

Research Article

From Smart Camera to SmartHub: Embracing Cloud for Video Surveillance

Mukesh Kumar Saini,¹ Pradeep K. Atrey,² and Abdulmotaleb El Saddik¹

¹ *Multimedia Communications Research Laboratory (MCRLab), School of Electrical Engineering and Computer Science (EECS), University of Ottawa, Ottawa, ON, Canada K1N 6N5*

² *Department of Applied Computer Science, The University of Winnipeg, Winnipeg, MB, Canada R3B 2E9*

Correspondence should be addressed to Mukesh Kumar Saini; msain2@uottawa.ca

Received 22 November 2013; Accepted 20 February 2014; Published 1 April 2014

Academic Editor: Mohammad Mehedi Hassan

Copyright © 2014 Mukesh Kumar Saini et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Smart cameras were conceived to provide scalable solutions to automatic video analysis applications, such as surveillance and monitoring. Since then, many algorithms and system architectures have been proposed, which use smart cameras to distribute functionality and save bandwidth. Still, smart cameras are rarely used in commercial systems and real installations. In this paper, we investigate the reason behind the scarce commercial usage of smart cameras. We found that, in order to achieve scalability, smart cameras put additional constraints on the quality of input data to the vision algorithms, making it an unfavourable choice for future multicamera systems. We recognized that these constraints can be relaxed by following a cloud based hub architecture and propose a cloud entity, SmartHub, which provides a scalable solution with reduced constraints on the quality. A framework is proposed for designing SmartHub system for a given camera placement. Experiments show the efficacy of SmartHub based systems in multicamera scenarios.

1. Introduction

The basic purpose of smart cameras is to cope with the ever increasing resource demands of processing and managing gigantic video data. Researchers argue that efficient and scalable solutions can be achieved by pushing the processing to the edge of the system so that most of the processing takes place at the sensor itself [1–6]. As a result, we witness a number of workshops and conferences on distributed smart cameras [7, 8]. Smart cameras perform video analysis tasks at the sensor itself and send only an abstract description of the scene for further processing and viewing. It has been articulated that smart cameras are the key elements of the ongoing paradigm shift from central to distributed surveillance systems [9–11].

Despite great progress in terms of research, smart cameras have not seen enough success in commercial systems and real installations. In this paper, we investigate the reasons behind the restricted use of smart cameras through a comparative assessment. It is found that smart cameras are effective only

for sparse camera networks where multisensor information fusion is minimal, such as a highway traffic monitoring systems, but they are not suitable for applications requiring the assimilation of data from multiple cameras.

With the decreasing cost of video sensors, however, more applications are using densely placed cameras with overlapping views. For instance, in the surveillance context, multiple cameras with overlapping views are used to seamlessly track targets in the presence of occlusions. Typically, redundant sensors are used to achieve higher accuracy and robustness of detection tasks [11, 12]. Information from multiple cameras is fused to improve the accuracy and robustness of detection tasks [13–15]. This type of information fusion is not possible in smart camera systems as the videos are processed in isolation and only abstract data is available at the fusion node. In this way, smart cameras are not the best choice for synergistic integration of current and future research in interdisciplinary areas of multicamera applications. There are multiple limitations of smart cameras that hinder their general usage, such as

TABLE 1: Video analysis applications.

Application	Purpose	Analysis tasks
Surveillance and monitoring [11]	Anomaly detection, alarm generation	Object/face detection, recognition, and tracking
Ambient intelligence [44]	Supporting persons in daily life, living assistance	Posture recognition, face detection, and tracking
Smarthome [45]	Healthcare, eldercare	Activity detection, gait recognition
Teleconferencing [46]	Enhanced communication experience	Face detection and tracking
Traffic surveillance [47]	Monitoring and analysis of traffic flow	Segmentation, motion detection, and object classification
Crowd management [48]	Avoid congestion in narrow areas	Tracking, flow analysis
Pervasive computing [49]	Living assistance, healthcare	Activity and behaviour detection

- (i) multisensor coordination and information fusion is usually inefficient as video from each camera is processed in isolation;
- (ii) only metadata and compressed video data are available for multicamera coordination. Therefore, detection and recognition tasks involving multiple cameras perform poorly;
- (iii) the cost of smart cameras is too high in comparison to basic IP cameras, without equivalent benefit in performance;
- (iv) algorithms designed for smart cameras are custom designed and hence are very difficult to upgrade.

