation algorithm considering both temporal and view-directional correlations is proposed to achieve high prediction accuracy. In [14], video sensors are arranged in an array to monitor the same scene from different view points. The impact of disparity fields at the central decoder and how to estimate the centralized disparity compensation at the decoder to improve the efficiency of the video sensor networks are discussed. In [15], based on multiview videos and DVC, a scheme for coding video surveillance camera networks is introduced. Then a new fusion technique between temporal side information and homography-based side information is proposed to improve the rated-distortion performance. In [16], based on WVSNs, a low-complexity video compression algorithm that uses the edges of objects in the frames to estimate and compensate

for motion is put forward, and two schemes that balance energy consumption among nodes in a cluster on a WVSN are proposed. In [17], based on wireless multimedia sensor networks (WMSNs), a power-rate-distortion (PRD) optimized resource-scalable low-complexity multiview video encoding scheme is proposed, and resource allocation achieved at the encoder while optimizing the reconstructed video quality is discussed.

From the above existing works in DVC and MDVC, lots of outstanding accomplishments have achieved, but there are some shortcomings still existing; for example, some coding architectures need feedback channel to perform rate control at the decoder, which would result in a large amount feedback loops. Obviously, this is unrealistic when the sensor node scale of WVSNs is very tremendous. Another example, in the encoder, the type of the more than half of the encoded frames is traditional intracoded in current most MDVC schemes. The coding of these frames is complexity and inefficiency. Moreover, using the image postprocessing technique could effectively improve the quality of decoded video in decoder, which is rarely involved in existing MDVC scheme. Meanwhile, based on turbo or LDPC, the Wyner-Ziv frame of MDVC in all regions without distinction, the side information is all fused by the temporal SI and interview spatial SI.

For the above problems, in this paper, we propose an improved MDVC coding and postprocessing framework. First, in the encoder of main perspective, according to the Sum of absolute difference (SAD) criteria, we differentiate the motion intense regions (MIRs) and the nonmotion intense regions (NMIRs) of a raw video frame and encode severally. In the decoder, for the MIR, the side information (SI) is generated by fusion temporal SI and interview SI. But for the NMIR, we directly use the temporal SI, the scheme utilizes motion compensated temporal interpolation (MCTI) to generate temporal SI. After the above steps in frame SI generation, we process the image postprocessing of the single decoded frame. Then mix the decoded frames in order to produce the decoded video.

3. Proposed Multiview Distributed Video Coding Scheme

In this section, we present a new MDVC scheme for WVSNs. First, we introduce the principles of DVC, then the frame and coding structure, the temporal, and spatial SI calculation techniques based on SAD criteria, and SI mask fusion method of this proposed MDVC scheme is described.

3.1. Principles of DVC. As we know, the principles of DVC are Slepian-Wolf [6] and Wyner-Ziv [7] theorems. The rate boundaries defined by the Slepian-Wolf theorem for the independent encoding and joint decoding of two statistically dependent discrete random independent and identically distributed (i.i.d.) sources are illustrated in Figure 1.

In Figure 1, X and Y are two i.i.d. random variables/sequences; that is, the raw signals, $H(X)$ and $H(Y)$ are the entropies of X and Y. R_X and R_Y are the bit rates of

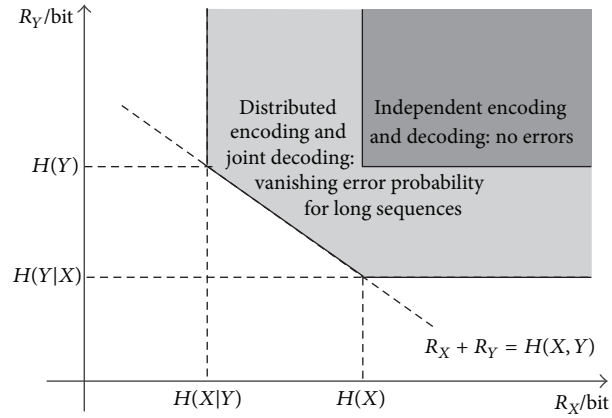


Figure 1: Rate boundaries defined by the Slepian-Wolf theorem.

lossless coding of X and Y, respectively. In traditional entropy encoding, we can get $R_X \geqslant H(X)$ and $R_Y \geqslant H(Y)$. According to the information theory, X and Y can be encoded using joint coding at the bit rate of conditional entropy $R_X \geqslant H(X \mid Y)$ and $R_Y \geqslant H(Y \mid X)$, respectively, and the total bit rate is their joint entropy $R_X + R_Y \geqslant H(X,Y)$. Conversely, with distributed coding, these two signals are independently encoded but jointly decoded. In this case, the Slepian-Wolf theorem proves that the minimum rate is still $H(X,Y)$ with a residual error probability which tends towards 0 for long sequences. In other words, Slepian-Wolf coding allows the same coding efficiency to be asymptotically attained.

Subsequently, Wyner-Ziv theorem extended the Slepian-Wolf theorem by characterizing the achievable rate-distortion region for lossy coding with SI. Wyner-Ziv theorem studied a particular case of distributed source coding, asymmetric coding, that deals with lossy compression of source X associated with the availability of the Y source at the decoder but not at the encoder, and Y (or a derivation of Y) is known as side information. A conclusion is derived that, typically, there is a rate loss incurred when the side information is not available at the encoder. But when performing independent encoding with side information under certain conditions, that is, when X and Y are jointly Gaussian, memoryless sequences and a mean-squared error distortion measure are considered. There is no coding efficiency loss with respect to the case when joint encoding is performed, even if the coding process is lossy. The structure of Wyner-Ziv codec is shown in Figure 2. Together, the Slepian-Wolf and Wyner-Ziv theorems suggest that it is possible to compress two statistically dependent signals in a distributed way, namely, separate encoding, joint decoding, approaching the coding efficiency of conventional joint encoding, and decoding predictive coding schemes. And in general, DVC uses Wyner-Ziv coding scheme as its lossy particular case [18, 19].

3.2. Frame and Coding Structure of the Proposed MDVC. In this section, based on the sensor characteristic of WVSNs, we propose a feasible MDVC scheme, which could be regarded as an extension of conventional DVC. In a WVSN, video sensors are separated at certain distances and angles by each

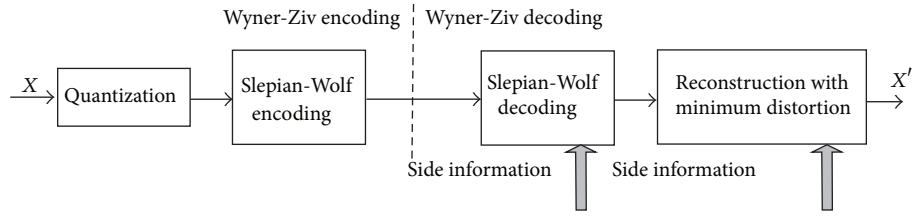


Figure 2: The structure of Wyner-Ziv codec.

other. All nodes synchronized shoot the same scenario in an area of interest of WVSN, and the relevant video sequences are produced. Then the encoder, that is, the video sensor node, encodes the captured sequences in order independently. In encoder, MDVC does not use the complexity encoder for encoding, and interview does not perform data communication. It exploits the source statistics of intraview and interview to obtain high quality decoded video, so it is superior to traditional multiview coding. The frame and coding structure of the proposed MDVC scheme is shown in Figure 3. Multiview video frames are classified into two categories: key frames and Wyner-Ziv frames, noted by K and WZ, respectively. The $C_v, C_{v-1}, C_{v+1}, \ldots$ denote the video sensors in adjacent views, and the $F_t, F_{t-1}, F_{t+1}, \ldots$ represent the successive frames collected by the video sensors with time order. In encoder, the key frames are encoded by the intraframe codec with the traditional DCT based intracoding method, and the Wyner-Ziv frames are encoded by Wyner-Ziv codec. In decoder, the key frames are individually decoded by conventional intraframe decoder and the Wyner-Ziv frames are joint decoded by fusion information of the temporal SI from previous and next key frames in same view and spatial SI from the key frames in adjacent two views. Since Wyner-Ziv frames can be intraencoded and inter decoded, the whole scheme consists of independent encoder and joint decoder. Figure 4 shows a sample of WZ frame coding scheme in MDVC. All key frames are separately intraframe encoding and decoding. For the Wyner-Ziv frame, we use intraframe encoding and interviews decoding with temporal SI and spatial SI.

3.3. MDVC Scheme Based on the SAD Criteria. The key technique of the MDVC is about exploiting both temporal and interview correlations in an efficient way. Most of the existing works on DVC/MDVC are based on on turbo or LDPC, the Wyner-Ziv frame coding in all regions without distinction, motion estimation techniques cannot accurately predict the area which are more intense exercise. The decoder needs to request more feedback information, thus not only the rate increases, but the decoded image is still not accurate enough. For this problem, we propose an improved MDVC scheme based on SAD criteria.

In some video sequences, motion vector of many macroblocks is equal to zero or very small, so only a small part of macroblocks has large moving. For the macroblocks with motion vector same as being zero or very small, motion compensated temporal interpolation (MCTI) can make a good predictive coding performance. While for other intense

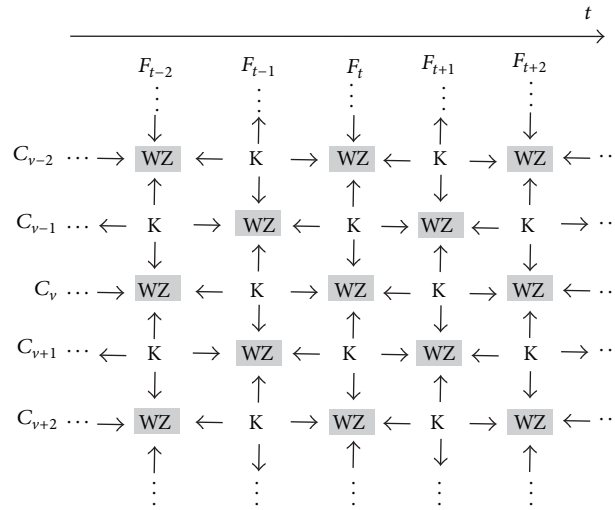


Figure 3: Successive timestamp frame structure for MDVC scheme in a WVSN.

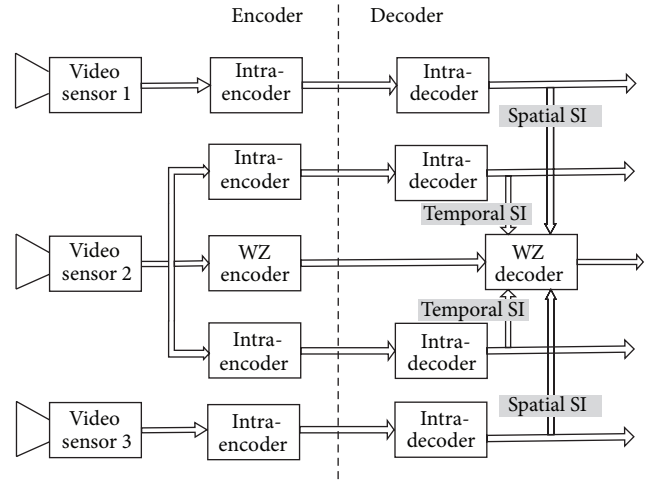


Figure 4: A sample of Wyner-Ziv frame coding scheme in MDVC.

motion macroblocks, decoder needs to use information from other cameras. In the encoder of Wyner-Ziv frame, according to the SAD criteria, we can get the motion intense regions (MIRs) and the nonmotion intense regions (NMIRs). For the MIR, the side information is generated by fusion temporal side information and inter-camera side information at the pixel level. Inversely, for the NMIR, we directly use

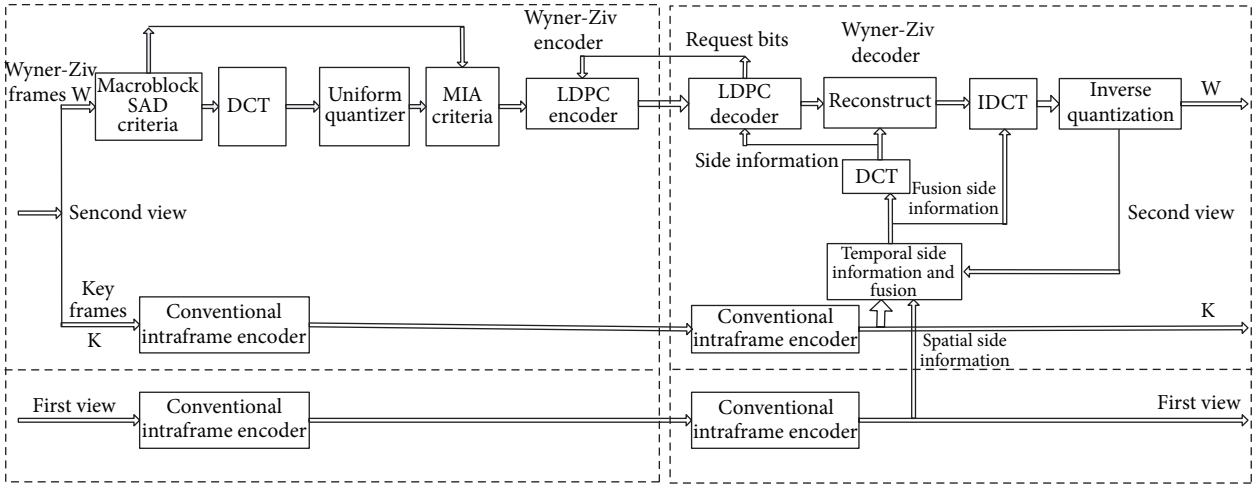


Figure 5: Block diagram of the encoder/decoder of the proposed MDVC Scheme based on SAD.

the temporal side information. Encoder uses SAD criteria to get MIR and NMIR macroblock and mark the MIR macroblock. Suppose that M_w is the current frame and M_p is the previous reference frame, so SAD of a macroblock is computed as (1). Consider

$$\mathrm{SAD} = \sum_{(x,y)\in B_i} \left| M_w(x,y) - M_p(x,y) \right|. \tag{1}$$

B_i represents an 8×8 macroblock of the image, x, y is, respectively, horizontal and vertical coordinates of a pixel. If SAD $\geqslant T$, we regard the macroblock as MIR. An adequate threshold T has been found experimentally.

The detailed block diagram of the encoder/decoder of the proposed MDVC Scheme based on SAD criteria is illustrated in Figure 5. We use two video sensor views to present the encoding and decoding process of Wyner-Ziv frames for brevity. The K frames of the second view are encoded and decoded by conventional intraframe scheme, while for the WZ frames, according to the SAD criteria, we get MIR (8×8 macroblock) and NMIR (8×8 macroblock). In encoder, mark the MIR and NMIR macroblock, respectively. The K frame in first view is used conventional intracoding mode, and the decoded video stream is transformed into the interview SI for WZ frame of the second view by homography compensated interview interpolation (HCII). Side information is used in the reconstruction, to obtain the decoded DCT coefficients; then, IDCT and IQ (inverse quantization) are applied to generate the decoded frame W', as the structure of Wyner-Ziv codec shown in Figure 2.

3.4. Side Information Generation by Fusing. Unlike DVC, the generation of SI in MDVC must synthetically consider the temporal SI and interviews spatial SI. In this paper, for the temporal SI, we choose to use motion compensated temporal interpolation (MCTI) technique estimating temporal motion vector from the previous frame towards the forward frame. Then, the motion vectors are interpolated at midpoint to generate the SI [13, 15].

For the interviews spatial SI, in the scene of WVSN, all nodes synchronized monitor the same scenario in an area of interest, and the video sensor network contains a large number of interview statistical dependencies. The position and view of the cameras have been fixed; homography matrix [20] could be used to generate the interview SI from other cameras. The homography is a 3×3 matrix that relates video sensor 1 to video sensor 2 in the homogenous coordinates system. There are 8 parameters in matrix, we use a gradient descent method to get the parameters [21]. The pixel point of video sensor 1 is mapped to a pixel point (x_2, y_2) of video sensor 2 up to a scale μ such that

$$\mu \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} b_0 & b_1 & b_2 \\ b_3 & b_4 & b_5 \\ b_6 & b_7 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix},$$

$$x_2 = \frac{b_2 + b_0 x_1 + b_1 y_1}{b_6 x_1 + b_7 y_1 + 1}, \tag{2}$$

$$y_2 = \frac{b_5 + b_3 x_1 + b_4 y_1}{b_6 x_1 + b_7 y_1 + 1}.$$

Harris corner and edge detector [22] also could be used in iteratively calculating and adjusting the parameters in homography of two video sensor nodes. Then, we can use homography to generate the spatial SI of the Wyner-Ziv frame in adjacent sensor view.

When the temporal SI and spatial SI of a preparing Wyner-Ziv frame have generated, we present a binary masks fusion method to generate SI according the MIR or NMIR type of macroblocks transmitted from encoder. Figure 6 shows how to fuse the binary masks from two interview sensors, which is done by a simple logical OR operation. We look for the pixel that predicts Wyner-Ziv frame better from both side information. In binary mask, 0 represents that the pixel from the same position of interview SI is reliable and 1 represents that the pixel from the same position of temporal SI is reliable. In the fusion process of the MIR, we take a simple difference operation between the current

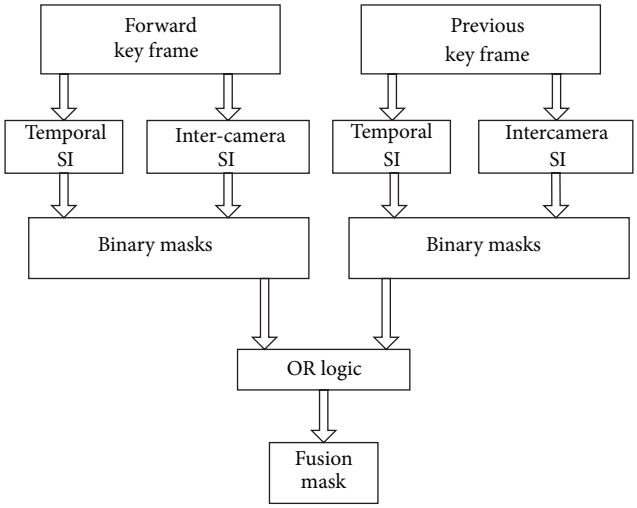


Figure 6: Side information fusion of Wyner-Ziv frame in decoder.

pixel from the previous key frame and the pixel an the same position from interview SI to obtain a value of A and take a simple difference operation between the current pixel from the previous key frame and the pixel at the same position from Temporal side information to obtain a value of B. If $|A| \geqslant |B|$; we set binary mask to 1; otherwise, we set binary mask to 0. For the NMIR, we set binary mask to 0, and it save a lot of computing time. We perform the same process with forward key frame. Thus, we obtain a second binary mask. Finally, we perform an OR logic operation between both binary masks of MIR to obtain the binary fusion mask.

4. Decoded Image Postprocessing Based on Spatial Domain

In Wyner-Ziv DVC and MDVC, quantification is a significant step. But the quantization matrix or quantization coefficient adopted in quantization process would affect the quality of image decoding. In image processing, the most common divided blocks processing technique, discrete cosine transforms (DCT), and quantization could lead to coding artifacts, such as blocking and ringing artifacts. In this section, we try to conduct the image postprocessing for the video decoded from the proposed MDVC by using filtering techniques.

4.1. Deblocking Filter. The quantization process for DCT coefficients causes the blocking effect during image and video coding. Because the quantization error is different in each block, the coding based on blocks, for example, 4×4, 8×8 (which is adopted in this paper), or 16×16, brings to the discontinuity in the border of adjacent blocks. In video coding, motion compensation would spread the blocking effect to the prediction coding frames. This is because the location of blocking effect is unfixed, and it would move with the motion compensation. The key technique of blocking effect eliminating is how to eliminate the blocking effect with protecting the real edges in the border of blocks. In general, the blocking effect eliminating algorithm should just handle the pixels near the border of block.

The algorithm of deblocking filter along the border of the 8×8 macroblock executes one-dimension filtering in the decoder. First, using horizontal filter eliminate the horizontal blocking effect; then, using vertical filter eliminates the vertical blocking effect. If a pixel had changed in the last filtering process, the modified pixel would be used in the next filtering. The filtering process has three modules: mode selection, DC offset mode for smooth area, and default mode for complex area.

To eliminate the blocking effect more powerful, in this paper, according the satisfied condition of near pixels at the border of 8×8 blocks, different filtering patterns are adopted in filtering process: DC filter of strong filtering for smooth area and default mode for pixels in the border of blocks for complex area.

Suppose that $p_0, p_1, \ldots, p_9$ are ten consistent pixels in horizontal direction, and the value of eq_cnt denotes the smooth level of the near image of blocks border; and T_1 is a small threshold, we have

$$\text{eq\_cnt} = \sum_{i=0}^{8} \phi\left(p_i - p_{i+1}\right), \tag{3}$$

where

$$\phi(\sigma) = \begin{cases} 1 & \text{if } |\sigma| \leqslant T_1, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

If present pixels $p_0, p_1, \ldots, p_9$ are in smooth area, the value of eq_cnt would be relatively great. On the contrary, if the present pixels are in complex area, the value of eq_cnt would be very small. We employ (5) to determine the desired filtering mode, DC offset mode or default mode, denoted by filter_mode. Consider

$$\text{filter\_mode} = \begin{cases} \text{DC\_offset} & \text{if eq\_cnt } T_2, \\ \text{default} & \text{otherwise.} \end{cases} \tag{5}$$

Suppose that T_2 is a threshold larger than T_1. Examining the blocking effect areas cause by little DC offset. If eq_cnt $\geqslant$ T_2, the blocking area is smooth area and DC filtering mode is adopted. Inversely, the blocking area is complex area and default filtering mode is selected.

4.2. Deringing Effect Filter. The coarse quantification of the high-frequency components of image results in ringing effect. If high-frequency components, which corresponding to the strong edges of image, such as high contrast, occur quantization error, the nearby region of the strong edges would appear to be fake edge. The core of the ringing effect eliminating algorithm is how to distinguish the ringing and real edge. The most common method is examining the edge of image first, then using low-pass filter screening the no-edge pixels to achieve the objective of eliminating ringing effect.

The deringing effect filter includes threshold determination, identifier evaluation and adaptive smoothing. Every pixel in 8×8 macroblock all needs to filter. In fact, 10×10

pixels are used in the process of every 8 × 8 macroblock filtering.

In threshold determination module, first, we divide the 8×8 macroblock to 4 smaller blocks, then find the maximum and minimum grey value max[i] and min[i] of the divided ith block. Calculate the threshold thr[i] and the dynamic range of grey range[i] by (6) and (7), respectively. Consider

$$\text{thr}[i] = \frac{\max[i] + \min[i] + 1}{2}, \tag{6}$$

$$\text{range}[i] = \max[i] - \min[i]. \tag{7}$$

Calculate the dynamic range of 4 luminance blocks, and number the block, which has maximum dynamic range; as $i_{\max}$, we have

$$\max\_\text{range} = \text{range}\left[i_{\max}\right]. \tag{8}$$

Revise the threshold of 4 smaller blocks by (9). Consider

$$\underset{i\in[0,3]}{\text{thr}[i]} = \begin{cases} \text{thr}\left[i_{\max}\right] & \text{if } \left(\text{range}[i] < 32 \,\&\&\, \max\_\text{range} \geqslant 64\right), \\ 0 & \text{if } \max\_\text{range} < 64. \end{cases} \tag{9}$$

In identifier evaluation module; after threshold determined, the following operations are all in original 8 × 8 macroblock. The grey value and binary identifier of the pixel (h, v) are denoted by rec(h, v) and bin(h, v), respectively. The determination criterion of binary identifier is shown in (10). Consider

$$\underset{h,v\in[0,7]}{\text{bin}(h, v)} = \begin{cases} 1 & \text{if rec}(h, v) \geqslant \text{thr}, \\ 0 & \text{otherwise}, \end{cases} \tag{10}$$

where thr is the threshold of the current block i. The purpose of binary identifier processing is to distinguish the grey value with the range with thr in block.

Adaptive smoothing module is constituted by adaptive filtering and numerical pruning two parts. Calculate the binary identifiers of all the pixels in the 3×3 window, whose center is current pixel. If all binary identifiers are "1" or "0", this region is smoothed area, so to filter this current pixel by smooth filtering. The reconstructed value after filtering $\text{flt}'(h, v)$ is computed by (11). Consider

$$\text{flt}'(h, v) = \frac{1}{16}\left[\sum_{i=-1}^{1}\sum_{j=-1}^{1} \text{coef}(i, j) \cdot \text{rec}(h + i, v + j) + 8\right], \tag{11}$$

where coef(i, j) denotes the coefficient, $i, j = -1, 0, 1$, (i, j) represents the location of pixel in 3 × 3 image window.

In order to prevent excessive handling of pixel, the gray level difference dif(h, v) between the reconstructed values with smooth filtering and the original pixel value must change limiting in the range from the change of quantization parameters. $\text{fit}''(h, v)$ and $\text{fit}'(h, v)$ indicate the reconstructed value with smooth filtering and the without limited reconstructed value pixel value. We have

$$\text{flt}''(h, v) = \begin{cases} \text{rec}(h, v) + \max\_\text{dif} & \text{if } (\text{dif}(h, v) > \max\_\text{dif}), \\ \text{rec}(h, v) - \max\_\text{dif} & \text{if } (\text{dif}(h, v) < \max\_\text{dif}), \\ \text{flt}'(h, v) & \text{otherwise}, \end{cases} \tag{12}$$

where OP denotes quantization parameter and max_dif is OP/2 no matter the macroblock in internal coding or non-internal coding.

5. Performance Evaluation

In this section, we design some simulations to evaluate the effectiveness of the proposed coding and postprocessing framework by using public and representative multiview video sequences.

We use the two multiview sequences: "Exit" and "Vassar" which are made public available by Mitsubishi Electric Research Laboratories (MERL) [23]. For reasons of computation complexity, the spatial resolution was halved from VGA (640×480, YUV 4 : 2 : 0) to QCIF (176×144, YUV 4 : 2 : 0). For both, the time resolution is 25 fps and we used the 2 cameras with 100 frames per view, but only luminance component is evaluated for per frame. Our simulated environment is Microsoft Visual Studio 2005 on Windows XP SP3 system with Intel Core 2 CPU 2.40 GHz and 2.00 GB memory. We extend the exiting Wyner-Ziv video codec [24] to MDVC, and all codes are written in ANSI C++. In the experiments, LDPC codes are generated by PEG algorithm [25], and the rate of LDPC code is 7/8. After several experimental analyses and comparisons, 64 is the ideal threshold of SAD criteria. According to the frame structure shown in Figure 3, the video streams of Camera 0 and Camera 1 all use distributed video coding, but the key frames and Wyner-Ziv frames coding sequence for Camera 0 and Camera 1 are "W-K-W-K" and "K-W-K-W", respectively. The key frames K and Wyner-Ziv frames W are alternative coding, and the key frame of Camera 1 is used to assist the Wyner-Ziv frame decoded in Camera 0, which is main perspective in our experiments.

To evaluate the rate-distortion performance of our proposed scheme, we compare the following five schemes:

(1) H.263+ "I-I-I-I": H.263 intraframe coding (I-I-I-I). H.263+ codec uses TMN8 [26].

(2) H.263+ "I-P-I-P": H.263 interframe coding (I-P-I-P). Like H.263+ "I-I-I-I" scheme, the codec used also is TMN8, but in different coding options.

(3) Only Temporal SI: MDVC with only temporal SI used in side generation.

(4) Only Temporal SI: MDVC with only spatial SI used in side generation.

(5) Fusion SI: MDVC with fusion temporal and spatial SI used in side generation.

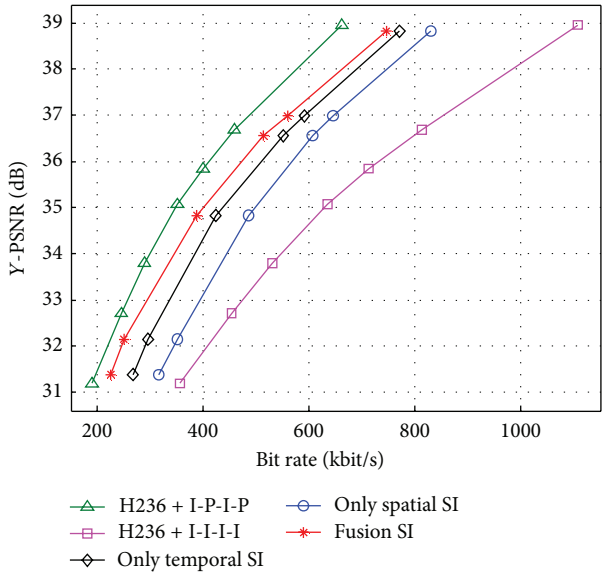


Figure 7: Luminance PSNR versus average bit rate for different coding schemes of "Exit" multiview sequences.

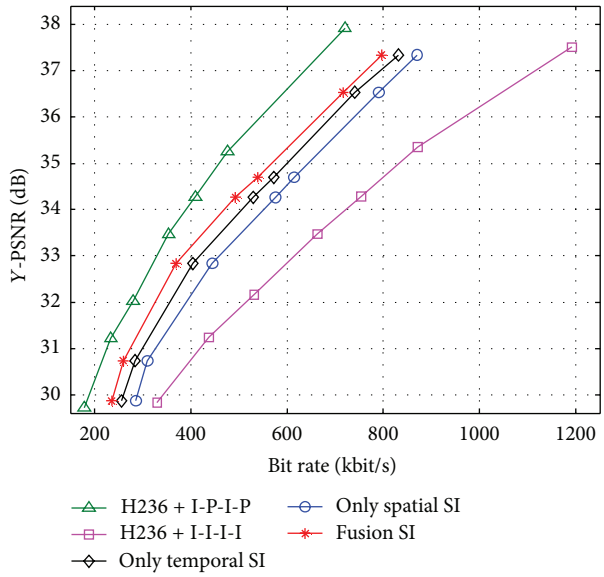


Figure 8: Luminance PSNR versus average bit rate for different coding schemes of "Vassar" multiview sequences.

The luminance PSNR performance as a function of the different average bit rate for the two multiview sequences "Exit" and "Vassar" is shown in Figures 7 and 8, respectively. We can find that the proposed MDVC scheme has significantly better performance 2-3 dB than that of H.263 intraframe coding, and MDVC system is less than the overall complexity of the H.263+ coding. Although there is still a performance gap between H.263 interframe coding and our proposed MDVC scheme, our scheme is easier at encoder which is fit for WVSNs. The proposed fusion SI approach can bring improvements up to 0.2–0.5 dB when compared to the MDVC with only temporal SI used in the same average bit rate. Our proposed MDVC scheme can gain more accurate motion estimation in the intense motion region, to save a lot of computing time. Figures 9(a) and 9(b), respectively, is 15th decoded frame (WZ frame) of "Exit" sequence and "Vassar" sequence using our proposed MDVC scheme; Figures 10(a) and 10(b), respectively, is 15th decoded frame (WZ frame) to "Exit" sequence and "Vassar" sequence using temporal SI. From the comparisons, it can be seen that the decoded frame in fusion SI in MDVC is significantly better than nonfusion scheme, for instance, only temporal SI scheme.

(a) PSNR = 36.12

(b) PSNR = 34.82

Figure 9: The decoded 15th frame in proposed fusion SI scheme. (a) "Exit". (b) "Vassar".

For evaluating the performance of postprocessing framework proposed in this paper, we set up a comparison experiments with the postprocessing and without postprocessing for the decoded multiview sequences. The comparison results of average YUV-PSNR versus QP with decoded "Exit" and "Vassar" sequences are shown in Figure 11. We can find that the postprocessing method could provide an additional improvement of about 0.1 dB to the decoded video sequences from Figure 11. The decoded video after the postprocessing filter can be reference data in the subsequently image processing and is useful to enhance the compression image quality in objective and subjective.

(a) PSNR = 35.81

(b) PSNR = 34.49

FIGURE 10: The decoded 15th frame in only Temporal SI scheme. (a) "Exit". (b) "Vassar".

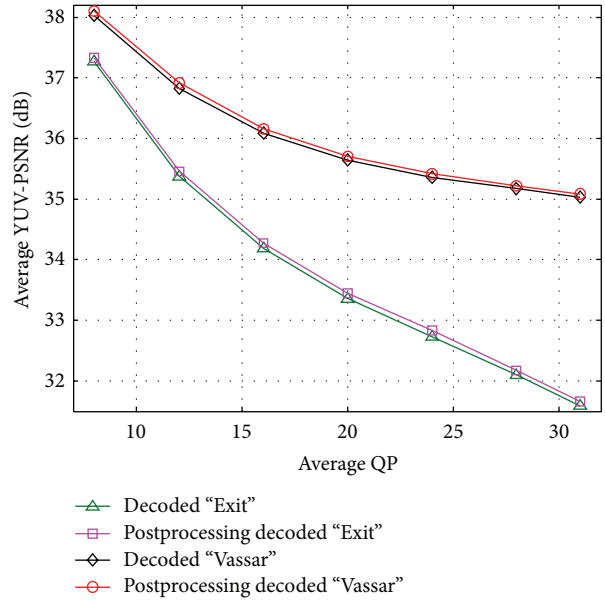


FIGURE 11: The decoded 15th frame in only Temporal SI scheme. (a) "Exit". (b) "Vassar".

6. Conclusions and Future Work

In this paper, we proposes an improved coding and postprocessing framework for multiview distributed video in WVSNs. In coding scheme, through distinguishing the motion intense regions and the nonmotion intense regions based on sum of absolute difference criteria, the coding scheme encodes macroblocks adaptively, and a fusion temporal and spatial side information is adopted to improve the quality of side generation in the decoder. To further enhance the quality of decoded video sequences, a postprocessing scheme is designed to get more additional compression gain. Experiments demonstrate the validity of the proposed framework.

In the postprocessing scheme, we enforce the image postprocessing based on spatial domain without considering time relevance of the previous and next frames, so the image quality gain we got in this scheme is limited. In our future work, we would consider talking about the postprocessing based on temporal filtering to guarantee the temporal continuity in the reconstructed video sequences to get higher video quality and more compression gain.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant nos. 61171053, 61300239, and 61373137; the Doctoral Fund of Ministry of Education of China under Grant no. 20113223110002; the Science & Technology Innovation Fund for Higher Education Institutions of Jiangsu Province under Grant no. CXZZ12_0480.

References

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[2] S. Saponara, C. Blanch, K. Denolf, and J. Bormans, "The JVT advanced video coding standard: complexity and performance analysis on a tool-bytool basis," in *Proceedings of the International Packet Video Workshop (PV '03)*, Nantes, France, April 2003.

[3] J. Ostermann, J. Bormans, P. List et al., "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, 2004.

[4] R. Puri and K. Ramchandran, "PRISM: a new robust video coding architecture based on distributed compression principles," in *Proceedings of the Allerton Conference on Communication, Control and Computing*, Allerton, Ill, USA, October 2002.

[5] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Proceedings of the 36th Asilomar Conference on Signals Systems and Computers*, pp. 240–244, Pacific Grove, Calif, USA, November 2002.

[6] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[7] A. D. Wyner and J. Ziv, "The rate-distortion functions for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[8] F. Dufaux, W. Gao, S. Tubaro, and A. Vetro, "Distributed video coding: trends and perspectives," *EURASIP Journal on Image and Video Processing*, vol. 2009, Article ID 508167, 2009.

[9] D. Varodayan, A. Aaron, and B. Girod, "Exploiting spatial correlation in pixel-domain distributed image compression," in *Proceedings of the International Picture Coding Symposium (PCS '06)*, Beijing, China, April 2006.

[10] A. Aaron, D. Varodayan, and B. Girod, "Wyner-Ziv residual coding of video," in *Proceedings of the International Picture Coding Symposium (PCS '06)*, Beijing, China, April 2006.

[11] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: a video coding paradigm with motion estimation at the decoder," *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2436–2448, 2007.

[12] Q. Xu and Z. Xion, "Layered Wyner-Ziv video coding," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3791–3803, 2006.

[13] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Distributed multiview video coding," in *Visual Communications and Image Processing (VCIP)*, vol. 6077 of *Proceedings of SPIE*, San Jose, Calif, USA, January 2006.

[14] M. Flierl and B. Girod, "Coding of multi-view image sequences with video sensors," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '06)*, pp. 609–612, Atlanta, Ga, USA, October 2006.

[15] M. Ouaret, F. Dufaux, and T. Ebrahimi, "Multiview distributed video coding with encoder driven fusion," in *Proceedings of the 15th European Signal Processing Conference (EUSIPCO '07)*, Poznan, Poland, September 2007.

[16] P. N. Huu, V. Tran-Quang, and T. Miyoshi, "Video compression schemes using edge feature on wireless video sensor networks," *Journal of Electrical and Computer Engineering*, vol. 2012, Article ID 421307, 20 pages, 2012.

[17] L.-W. Kang, C.-S. Lu, and C.-Y. Lin, "Low-complexity video coding via power-rate-distortion optimization," *Journal of Visual Communication and Image Representation*, vol. 23, no. 3, pp. 569–585, 2012.

[18] P. L. Dragotti and M. Gastpar, *Distributed Source Coding: Theory, Algorithms and Applications*, Academic Press, New York, NY, USA, 2009.

[19] H. Bai, A. Wang, Y. Zhao, J.-S. Pan, and A. Abraham, *Distributed Multiple Description Coding: Principles, Algorithms and Systems*, Springer, London, UK, 2011.

[20] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2011.

[21] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 497–501, 2000.

[22] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, Manchester, UK, August 1988.

[23] Mitsubishi Electric Research Laboratories, "MERL multiview video sequences," ftp://ftp.merl.com/pub/avetro/mvc-testseq/.

[24] D. Chen, D. Varodayan, M. Flierl, and B. Girod, "Wyner-Ziv video codec with unsupervised motion learning," http://msw3.stanford.edu/~dchen/DVC/.

[25] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, 2005.

[26] Video codec test model, ITU-T/SG15, TMN8, Portland, Ore, USA, June 1997.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 478470, 12 pages
http://dx.doi.org/10.1155/2013/478470

Research Article

Coverage Optimization Algorithm Based on Sampling for 3D Underwater Sensor Networks

Du Xiaoyu,[1,2] Sun Lijuan,[1,3] and Liu Linfeng[1]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China*
[2] *Basic Experiment and Teaching Center, Henan University, Kaifeng 475003, China*
[3] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China*

Correspondence should be addressed to Sun Lijuan; sunlj@njupt.edu.cn

Received 9 June 2013; Accepted 2 August 2013

Academic Editor: Han Zhijie

We discuss the problem of maximizing the sensor field coverage for a specific number of sensors while minimizing the distance traveled by the sensor nodes. Thus, we define the movement task as an optimization problem that involves the adjustment of sensor node positions in a coverage optimization mission. We propose a coverage optimization algorithm based on sampling to enhance the coverage of 3D underwater sensor networks. The proposed coverage optimization algorithm is inspired by the simple random sampling in probability theory. The main objective of this study is to lessen computation complexity by dimension reduction, which is composed of two detailed steps. First, the coverage problem in 3D space is converted into a 2D plane for heterogeneous networks via sampling plane in the target 3D space. Second, the optimization in the 2D plane is converted into an optimization in a line segment by using the line sampling method in the sample plane. We establish a quadratic programming mathematical model to optimize the line segment coverage according to the intersection between sensing circles and line segments while minimizing the moving distance of the nodes. The intersection among sensors is decreased to increase the coverage rate, while the effective sensor positions are identified. Simulation results show the effectiveness of the proposed approach.

1. Introduction

A wireless sensor network (WSN) consists of a large number of resource-limited (such as CPU, storage, battery power, and communication bandwidth) tiny devices, which are called sensors. These sensor nodes can sense task specific environmental phenomenon, perform in network processing on the on the sensing field and communicate wirelessly with other sensor nodes or with a sink (also called data gathering node), usually via multihop communications [1]. Water will be the material basis of human sustainable development for affording food, raw materials, and living space for the survival of mankind in the future [2]. However, people cannot easily monitor and develop resources in a large scale because underwater environment is still more serious to human being. Therefore a lot of science and technology communities in different countries have focused on monitoring and exploring underwater resources reasonably by various technologies [3]. Underwater sensor networks are envisioned to enable applications for oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention, assisted navigation, and tactical surveillance applications [4], so it will play an important role in monitoring and developing water resource. The coverage and connectivity are fundamental issues in WSNs, which can determine the scope of services provided by the network, and have a great influence on the network cost and performance of specific applications. The coverage can be considered as a measure of the service quality of WSNs [5, 6].

We analyze the coverage issues in 3D underwater sensor networks (USNs), where sensors are randomly deployed in a 3D field (Figure 1). Underwater sensor nodes are placed in 3D space, and the computational complexity of the coverage optimization algorithm in 3D space is higher than the coverage optimization algorithm in a 2D plane. Therefore, some applications cannot be effectively modeled in 3D space. We propose a 3D coverage optimization algorithm

based on sampling (COS) that combines the concepts of simple random sampling in statistics [7] and optimization algorithms [8]. First, we transfer the coverage in 3D space into a 2D plane for heterogeneous networks by plane sampling in a target 3D space. Second, we convert the plane coverage to line segment coverage by the sampling line in the sample plane. Finally, we establish a quadratic programming mathematical model to optimize the line segment coverage according to the intersection between sensing circles and the line segment while minimizing the moving distance of the nodes. The plane coverage should be optimized to enhance the line segment coverage. We summarize the main contributions of our study as follows.

(1) Sample random sampling is applied to the coverage problem in 3D space for dimension reduction.

(2) A quadratic programming mathematical model is established according to the intersection position of the sensing circles and sample line, position of the sensing circles, and radii of the sensing circles.

(3) We conclude that the sum of the node traveling distance for the unchanging node sequence is less than the sum of the changing node sequence. This conclusion is mathematically proven in this study.

(4) A nonlinear quadratic optimization problem is converted into a linear quadratic optimization problem by tightening constraints to obtain the suboptimum solution.

The remainder of this paper is organized as follows: Section 2 reviews prior studies on sensor network coverage and discusses related works. Section 3 provides some assumptions and key definitions and describes the system model for underwater wireless sensor networks. Section 4 discusses the development of the mathematical model for sensor network coverage and presents the COS algorithm in detail. Section 5 presents the simulation results of the COS algorithm. Finally, Section 6 concludes and outlines the direction of future works.

2. Related Works

The sensor coverage problem has been extensively studied in the field of multirobot systems and computational geometry the Art Gallery Problem (AGP) and the circumference coverage are related to the coverage of WSNs closely [9]. k-Coverage of sensor networks is one of research focuses in WSNs [1, 10–14]. Huang et al. present a polynomial time algorithm formulating circumference k coverage to a decision problem, whose goal is to determine whether every point in the service area of the sensor network is covered by at least k sensors [10] the optimal deployment density bound for two-coverage deployment patterns is derived in [11] where Voronoi polygons generated by sensor nodes are congruent. The authors design a set of optimal patterns that achieve two-coverage and one-, two-, and three-connectivity thought extending these patterns by considering the connectivity requirement. Ammari and Das focus on coverage

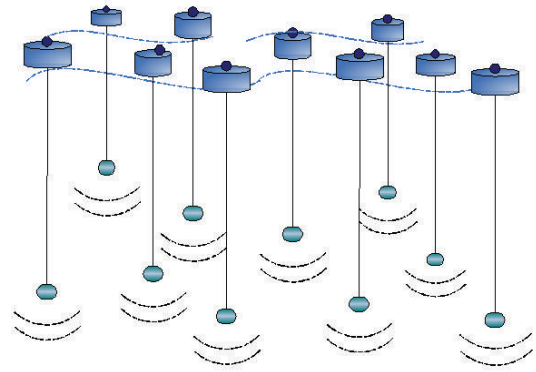


Figure 1: Underwater sensor networks.

and connectivity in 3D WSNs using a continuum percolation theory based approach [13]. The network exhibits a coverage percolation (resp., connectivity percolation) when a giant covered region (resp., giant connected component) almost surely spans the entire network for the first time. It proposes an integrated-concentric-sphere model to address coverage and connectivity in 3D WSNs in an integrated way. Ammari uses Helly's Theorem and the Reuleaux tetrahedron model for characterizing k-coverage of a 3D deployment field and develops a placement strategy of sensors to fully k-cover a 3D field [12, 13].

Coverage algorithm based on virtual force is one of research hotspots currently [15–19]. For a given number of sensors, the VFA algorithm attempts to maximize the sensor field coverage. The sensor nodes are assumed to be deployed in virtual force field; moreover, a judicious combination of attractive and repulsive forces is used to determine virtual motion paths and the rate of movement for the randomly placed sensors. Once the effective sensor positions are identified, a one-time movement with energy consideration incorporated is carried out; that is, the sensors are redeployed to these positions. In [18], Delaunay triangulation is formed with these nodes, and adjacent relationship is defined if two nodes are connected in the Delaunay diagram. Force can only be exerted from those adjacent nodes within the communication range.

The application of linear programming in coverage problem focuses on point (or target) coverage, where the objective is to cover a set of points (targets) [6, 8, 20–22]. Zhao and Gurusamy consider the connected target coverage (CTC) problem with the objective of maximizing the network lifetime by scheduling sensors into multiple sets, each of which can maintain both target coverage and connectivity. The authors determine an upper bound and a lower bound on the network lifetime for the MCT problem and then develops an approximation algorithm to solve it [20]. In [21], the sensing task of covering maximum number of targets while minimizing the energy consumption of the sensing operation is defined as an optimization problem of adjusting the sensing range parameter jointly with selection of nodes in a target coverage mission. A distributed greedy-based heuristic is proposed that nodes with less priority reduce their sensing range before their neighbors and optimal adjustment

of sensing range of active nodes is done via a dual-based algorithm. Cardei et al. model the solution as the maximum set covers problem and design two heuristics that efficiently compute the sets, using linear programming and a greedy approach [8]. In [22] the sensor node deployment task has been formulated as a constrained multiobjective optimization problem where the aim is to find a deployed sensor node arrangement to maximize the area of coverage, minimize the net energy consumption, maximize the network lifetime, and minimize the number of deployed sensor nodes while maintaining connectivity between each sensor node and the sink node for proper data transmission.

As described earlier, almost all of coverage strategies in 3D space are based on a premise that sensor nodes are deployed in a specified location and stay there so that the maximum coverage of 3D space is achieved. However, sensor nodes are usually deployed at random and are impossible suspended at specified location without any connection in water. Most of the researches on coverage and connectivity in WSNs are not suitable for underwater because of two-dimensional application only. We propose a 3D coverage optimization algorithm for underwater 3D sensor networks (USNs) in which sensor nodes are randomly deployed and are fixed by cables, in which quadratic programming is applied to coverage optimization problem in 2D plane and 3D space.

3. Assumptions and Definitions

3.1. Assumptions. We assume that a large number of underwater sensors are distributed in an area of interest. A 3D USN model is described as follows [23, 24].

(1) A 3D USN contains two types of sensor nodes: one is used for communications and is deployed on the water surface; the other is used for sensing and is deployed underwater.

(2) All sensor nodes that are deployed underwater have homogeneous models; that is, all sensors have binary sensing coverage models. Thus, the sensing model is a sphere, and all sensors adopt the radius R of the sensing sphere model.

(3) The underwater sensor node communicates with the surface node via a cable that connects the two sensors.

(4) Sensor node position and depth can be acquired via the sensor nodes deployed on the water surface and cables.

(5) The sensor nodes on the water surface are fixed in place with an anchor. However, the underwater nodes can move to a specified location vertically.

(6) Sensor nodes have enough residual energy to move the specified location because coverage algorithm is executed during network initialization.

We assume that sensor nodes are randomly deployed in a 3D cuboid space and denote the distance between two spots as a Euclidean distance.

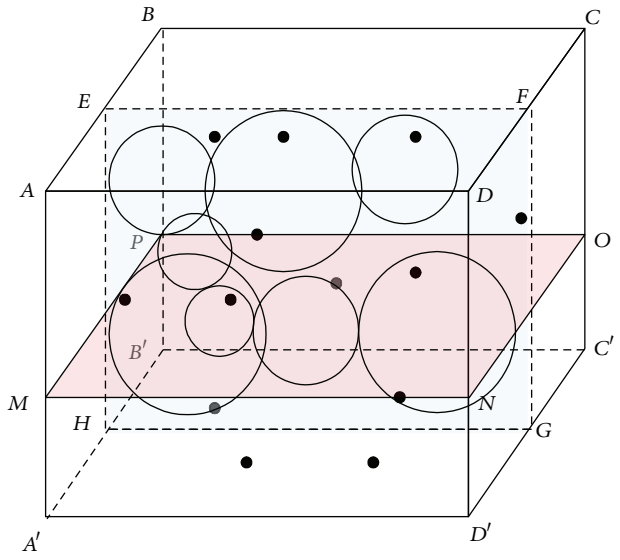


Figure 2: Coverage model in underwater 3D space.

3.2. Definitions

Definition 1 (binary sensor model). Suppose that the distance between sensor node X_i and space spot Y is $D(X_i, Y)$ and that each sensor node has a sensing range R. The sensing probability is shown as follows:

$$p(X_i, Y) = \begin{cases} 1 & D(X_i, Y) < R, \\ 0 & \text{else.} \end{cases} \quad (1)$$

Definition 2 (sensing sphere). Sensor node X_i is positioned at (x_i, y_i, z_i), and the sensor model is the binary sensor model with radius R. Thus, the coverage space of the sensor node underwater is a sphere with a center located at (x_i, y_i, z_i). The radius is denoted by R.

Definition 3 (sensing circle). The distance of underwater sensor node X_i to the sample plane H is d_i ($d_i < R$). When the sensing sphere of X_i and sample plane H intersect, a circle in the plane will be formed. The circle center is the projection of X_i to H, and the radius of the circle is $\sqrt{R^2 - d_i^2}$.

Definition 4 (sample plane). An arbitrary plane in 3D space intersect the boundary of 3D monitoring space to form polygon MNOP, as shown in Figure 2. The plane area which is contained by polygon MNOP is defined as sample plane.

Definition 5 (sample line). Choose a line at random in sample plane and denote it as "c", line c, and the boundary of sample plane intersect at points A and B, as illustrated in Figure 3. Line AB is defined as sample line.

Definition 6 (coverage rate). Coverage rate η is determined by the ratio between the volume of coverage space P and the volume of deployed space S, which is given by the following equation:

$$\eta = \frac{V_P}{V_S}. \quad (2)$$

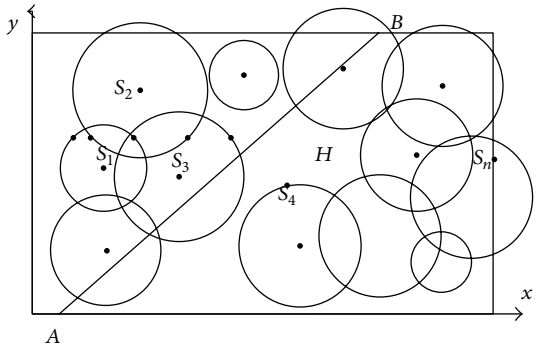


Figure 3: Sampling line.

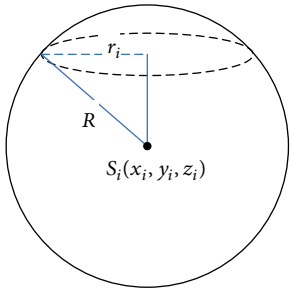


Figure 4: Sensing circle.

3.3. Problem Definition. The proposed algorithm attempts to maximize the coverage of USNs while striving to minimize the movement distance of the nodes. This problem can be defined as follows: given that n sensors are randomly and uniformly deployed on the surface of the ocean/sea, we devise an algorithm that maximizes the total 3D coverage of the nodes and minimizes the message and travel distance complexity. As mentioned in the Introduction, we seek a fully distributed algorithm that will enable the sensor nodes to self-organize before data collection commences. Our proposed algorithm has five phases: (1) sample plane in a target 3D space, (2) sample line in the sample plane, (3) establishment of a quadratic programming mathematical model to optimize line segment coverage, (4) depth assignment, and (5) additional rounds.

4. Coverage Optimization Algorithm Based on Sampling

The sample planes intersect with the sensing spheres. The distance of the center of the sensing spheres to the sample planes is less than R. The intersection of the sensing spheres and sample planes are a set of sensing circles with different radii (Figure 4). We analyze the influence of the position of coverage circles on the coverage rate of sample planes in the x-y and x-z directions. Although sensor nodes move in the z-axis, the radii of the sensing circles will change in the horizontal plane and locations will remain unchanged, respectively. In the vertical plane, the sensing circles also move in the z-axis; however, the radii of these sensing circles contain an original value. A coverage optimization algorithm is executed to change the radii and locations of the

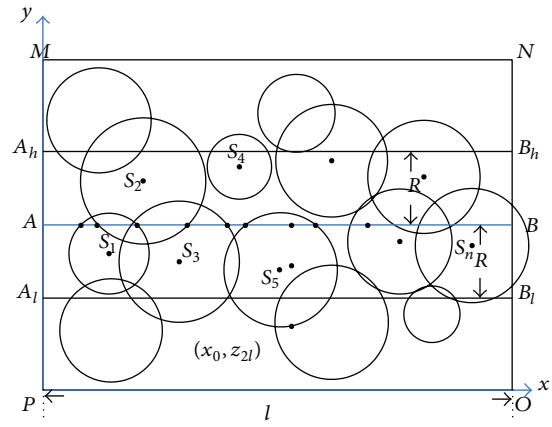


Figure 5: Sensing circles in a horizontal sample plane.

sensing circles in the sample planes. Thereafter, the sensor nodes corresponding to the sensing circles will be redeployed according to the new radii and locations of the sensing circles.

4.1. Coverage Optimization Analysis Based on the Horizontal Sample Plane. $MNOP$ is a horizontal sample plane in 3D space and has a plane function of $z = z_0$ (Figure 4). The sample plane interacts with the coverage sphere to form a coverage circle (Figure 5). The radius of the sample plane is expressed as follows:

$$r_i = \sqrt{R^2 - (z_i - z_0)^2}. \tag{3}$$

AB is a random straight line in plane $MNOP$. The distance of lines A_lB_l and A_hB_h to line AB is R. Suppose that n sensing circles exists $(S_1, S_2, \ldots, S_n)$ and that the centers of these sensing circles are located between lines A_lB_l and A_hB_h. Each of these sensing circles may intersect with line AB; for example, S_i and line AB may intersect at points (x_{il}, y_0) and (x_{ih}, y_0). We describe the sensing line segment of line AB and S_i as a set; that is, $C_i = \{x/x_{il} \le x \le x_{ih}\}$. C_i will be an empty set if S_i and line AB do not intersect. The union of these sets should include line segment set $C = \{x/0 \le x \le l\}$, and straight line AB is completely covered by the sensing circles

$$\bigcup_{i=1}^{n} C_i \supset C. \tag{4}$$

The radii of the sensing circles increase as the underwater sensor node moves closer to plane $EFGH$ in the z-axis. Thus, the sensing line segment is longer than the previous sensing line segment. The radius of the sensing circle corresponding to the underwater sensor node increases with the movement of the underwater sensor node toward the plane.

Suppose that the center of the ith sensing circle is at (x_i, y_i) and that the radius is r_i. For $r_i + \Delta r_i > |y_i - y_0|$, the sensing circle intersects line AB. The points of the intersection are shown as follows:

$$x_{il} = x_i - \sqrt{(r_i + \Delta r_i)^2 - y_i^2}, \tag{5}$$

$$x_{ih} = x_i + \sqrt{(r_i + \Delta r_i)^2 - y_i^2}. \tag{6}$$

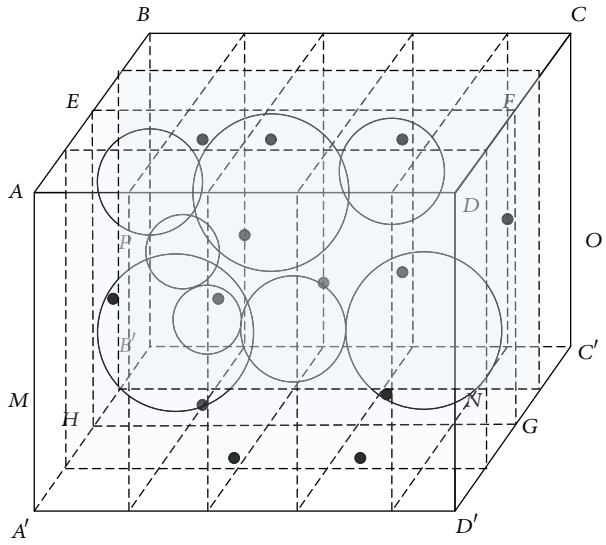


Figure 6: Vertical sample plane in underwater 3D space.

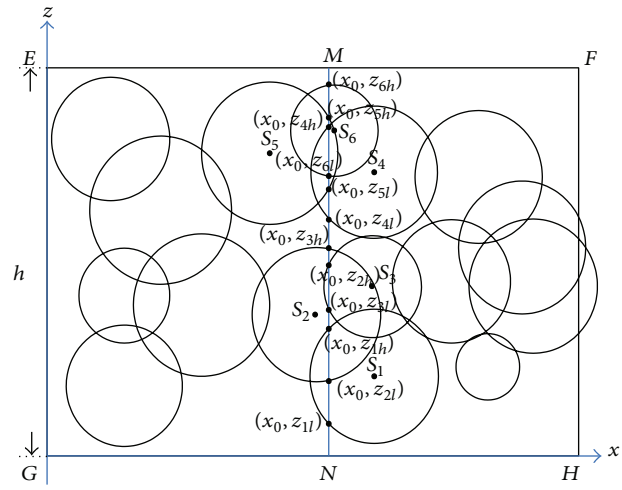


Figure 7: Vertical sample plane.

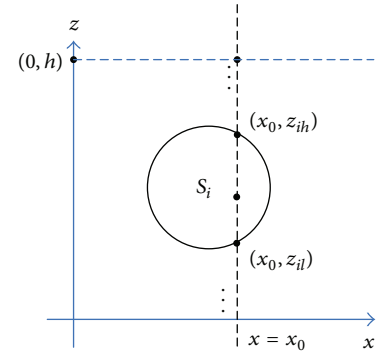


Figure 8: Upper and lower intersection points.

For $R_i + \Delta R_i < y_i$, the segment set corresponding to the sensing circle will be empty because the sensing circle and line AB do not intersect. This problem is hard to resolve because of two reasons: the constraints in (5) cannot be algebraically described because confirming the relationship among the sets $(C_1, C_2, \ldots, C_n, C)$ is difficult; this problem is a nonlinear programming problem because the upper and lower bounds of the sets and variables $(\Delta R_1, \Delta R_2, \ldots, \Delta R_n)$ are nonlinear, so the problem is hard to be resolved [21].

4.2. Coverage Optimization Analysis Based on the Vertical Sample Plane. In this section, we derive the optimization coverage in a vertical sample plane (Figures 6 and 7). $EFGH$ is a vertical sample plane, and the function of plane $EFGH$ is $y = y_0$. MN is a random straight line in plane $EFGH$ and is parallel to the z-axis. Line MN intersects n sensing circles $(S_1, S_2, \ldots, S_n)$, which correspond to n sensor nodes $(X_1, X_2, \ldots, X_n)$. The sensor nodes $(X_1, X_2, \ldots, X_n)$ are located at $(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)$, and the centers of the sensing circle are located at (x_1, z_1), $(x_2, z_2), \ldots, (x_n, z_n)$. Plane $EFGH$ has $z_1 \le z_2 \le \cdots \le z_n$. The radii of the sensing circles are expressed as follows:

$$r_i = \sqrt{R^2 - (y_0 - y_i)^2}. \tag{7}$$

Definition 7 (upper and lower intersections). Coverage circle S_i and sample line $(x = x_0)$ intersect at points (x_0, z_{il}) and (x_0, z_{ih}), where $z_{il} < z_{ih}$. Points (x_0, z_{il}) and (x_0, z_{ih}) are denoted as the upper lower intersections, respectively, (Figure 8).

The sensing circles and line segments intersect at (x_0, z_{1l}), $(x_0, z_{1h}), (x_0, z_{2l}), (x_0, z_{2h}), \ldots, (x_0, z_{nl}), (x_0, z_{nh})$. The intersection segments of the sensing circles and line MN are described as sets; that is, $Z_1 = \{z/z_{1l} \le z \le z_{1h}\}, Z_2 = \{z/z_{2l} \le z \le z_{2h}\}, \ldots, Z_n = \{z/z_{nl} \le z \le z_{nh}\}$. The union of these segment sets should include the segment set $Z = \{z/0 \le z \le h\}$ if line MN is completely covered by sensing circles

$$\bigcup_{i=1}^{n} Z_i \supset Z, \tag{8}$$

$$z_{il} = z_i - \sqrt{r_i^2 - (x_i - x_0)^2}, \tag{9}$$

$$z_{ih} = z_i + \sqrt{r_i^2 - (x_i - x_0)^2}. \tag{10}$$

The sensing line segments move with the corresponding underwater sensor nodes moving in z-axis, but the length of sensing line segment is unchanged. We suppose that the movement distance of n sensing circles is $\Delta d_1, \Delta d_2, \ldots, \Delta d_n$ separately. The sensing line segments after moving are described by sets, that is, $(z_{1l} + \Delta d_1, z_{1h} + \Delta d_1), (z_{2l} + \Delta d_2, z_{2h} + \Delta d_2), \ldots, (z_{nl} + \Delta d_n, z_{nh} + \Delta d_n)$. The optimization algorithm is developed to deal with the sample straight line coverage completely with the object minimum of the sum of sensing line segment moving distance. That is,

$$\begin{aligned} &\min \quad \sum_{i=1}^{n} \Delta d_i^2 \\ &\text{st:} \quad \bigcup_{i=1}^{n} (z_{il} + \Delta d_i, z_{ih} + \Delta d_i) \supset (0, h). \end{aligned} \tag{11}$$

Constraint conditions are shown as

$$\begin{aligned}
&\min\left(z_{1l}+\Delta d_1, z_{2l}+\Delta d_2, \ldots, z_{nl}+\Delta d_n\right) < 0,\\
&z_{2l}+\Delta d_2 > z_{1h}+\Delta d_1,\\
&z_{3l}+\Delta d_3 > \max\left(z_{1h}+\Delta d_1, z_{2h}+\Delta d_2\right),\\
&\qquad\vdots\\
&z_{nl}+\Delta d_n > \max\left(z_{1h}+\Delta d_1, z_{2h}+\Delta d_2, \ldots, z_{(n-1)h}+\Delta d_{n-1}\right),\\
&\max\left(z_{1h}+\Delta d_1, z_{2h}+\Delta d_2, \ldots, z_{nh}+\Delta d_n\right) > h.
\end{aligned} \tag{12}$$

It is difficult to find the optimal solution because the previous problem is nonlinear. Therefore we seek the feasible solution by tightening constraints. First, we analyze the affection of the sensing circles' position on optimization algorithm.

Theorem 8. *The sum of the traveled distance of the nodes is small when the order of the nodes remains the same compared with other nodes.*

Proof. We assume that only two sensing circles S_1, S_2 with centers located at (x_1, z_1) and (x_2, z_2) exist in interval $(0, l)$ (Figure 9). Suppose that $z_1 < z_2$. These two sensing circles and the z-axis intersect at points $(0, z_{1l})$, $(0, z_{1h})$ and $(0, z_{2l})$, $(0, z_{2h})$. The spot coordinates are expressed as follows:

$$\begin{aligned}
&z_{1h} - z_{1l} + z_{2h} - z_{2l} > l,\\
&0 \le z_{1l}, z_{1h}, z_{2l}, z_{2h} \le l.
\end{aligned} \tag{13}$$

Both Δd_1 and Δd_2 are the moving distance of sensing circles S_1 and S_2, respectively. We assume that both sensing circles will be arranged in the same order along the z-axis after optimization coverage

$$z_1 + \Delta d_1 < z_2 + \Delta d_2. \tag{14}$$

The complete coverage of line segment $(0, l)$ implies the following:

$$\begin{aligned}
&z_{2h} + \Delta d_2 \ge l,\\
&z_{1l} + \Delta d_1 \le 0.
\end{aligned} \tag{15}$$

We then derive the following equation:

$$\left|\Delta d_1\right| + \left|\Delta d_2\right| \ge l - z_{2h} + z_{1l}. \tag{16}$$

The minimum moving distance of the sensing circles is computed as follows:

$$\Delta d_{\min} = l - z_{2h} + z_{1l}. \tag{17}$$

We obtain $z_1 + \Delta d_1 > z_2 + \Delta d_2$, and the order of the sensing circles changes. The complete coverage of line segment $(0, l)$ implies the following:

$$\begin{aligned}
&z_{1h} + \Delta d_1' \ge l,\\
&z_{2l} + \Delta d_2' \le 0.
\end{aligned} \tag{18}$$

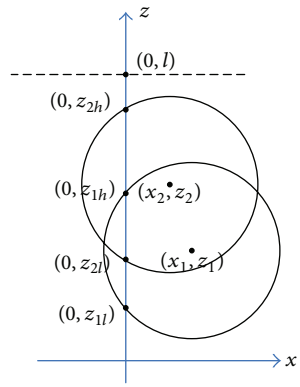


Figure 9: Position of two sensing circles.

Equations (18) denote the following:

$$\left|\Delta d_1'\right| + \left|\Delta d_2'\right| \ge l - z_{1h} + z_{2l}. \tag{19}$$

Thus, the minimum moving distance of the sensing circles is computed as follows:

$$\Delta d'_{\min} = l - z_{1h} + z_{2l}; \tag{20}$$

that is,

$$\Delta d'_{\min} - \Delta d_{\min} = z_{2h} + z_{2l} - z_{1h} - z_{1l}. \tag{21}$$

According to the nature of the circle, we derive the following equation:

$$z_2 = \frac{z_{2l} + z_{2h}}{2} \ge z_1 = \frac{z_{1l} + z_{1h}}{2}. \tag{22}$$

Thus, $z_{2l} + z_{2h} - z_{1l} - z_{1h} \ge 0$; that is, $\Delta d'_{\min} > \Delta d_{\min}$. We conclude that the sum of the moving distance of sensing circles in the same order is less than the sum of the moving distance of sensing circles in different orders. Thus, n sensing circles intersect at the sampling line and can be decomposed into multicoupled combinations of mobile nodes. After executing the coverage optimization algorithm, each pair of sensing circles will maintain their order. Thus, the order of all nodes will not change. □

On the basis of the previous analysis, we assume that the relative order of the sensing circles after optimization is the same before optimization. We propose different constraint conditions depending on the length of the sensing line segments. The length of the sensing line segment is $z_{ih} - z_{il}$, and the sensing circle S_i and sample line intersect at (x_0, z_{il}) and (x_0, z_{ih}). If n sensing circles intersect at the sample straight line, the sum of the length of the sensing line segments can be written as follows:

$$L = \sum_{i=1}^{n} \left(z_{ih} - z_{il}\right). \tag{23}$$

For $L \ge h$, the sample can be completely covered by sensing circles. Therefore, the intersection coordinates of the sensing circle S_i and straight line imply the following inequalities:

$$\begin{aligned}
&z_{il} + \Delta d_i \ge z_{(i-1)h} + \Delta d_{i-1},\\
&z_{ih} + \Delta d_h \le z_{(i+1)l} + \Delta d_{i+1}.
\end{aligned} \tag{24}$$

The upper intersection of circle S_i should be the lower side of the lower intersection of sensing circle S_{i-1}. The lower intersection of circle S_i should be the upper side of the upper intersection of sensing circle S_{i+1}. The constraint conditions of the optimization algorithm follow the following equation:

$$\begin{aligned} \min \ & \sum_{i=1}^{n} \Delta d_i^2 \\ & z_{1l} + \Delta d_1 \le 0, \\ & z_{2l} + \Delta d_2 \ge z_{1h} + \Delta d_1, \\ & z_{3l} + \Delta d_3 \ge z_{2h} + \Delta d_2, \\ \text{st:} \ & \vdots \\ & z_{nl} + \Delta d_n \ge z_{(n-1)h} + \Delta d_{n-1}, \\ & z_{nh} + \Delta d_n \ge h. \end{aligned} \tag{25}$$

For $L < h$, the sample cannot be considered a solution to the optimization problem because the sample straight line cannot be fully covered by sensing circles. Therefore, the maximum coverage of the sample straight line is expected. The sample straight line is covered maximally when the sensing line segments in the sample straight line are disjointed. We establish the new optimization algorithm as follows:

$$\begin{aligned} \min \ & \sum_{i=1}^{n} \Delta d_i^2 \\ & z_{1l} + \Delta d_1 \ge 0, \\ & z_{1h} + \Delta d_1 \le z_{2l} + \Delta d_2, \\ & z_{2h} + \Delta d_2 \le z_{3l} + \Delta d_3, \\ \text{st:} \ & \vdots \\ & z_{(n-1)h} + \Delta d_{n-1} \le z_{nl} + \Delta d_n, \\ & z_{nh} + \Delta d_{n-1} \le h. \end{aligned} \tag{26}$$

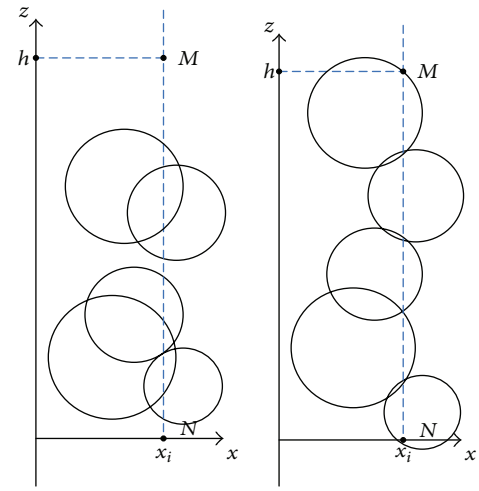
The new positions of the sensing circles are $(x_1, z_1 + \Delta d_1)$, $(x_2, z_2 + \Delta d_2), \ldots, (x_n, z_n + \Delta d_n)$, and the corresponding sensor nodes are located at $(x_1, y_1, z_1 + \Delta d_1), (x_2, y_2, z_2 + \Delta d_2), \ldots, (x_n, y_n, z_n + \Delta d_n)$. Figures 10(a) and 10(b) illustrate the position of the sensing circles after random placement and the final sensor positions determined by the COS algorithm, respectively. Figures 10(a) and 10(b) illustrate an example of $L \ge h$, whereas Figures 11(a) and 11(b) illustrate an example of $L < h$.

We find a feasible solution after limiting the constraints of the optimization algorithm in vertical planes. Moreover, the 2D plane coverage optimization in the y-z direction is similar to the x-z direction. Thus, we only sample the vertical plane in underwater 3D space (Figure 6).

4.3. Complete Algorithm. The pseudocode of the COS algorithm and the single sample plane coverage function are presented in Algorithms 1 and 2, and the flowchart of the single sample line coverage function is presented in Figure 12. The following parameters are defined for the mathematical modeling process:

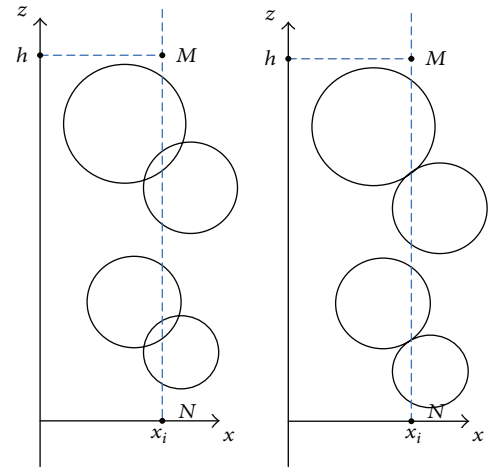
Sensor_set: the set of sensor node position, S_i $(x_i, y_i, z_i) \in$ *Sensor_set*,

circle_set: the set of sensing circle position and radius,



(a) Before optimization (b) After optimization

Figure 10: $L \ge W$ example.



(a) Before optimization (b) After optimization

Figure 11: $L < W$ example.

circle_N_set: the set of sensing circle position and radius after optimization,

d_step_plane: the step size of plane sampling,

d_step_line: the step size of line sampling,

(l, w, h): the length, width, and height of the 3D space, respectively,

x_int, *y_int*: the initial position of the sample line in the x and y directions, respectively,

x_line, *y_line*: the position of the sample line in the x and y directions, respectively,

R: the sensing radius of the sensor node.

The COS algorithm is performed by the sink node on land. First, a plane that is parallel to the x-z plane is sampled in interval $(0, R/2)$ (Line 1). In Lines 3 to 6 of

COS algorithm

%Input: *Sensor_set*, *d_step_plane*, *d_step_line*, l, w, h, *x_int*, *y_int*, R
%Output: *Sensor_set*.
Main procedure:
(1) y = random$(0, R/2)$; % the initial position of sample plane in $x - z$.
(2) While $y < l$
(3) For $S_i(x_i, y_i, z_i) \in$ *Sensor_set*
(4) If (the sensing sphere and sample plane intersect)
(5) calculate the radius of the sensing circle according to (7), and save the position (x_i, z_i) and radius r_i of sensing circle to *circle_set*.
(6) End
(7) end
(8) *circle_N_set* = plane_cover(*circle_set*, *x_int*, w, h, *d_step_line*);
% the single sample plane coverage optimization function and its procedure are shown in Figure 11.
(9) Update the position set *Sensor_set* according to *circle_N_set*.
(10) $y = y +$ *d_step_plane*
(11) Empty the set *circle_set*, *circle_N_set*;
(12) End
(13) x = random$(0, R/2)$; % initial position of the sample plane in $y - z$.
(14) While $x < w$
(15) For $S_i \in$ *Sensor_set*
(16) If (the sensing sphere and sample plane intersect)
(17) calculate the radius of the sensing circle according to (7), and save the position (x_i, z_i) and radius r_i of the sensing circle to *circle_set*.
(18) End
(19) end
(20) *circle_N_set* = plane_cover(*circle_set*,*y_int*, l, h, *d_step_line*);
(21) Update the position set *Sensor_set* according to *circle_N_set*;
(22) $x = x +$ *d_step_plane*;
(23) Empty the set *circle_set*, *circle_N_set*;
(24) End

Algorithm 1: Pseudocodes for the COS algorithm.

Plane coverage

function *circle_N_set* = plane_cover(*circle_set*, *x_int*, w, h, *d_step_line*)
(1) *x_line* = *x_int*;
(2) while *x_line* < w
(3) *circle_N_set* = line_cover(*circle_set*, *x_line*, h);
% the sample line coverage optimization function and its flowchart is shown in Figure 12;
(4) *x_line* = *x_line* + *d*;
(5) *circle_set* = *circle_N_set*;
(6) end

Algorithm 2: Pseudocodes for plane coverage.

the COS algorithm, the positions and radii of the sensing circles in the sampling plane are calculated if the sensing spheres intersect at the sampling plane. The plane coverage subroutine is invoked in Line 8 to optimize the sampling plane coverage. Thereafter, the sink node sends DEPTH messages to the underwater sensor nodes. Any node receiving a DEPTH message performs its specified movement (Line 9). After the first optimization, additional planes are sampled according to the step length until the sampling plane is beyond the boundary of 3D space, and the optimization algorithm is conducted numerously to optimize all sampling planes coverage. Additional rounds are also conducted when the sampling planes are parallel to the y-z plane to increase the total coverage in Lines 14–24.

Given the sampling plane function, the position of the initial sampling line, boundary of the plane, step length of the sampling, and positions and radii of the sensing circles, the subprogress can be used to optimize the coverage of the sampling plane. Line 1 is the line function of the initial sampling line. Line 3 performs the line coverage optimization function and calculates the new positions of the sensing circles. Additional optimization sequences are conducted until the sampling line is beyond the boundary of the sampling plane.

We describe the single-line coverage optimization function in a flowchart (Figure 12). The following equation is the expression of single-line coverage optimization function:

$$\text{function circle\_}N\text{\_set} = \text{line\_cover}\,(\text{circle\_set}, x\text{\_line}, h)\,. \tag{27}$$

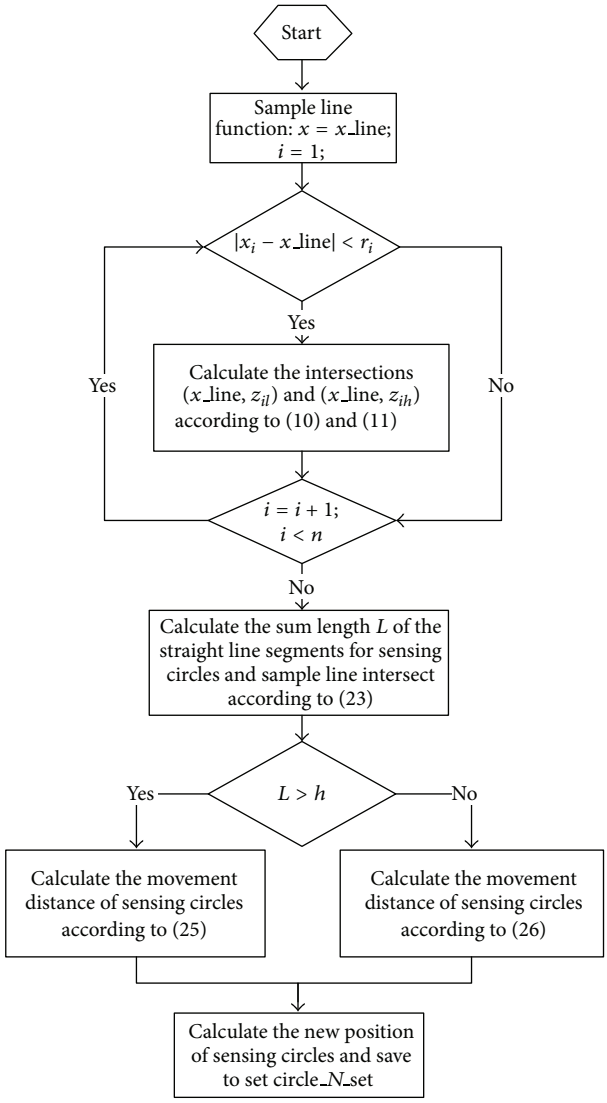


Figure 12: Straight line coverage flowchart.

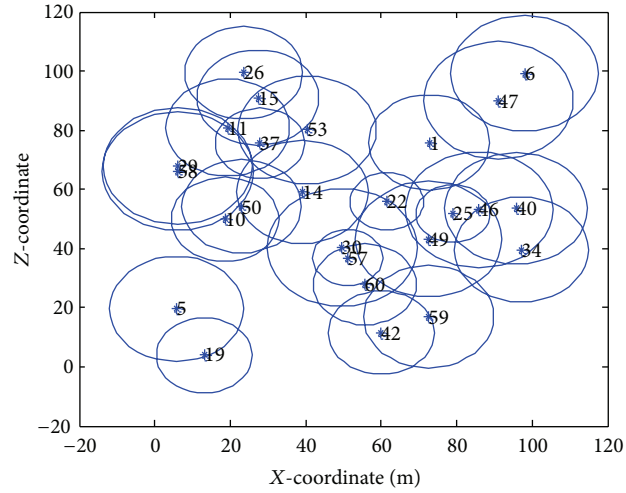


Figure 13: Initial position of the sensing circles after random distribution.

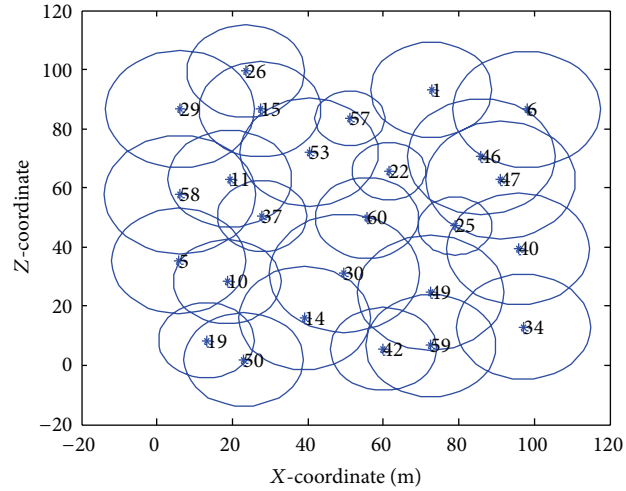


Figure 14: Position of the sensing circles after execution of the COS algorithm.

5. Simulation

In this section, we describe the simulation setup, performance metrics, and performance results of the study. We assume that all messages can be transmitted/received without any errors and that the sensor nodes are uniformly distributed in a 100 m × 100 m × 100 m 3D space. The sensing radius of the underwater sensor node is 20 m, and the communication nodes that are deployed on the water surface can communicate with each other. A coordinator node is attached to the server and acts as a gateway to the USNs. The server implements the COS algorithm presented in Section 4 and conveys the new position of the underwater sensor nodes to the coordinator. The location of an underwater sensor node is adjusted by its corresponding buoy via the cable after receiving the information.

Figure 13 shows the initial position of the sensing circles in one of the vertical sample planes after random distribution. Figure 14 shows the final position of the sensing circles of the COS algorithm in the same vertical sample planes. We observe that the overlap area in Figure 14 is less than the overlap area in Figure 13.

Figure 15 shows the coverage rate comparison between the random placement and COS algorithm in the sample vertical planes. The simulation is conducted with 60 sensor nodes, and the sensing range is set to 20 m. The sampling plane coverage rate of the COS algorithm outperforms the random approach by approximately 10%. This improvement is insignificant and is caused by the limited movement direction of the nodes (i.e., vertical movement only).

We have conducted experiments with different node quantities. The results shown in Figure 16 clearly reveal that our algorithm is valid for various node quantities. The simulation is implemented twice, and the sensing range is set to 20 m. The COS algorithm outperforms the random approach by approximately 10% in terms of coverage. The performance of the COS algorithm is superior when the number of nodes is relatively few. Considering that the 3D

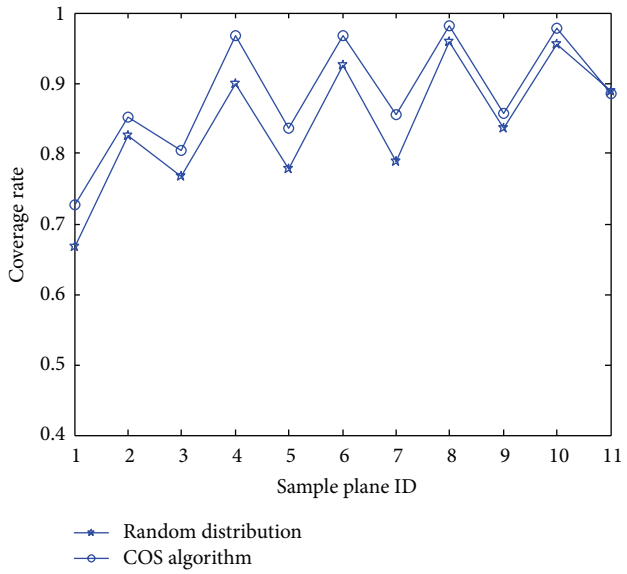


Figure 15: Coverage rate of sample planes for different distributions.

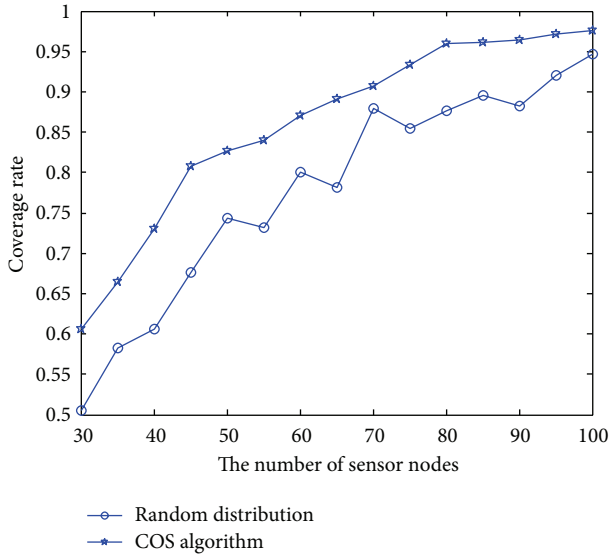


Figure 16: Coverage rate for the number of nodes.

space coverage and node deployment on the surface have a close relationship, the optimal coverage in 3D space will be achieved by the COS algorithm if the nodes on the surface are distributed well.

Figure 17 presents the simulation results in different sampling steps for 40 to 80 underwater sensor nodes. The algorithm performance is weak when additional iterations are implemented because the optimization in the second plane will have a negative effect on the first plane if two adjacent planes intersect at the same sensing sphere.

Similarly, when sensing range is increased while the number of nodes is fixed as 60 and 80, respectively, coverage increases as seen in Figures 18 and 19. The patterns show that the coverage can be effectively enhanced by the COS

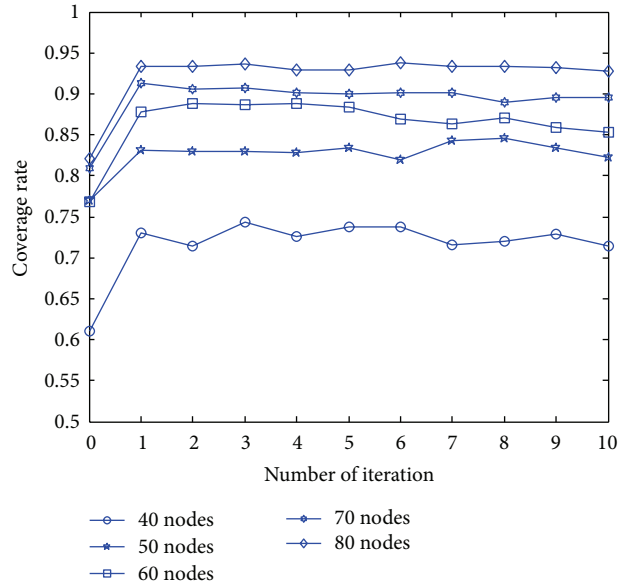


Figure 17: Coverage rate for iterations.

algorithm. We consider the influence of node quantity, iteration, and sensing range in the experiments and find that the proposed algorithm works effectively under extreme surroundings.

6. Conclusion

This study proposes a 3D COS to enhance the coverage of USNs. We convert a 3D space coverage into a 2D plane coverage for heterogeneous networks by a sampling plane in the target 3D space. The positions of the sensing circles in the sample plane are calculated by the COS algorithm to enhance coverage. Thereafter, the sensor nodes that correspond to the sensing circles are redeployed. The simulation results show that the COS algorithm can improve the coverage provided by an initial random placement. To a certain degree, the coverage in the 3D space improves when the iterations increase. However, the coverage will remain unchanged or decrease when the number of iterations surpasses a certain value for the adjacent sample planes. The performance of the COS algorithm is affected by two aspects:

(1) for the limitation of the network model, the coverage cannot be significantly improved when the wireless sensor nodes are nonuniformly distributed,

(2) the COS algorithm is performed on each sample plane alone because we do not consider the relativity between the adjacent sample planes.

In the future, we will consider the relativity of the sample planes to ensure the flexibility of the COS algorithm as well as adjust the positions of the wireless sensor nodes on the water surface to distribute uniformly the sensor nodes and enhance their coverage in 3D space.

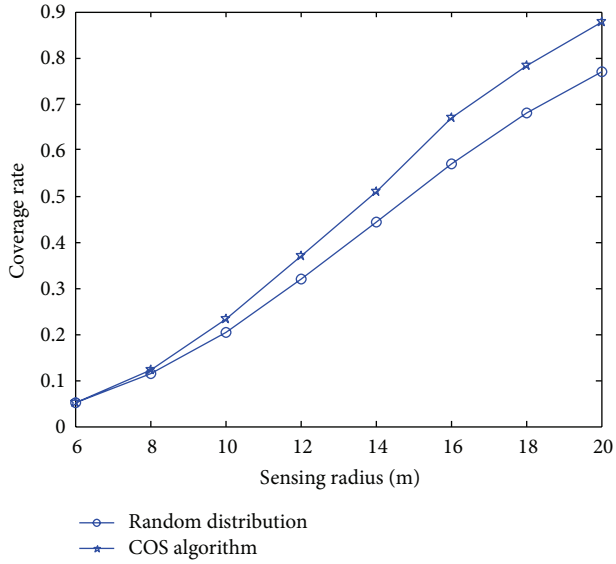


Figure 18: Coverage versus the sensing radius for 60 nodes.

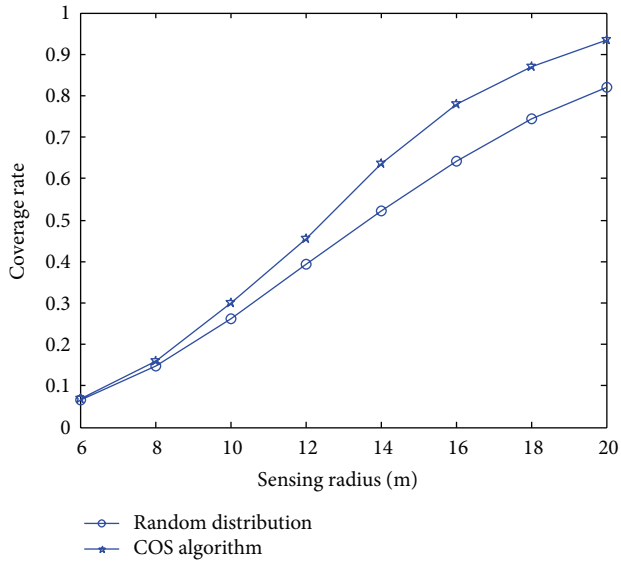


Figure 19: Coverage versus sensing radius for 80 nodes.

Acknowledgments

The subject is sponsored by the National Natural Science Foundation of China under Grant nos. 61373139, 61300239 and 61171053; the Doctoral Fund of Ministry of Education of China under Grant no. 20113223110002; Science & Technology Innovation Fund for higher education institutions of Jiangsu Province (CXZZ12_0481).

References

[1] H. M. Ammari and S. Das, "A study of k-coverage and measures of connectivity in 3D wireless sensor networks," *IEEE Transactions on Computers*, vol. 59, no. 2, pp. 243–257, 2010.

[2] C. Wen-Yu, L. Jing-Biao, and Z. Xue-Ting, "CDS-based coverage control algorithm for Buoys based sensor networks," in *Proceedings of the IEEE (OCEANS '10)*, pp. 1–5, Sydney, Australia, May 2010.

[3] A. Stefanov and M. Stojanovic, "Design and performance analysis of underwater acoustic networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 10, pp. 2012–2021, 2011.

[4] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[5] L. Peng, K. Liu, and G. Liu, "Research on wireless sensor networks fault-tolerant coverage algorithm base on particle swarm optimization," in *Proceedings of the IET International Conference on Wireless Sensor Network (IET-WSN '10)*, pp. 286–290, November 2010.

[6] Y. Gu, Y. Ji, J. Li, and B. Zhao, "QoS-aware target coverage in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 9, no. 12, pp. 1645–1659, 2009.

[7] M. Wang and H. Shen, *Probablility Theory and Mathematical Statistics*, Higher Education Press, Peking, China, 2007.

[8] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 1976–1984, March 2005.

[9] M. Marengoni, B. A. Draper, A. Hanson, and R. Sitaraman, "Placing observers to cover a polyhedral terrain in polynomial time," in *Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96)*, pp. 77–82, December 1996.

[10] C. Huang, Y. Tseng, and L. Lo, "The coverage problem in three-dimensional wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, pp. 3182–3186, December 2004.

[11] X. Bai, Z. Yun, D. Xuan, B. Chen, and W. Zhao, "Optimal multiple-coverage of sensor networks," in *Proceedings of the IEEE (INFOCOM '11)*, pp. 2498–2506, April 2011.

[12] H. M. Ammari, "On the problem of k-coverage in 3d wireless sensor networks: a Reuleaux tetrahedron-based approach," in *Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '11)*, pp. 389–394, December 2011.

[13] H. M. Ammari and S. K. Das, "Critical density for coverage and connectivity in three-dimensional wireless sensor networks using continuum percolation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 6, pp. 872–885, 2009.

[14] K. A. Crews Baumgartner, *Control and optimization of track coverage in underwater sensor networks [Ph.D. thesis]*, Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA, 2007.

[15] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 1293–1303, April 2003.

[16] J. Huang, H. Huang, P. Sun, R. Wang, and X. Wei, "Probability model based coverage-enhancing algorithm for WSNs of nodes' adjustable movement pattern," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 570643, 10 pages, 2012.

[17] J. Huang, L. Sun, R. Wang, and H. Huang, "Improved virtual potential field algorithm based on probability model in three-dimensional directional sensor networks," *International Journal*

of Distributed Sensor Networks, vol. 2012, Article ID 942080, 9 pages, 2012.

[18] X. Yu, W. Huang, J. Lan, and X. Qian, "A novel virtual force approach for node deployment in wireless sensor network," in *Proceedings of the IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS '12)*, pp. 359–363, May 2012.

[19] H. Ma, X. Zhang, and A. Ming, "A coverage-enhancing method for 3D directional sensor networks," in *Proceedings of the IEEE 28th Conference on Computer Communications (INFOCOM '09)*, pp. 2791–2795, April 2009.

[20] Q. Zhao and M. Gurusamy, "Lifetime maximization for connected target coverage in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 6, pp. 1378–1391, 2008.

[21] M. Naderan, M. Dehghan, and H. Pedram, "Sensing task assignment via sensor selection for maximum target coverage in WSNs," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 262–273, 2013.

[22] S. Sengupta, S. Das, M. D. Nasir, and B. K. Panigrahi, "Multi-objective node deployment in WSNs: in search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 405–416, 2013.

[23] H. Tezcan, E. Cayirci, and V. Coskun, "A distributed scheme for 3D space coverage in tactical underwater sensor networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM '04)*, pp. 697–703, November 2004.

[24] K. Akkaya and A. Newell, "Self-deployment of sensors for maximized coverage in underwater acoustic sensor networks," *Computer Communications*, vol. 32, no. 7–10, pp. 1233–1244, 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 798537, 5 pages
http://dx.doi.org/10.1155/2013/798537

Research Article

Sparse Signal Recovery by Stepwise Subspace Pursuit in Compressed Sensing

ZheTao Li,[1,2] JingXiong Xie,[1] DengBiao Tu,[3] and Young-June Choi[4]

[1] *The College of Information Engineering, Xiangtan University, Hunan 411105, China*
[2] *School of Computer, National University of Defense Technology, Hunan 410073, China*
[3] *National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China*
[4] *Department of Information and Computer Engineering, Ajou University, Suwon 443749, Republic of Korea*

Correspondence should be addressed to DengBiao Tu; tudengbiao@163.com

Received 26 May 2013; Accepted 10 July 2013

Academic Editor: Fu Xiao

In this paper, an algorithm named stepwise subspace pursuit (SSP) is proposed for sparse signal recovery. Unlike existing algorithms that select support set from candidate sets directly, our approach eliminates useless information from the candidate through threshold processing at first and then recovers the signal through the largest correlation coefficients. We demonstrate that SSP significantly outperforms conventional techniques in recovering sparse signals whose nonzero values have exponentially decaying magnitudes or distribution of $N(0, 1)$. Experimental results of Lena show that SSP is better than CoSaMP, OMP, and SP in terms of peak signal to noise ratio (PSNR) by 5.5 db, 4.1 db, and 4.2 db, respectively.

1. Introduction

In many applications such as statistical regression [1], digital communications [2], image processing [3, 4], multimedia sensor networks [5, 6], interpolation/extrapolation [7], and signal deconvolution [8, 9], recovering high-dimensional signals from relatively fewer measurements is a challenging task. Fortunately, in the real world many signals are, or can be, transformed (such as DCT, wavelet packet transform [10]) to sparse such that only a small part of signal coefficients are nonzero values. And compressed sensing [11, 12] allows us to recover sparse signal from high-dimensional signals with very few measurements. In fact, some works in the real world show that one can recover exactly K sparse signal of length M with only $O(K \log M)$ random measurements. Let $\mathbf{y} \in R^N$ be an observed signal, and let $\mathbf{\Phi} \in R^{N \times M}$ be a dictionary of atoms, then the standard of the formulation of sparse follows

$$\hat{\mathbf{x}} = \arg\min \|\mathbf{x}\|_0 \quad \text{s.t. } \mathbf{y} = \mathbf{\Phi x}, \tag{1}$$

where $\|\boldsymbol{\theta}\|_0$ is the $\boldsymbol{\theta}$'s ℓ_0-norm, which counts the number of nonzero elements of vector $\boldsymbol{\theta}$.

Finding the exact solution of (1) is known as NP-hard [7]. It is intractable for combinatorial approaches to solve the problems of moderate-to-high dimensionality, and thus one needs to resort to heuristic procedures. However, if the dictionary $\mathbf{\Phi}$ satisfies nearly orthogonality, (1) then becomes

$$\hat{\mathbf{x}} = \arg\min \|\theta\|_1 \quad \text{s.t. } \mathbf{y} = \mathbf{\Phi x} = \mathbf{\Phi\Psi}\boldsymbol{\theta} = \mathbf{\Theta}\boldsymbol{\theta}, \tag{2}$$

where $\|\boldsymbol{\theta}\|_1$ is the $\boldsymbol{\theta}$'s ℓ_1-norm, $\mathbf{\Theta} = \mathbf{\Phi\Psi}$, and there is a sparse vector such that $\mathbf{x} = \mathbf{\Psi}\boldsymbol{\theta}$.

The rest of the paper is organized as follows. Section 2 summarizes the related work of recover algorithm in compressed sensing. Section 3 contains the description of stepwise subspace pursuit algorithm. Section 4 makes a comparison of our work with the related papers through simulation. Section 5 presents our main conclusion.

2. Related Work

Existing recovery algorithms are roughly classified into three main families: convex relaxation algorithms, Bayesian algorithms, and pursuit algorithms. Our algorithm presented in this paper belongs to the pursuit family.

The convex relaxation algorithms approximate the nonsmooth and nonconvex ℓ_0-norm by functions that are easier to handle. The resulting problem can be solved by means of standard optimization techniques. Well-known instances of algorithms based on such an approach are Basis Pursuit (BP) [13] and FOCUSS [14] which approximate the ℓ_0-norm by the ℓ_1-norm and ℓ_p-norm ($p < 1$), respectively.

The Bayesian algorithms express the problem as the solution of a Bayesian inference problem and apply statistical tools to solve it, that is assuming a prior distribution for the unknown coefficients that favors sparsity. They develop a maximum a posteriori estimator that incorporates the observation. There are many algorithms that incorporate some of these features. For example, identify a region of significant posterior mass [15] or average over most probable models [16]. One key ingredient of Bayesian algorithms is the choice of a proper prior on the sought sparse vector.

The pursuit algorithms for sparse signal recovery are a greedy approach that iteratively refines the current estimate for the coefficient vector $\mathbf{x}$ by modifying one or several coefficients chosen to yield a substantial improvement in approximating the signal. The family of pursuit algorithms includes several approaches according to the way of updating the support set: single or multiple algorithms. The algorithms of single updating support set gradually increase the support by sequentially adding new atoms. The complexity of these algorithms is lower than the complexity of BP. However, they require more measurements M for accurate reconstruction, and they often have an effect on empirical work and do not offer the strong theoretical guarantees. Single algorithms include matching pursuit (MP) [17], orthogonal matching pursuit (OMP) [18]. However, for many applications, single updating support set does not offer adequate performance, so researchers have developed more sophisticated pursuit methods that work better in practice and yield essentially optimal theoretical guarantees, called multiple algorithms. These techniques depend on several enhancements to the basic greedy framework: (1) selecting multiple columns per iteration; (2) pruning the set of active columns at each step; (3) solving the least squares problems iteratively; (4) theoretical analysis using the RIP bound. There are many algorithms that incorporate some of these features. For example, stagewise orthogonal matching pursuit (StOMP) [19], hard thresholding pursuit (HTP) [20], regularized orthogonal matching pursuit algorithm (ROMP) [21] which was the first greedy technique whose analysis was supported by a RIP bound, compressive sampling matching pursuit (CoSaMP) [22] which was the first algorithm to assemble these ideas to obtain essentially optimal performance guarantees and the subspace pursuit (SP) [23], and so forth. Sparsity-predict in these algorithms bring to the perfect performance. However, once the sparsity is false predicted, many signals cannot be reconstructed accurately. All in all, pursuit algorithms have often been considered naive, in part because there are contrived examples where the approach fails spectacularly. However, recent research has clarified that greedy pursuits succeed empirically and theoretically in many situations where convex relaxation works.

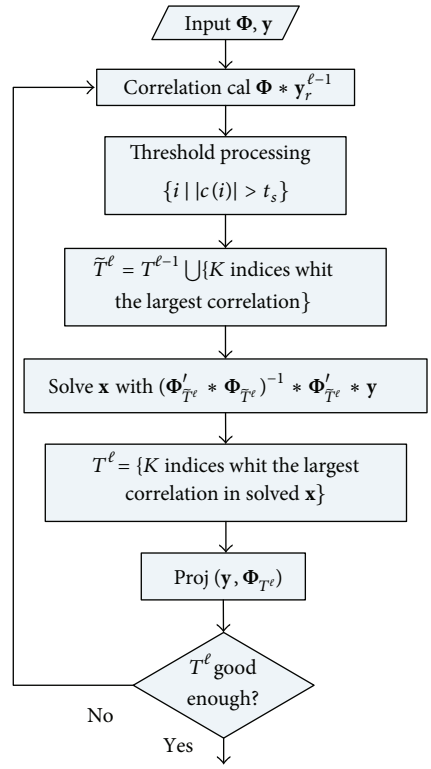


Figure 1: Description of reconstruction algorithms for K-sparse signals.

3. Sparse Signal Recovery by Stepwise Subspace Pursuit

Figure 1 depicts the schematic representation of the proposed SSP algorithm. The ℓth iteration applies matched filtering to the current residual and gets a candidate set $\mathbf{\Phi} * \mathbf{y}_r^{\ell-1}$, which contains a small number of significant nonzero values. Then we eliminate useless information in the candidate through threshold processing $\{i \mid |c(i)| > t_s\}$ and select K indices which are considered to be reliable on some iteration steps with the largest correlation by interim estimate. We merge the K indices of newly selected coordinates with the previous support estimate, thereby updating the intermediate estimate $\widetilde{T}^{\ell}$. We have the new approximation $\mathbf{x}$ supported in $\widetilde{T}^{\ell}$ with coefficients given by $(\mathbf{\Phi}'_{\widetilde{T}^{\ell}} * \mathbf{\Phi}_{\widetilde{T}^{\ell}})^{-1} * \mathbf{\Phi}'_{\widetilde{T}^{\ell}} * \mathbf{y}$. The updated support estimate can be gained from approximation $\mathbf{x}$ by the largest correlation. We then project the vector $\mathbf{y}$ on the columns of $\mathbf{\Phi}_{T^{\ell}}$ belonging to the updated support, and check the stopping condition, and if it is not yet time to stop, we set $\ell = \ell + 1$ and go to the next iteration.

The algorithm is depicted in Algorithm 1. The main contribution of the SSP reconstruction algorithm is that it generates a list of candidates sequentially and incorporates a simple method for re-evaluating the reliability of all candidates at iteration, thus gaining the correlation of candidates before the operation of SP with the correlation at iteration of the method.

Algorithm: stepwise sub space pursuit
Input:
An $N \times d$ measurement matrix $\mathbf{\Phi}$
An N dimensional data vector $\mathbf{y}$
The sparsity level K of the idea signal
Procedure:
Initialization:
(1) $T^0 = \{K \text{ indices with the largest correlation entries in the vector } \mathbf{\Phi} * \mathbf{y}\}$
(2) $\mathbf{y}^0 = \text{resid}\left(\mathbf{y}, \mathbf{\Phi}_{T_0}\right)$
Iteration: when sth the iteration, execute the following steps
(1) threshold processing y^{s-1}
(2) $\widetilde{T}^s = T^{s-1} \cup \{K \text{ indices with largest correlation in } \mathbf{\Phi}_{T^{s-1}} * \mathbf{y}^{s-1}\}$
(3) slove $\mathbf{x}$ with $\mathbf{x} = \left(\mathbf{\Phi}'_{\widetilde{T}^s} * \mathbf{\Phi}_{\widetilde{T}^s}\right)^{-1} * \mathbf{\Phi}'_{\widetilde{T}^s} * \mathbf{y}$
(4) $T^s = \{K \text{ indices with largest correlation in solved } \mathbf{x}\}$
(5) $\mathbf{y}^s = \text{resid}\left(\mathbf{y}, \mathbf{\Phi}_{T^s}\right)$
(6) if $\left\|\mathbf{y}^s\right\| > \left\|\mathbf{y}^{s-1}\right\|$ then break the iteration, else go next iteration
Output:
The approximate solve $\mathbf{x}$

Algorithm 1: Algorithm of SSP.

4. Simulation and Results

In this section, we show the performance of the proposed algorithm through simulation from two aspects: (1) for K-sparse 1-dimensional signal, we compare the reconstruction probabilities of OMP, CoSaMP, SP, and SSP algorithmies; (2) for sparse 2-dimensional images signal with DCT, we compare the effectiveness and accuracy of signal recovery with OMP, CoSaMP, SP, and SSP algorithms under the same test conditions.

In Figure 2, we compare the performance of SSP algorithm with that of OMP, CoSaMP, and SP algorithms. The original signal in Figure 2(a) is obtained, when the K-sparse signal is set by random nonzero values drawn from $N(0, 1)$, where the K nonzero coefficients are set by iid $N(0, 1)$ and the remaining coefficients of $\mathbf{x}$ are set by 0. The original signal in Figure 2(b) is obtained, when the K nonzero coefficients are a random permutation of K exponentially decay and the remaining coefficients of $\mathbf{x}$ are set to 0.

Figure 2(a) shows that SSP performs better than OMP, CoSaMP, and SP when the nonzero entries of the sparse signal are drawn according to zero-mean Gaussian with variance 1. We discover that the recovery probability is 1 in low sparsity level, but the recovery signal is not accurate when the sparsity increases to a certain level. And therefore more measurements are needed for a better signal recovery. Furthermore the results in Figure 2(a) show that the CoSaMP, OMP, and SP can accurately recover signal in sparsity level in 15, 17, and 18, respectively, while the SSP algorithm can reach 21. As depicted in Figure 2(b), SSP significantly outperforms existing methods for the exponential case.

Two 256×256 tested images (Lena and Cameraman) are used to illustrate the quality of reconstructed image. Our simulation experiments were performed in the MATLAB2010b environment using an AMD Athlon II X2 245 processor with 2 GB of memory. The Gaussian random matrix was applied to measure the coefficients of OMP, CoSaM, PSP, and SSP algorithms. In order to validate the effectiveness of OMP, CoSaMP, SP, and SSP algorithms, compression ratio of test images was set 0.3333. Figure 3 shows that the reconstructed quality of SSP is better than that of the OMP, CoSaMP, and SP in the same experimental condition. Table 1 is the PSNR of reconstructed images and the reconstructed times of OMP, CoSaMP, SP, and SSP algorithms for test images.

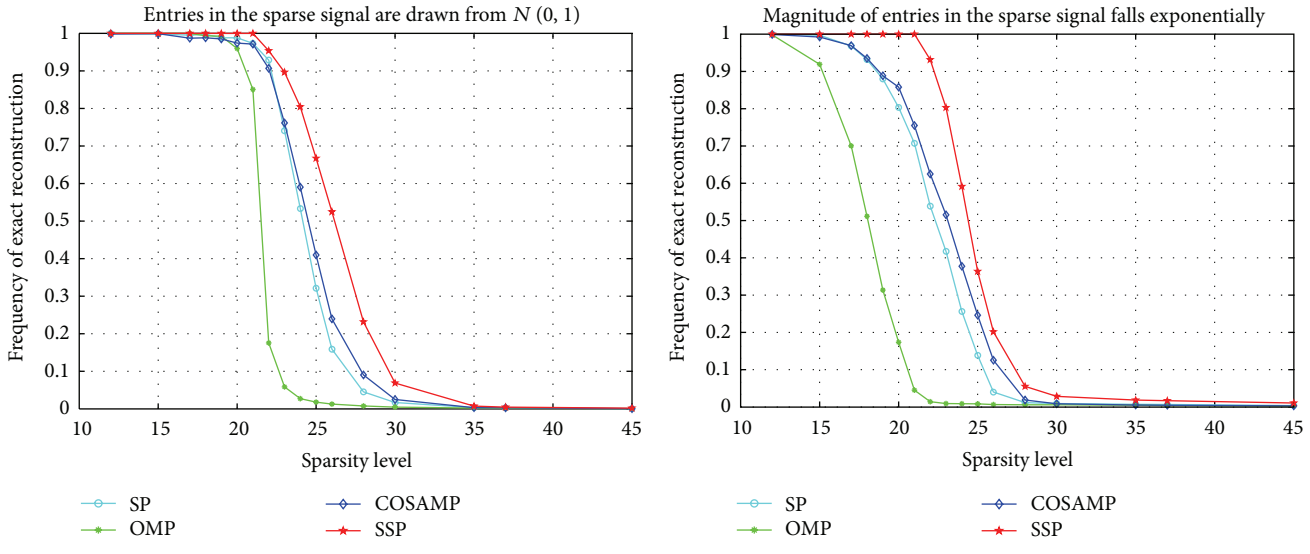
Table 1: PSNR and reconstructed time of different algorithms for images.

	CoSaMP	OMP	SP	SSP
Lena				
PSNR	19.8831	21.3572	21.2388	25.4261
TIME	3.9	0.5	2.4	3.9
Cameraman				
PSNR	16.3795	18.1784	17.3851	21.1282
TIME	4.4	0.5	2.5	4.3

As shown in Table 1, SSP has the maximum PSNR and the largest time consumed in reconstruction compared with other algorithms. It shows that searching for maximum correlation set from candidate in SSP is a double-edged sword. The maximum correlation of SSP explains the possibility of its relative higher quality compared to OMP, CoSaMP, and SP in this example.

5. Conclusion

In this paper, a stepwise subspace pursuit algorithm for signal reconstruction is proposed by using the largest correlation of the candidate set. It can obtain accurate solutions that preserve more important coefficients as well as recover more data than other existing algorithms. The experimental results



(a) Nonzero entries of $\mathbf{x}$ are distributed $N(0, 1)$

(b) Nonzero entries of $\mathbf{x}$ are decay exponentially

FIGURE 2: The probability of 256-length and K-sparse signal is recovered from 128 random projections exactly by different reconstruction algorithms.

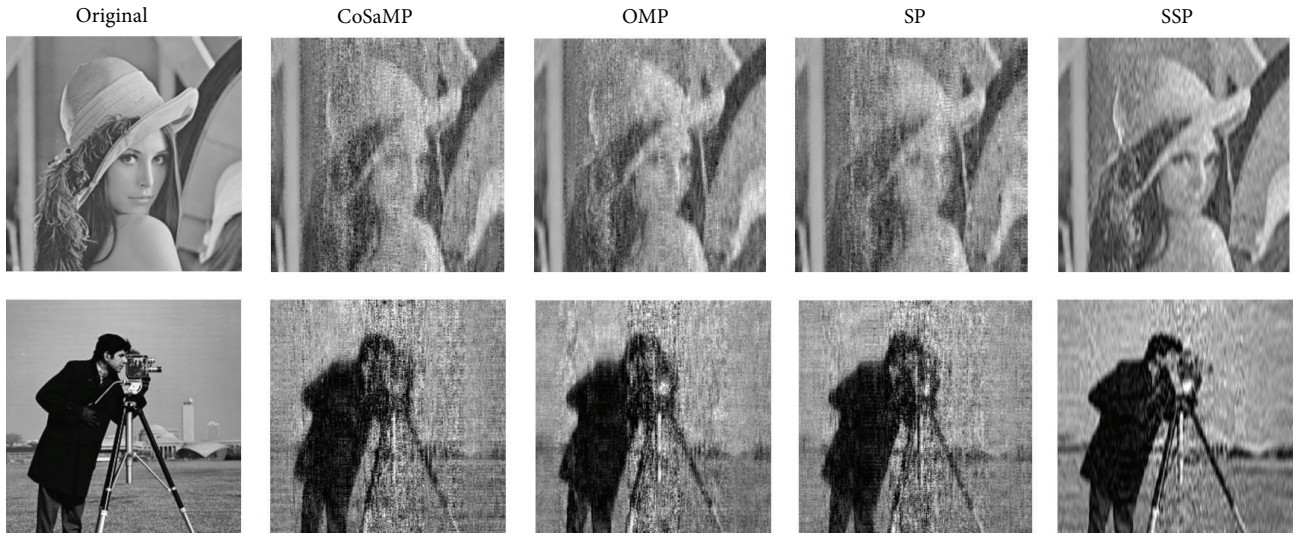


FIGURE 3: Performance of different algorithms for two images.

of Lena demonstrated that SSP is a more effective algorithm for signal recovery from random measurement than OMP, CoSaMP, and SP algorithms in peak signal to noise ratio by 5.5 db, 4.1 db, and 4.2 db, respectively. In future work, we need to investigate how to reduce the reconstruction time while improving the quality of signal.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants no. 61100215, no. 612111198, and no. 61070180, Hunan Provincial Natural Science Foundation of China with Grant no. 12JJ9021, Science and Technology Planning Project of Hunan Provincial Science & Technology Department with Grant no. 2011GK3200, Natural Science Foundation for Doctor, Xiangtan University with Grant no. 10QDZ30, and the construct program of the key discipline in Hunan province.

References

[1] A. Miller, *Subset Selection in Regression*, Chapman & Hall/CRC, New York, NY, USA, 2nd edition, 2002.

[2] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[3] B. D. Jeffs and M. Gunsay, "Restoration of blurred star field images by maximally sparse optimization," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 202–211, 1993.

[4] J. Bobin, J. L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho, "Morphological component analysis: an adaptive thresholding strategy," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2675–2681, 2007.

[5] F. Xiao, J. K. Liu, J. Guo, and L. F. Liu, "Novel side information generation algorithm of multi-view distributed video coding for multimedia sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 582403, 7 pages, 2012.

[6] F. Xiao, J. Wang, L. J. Sun, R. H. Wang, and J. Guo, "Coverage enhancement strategy based on novel perception and co-evolution for multimedia sensor networks," *Chinese Journal of Electronics*, vol. 22, no. 1, pp. 135–140, 2013.

[7] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.

[8] J. J. Kormylo and J. M. Mendel, "Maximum likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Transactions on Information Theory*, vol. IT-28, no. 3, pp. 482–488, 1982.

[9] C. Soussen, J. Idier, D. Brie, and J. Duan, "From Bernoulli-Gaussian deconvolution to sparse signal restoration," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4572–4584, 2011.

[10] Z. T. Li, J. X. Xie, Z. G. Yu, and Y. J. Choi, "Compressed sensing based on best wavelet packet basis for image processing," *Journal of Computers*, vol. 8, no. 8, pp. 1947–1950, 2013.

[11] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[12] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[13] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[14] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[15] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[16] P. Schniter, L. C. Potter, and J. Ziniel, "Fast bayesian matching pursuit," in *Proceedings of the Information Theory and Applications Workshop*, pp. 326–332, San Diego, Calif, USA, February 2008.

[17] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[18] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[19] D. L. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[20] M. A. Efroymson, "Multiple Analysis Regression," in *Mathematical Methods for Digital Computers*, vol. 1, pp. 191–203, 1960.

[21] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317–334, 2009.

[22] P. M. T. Broersen, "Subset regression with stepwise directed search," *Journal of the Royal Statistical Society C*, vol. 35, no. 2, pp. 168–177, 1986.

[23] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.