In this work we utilize cloud computing on a local area network (private cloud) as an alternative solution to scalability. We propose SmartHub as a logical entity which processes data from cameras that likely require information fusion. A number of SmartHub instances run on the cloud to process video from the cameras. SmartHub not only overcomes the limitations of smart cameras but also provides a scalable, distributed solution. Because the video streams from the cameras needing information fusion are processed at one node, SmartHub enables efficient multicamera information fusion. We study the trade-off between the scalability (in terms of the degree of processing distribution) and coordination (in terms of communication overhead) and propose SmartHub as an alternative to smart cameras.

With the increasing number of cameras, sending high quality video to the cloud may cause a bandwidth bottleneck. We propose subnet dependent geographical distribution of processing nodes and SmartHubs to avoid the bandwidth bottleneck. With the proposed distribution of processing nodes, high bandwidth data will remain within the subnet and only abstract data will flow across subnets.

The main contributions of this work are as follows.

- (i) We provide a comparative assessment of smart cameras with the conclusion that smart cameras are an inefficient choice for growing multicamera applications.
- (ii) We propose the cloud entity SmartHub that overcomes the limitations of smart cameras and propose a framework to make design decisions.

The rest of the paper is organized as follows. In Section 2 we review video analysis based applications and smart cameras. We discuss the limitations of smart camera based systems in Section 3. Section 4 describes SmartHub based system design and Section 5 describes the framework to make design decisions. We provide our conclusions in Section 6.

2. Context Description and Definitions

In this section we first describe potential applications where smart cameras can be used and derive a representative system architecture for these systems. Then we discuss smart camera works and how they are employed in video analysis systems.

2.1. Video Analysis Applications. A camera captures a snapshot of the scene in its view; in the same way a human eye observes a scene. The video captured by the camera is analysed to understand the semantics of the scene. Automatic video analysis is used to assist/automate decision-making in a number of application scenarios, a few of which are listed in Table 1.

The majority of video analysis applications are related to surveillance and monitoring. Another set of applications is concerned with healthcare and elderly care. Examining all these applications, we make the following observations.

- (i) The most common tasks are foreground detection, object/face detection, tracking, and activity/behaviour analysis.
- (ii) The tasks do not always follow a pipelined structure; that is, we generally need original video even for high level tasks such as tracking and activity detection.
- (iii) There is always a central unit that consolidates the analysis results and derives higher level semantics.

Based on these observations, a functional view of a typical video analysis system is drawn in Figure 1. In some cases, tracking is directly performed on the video. Object detection is still required to initialize the trackers. While intermediate functions (first three blocks of Figure 1) can be delegated to various distributed computing devices, the final aggregation and presentation generally take place at one (or more than one) central unit.

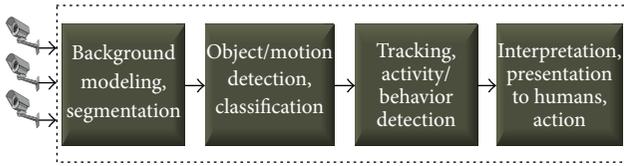


FIGURE 1: Functional view of typical video analysis systems.

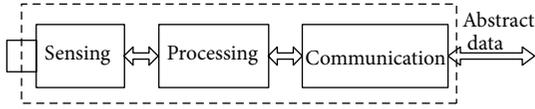


FIGURE 2: Smart camera architecture.

2.2. Smart Cameras. Earlier cameras used dedicated coaxial cables to transmit recorded video. Today, such cameras are almost completely replaced by IP cameras [16]. A basic IP camera captures images, compresses them, and streams the compressed video on the network [17]. In this paper, we will refer to these cameras as normal cameras.

A smart camera, on the other hand, integrates resource intensive advanced image and video processing techniques with compression and streaming. As shown in Figure 2, a smart camera consists of three main blocks: sensing, processing, and communication [4]. One of the initial works on smart cameras was by Moorhead and Binnie [18], who integrated edge detection in the camera. In [3], a smart camera extracts the high-level semantics of the scene it is capturing and sends them to the central unit. In this way, the smart cameras are mainly used to delegate detection and recognition tasks to the embedded platforms. Table 2 provides a list of smart camera works and the processing task implemented on the camera. To reduce the communication overhead, smart cameras analyse the image data and only transmit the abstract information [4, 19].

Processing video locally at the source camera reduces the communication load by avoiding the transmission of high quality images; only concise descriptions of extracted features are communicated for multicamera collaboration [9, 20–23]. Hence, smart cameras stream low-bandwidth processed information to save bandwidth [24–26]. Yet, for optimal performance, computer vision algorithms require heavy computing resources. There have been attempts to develop lightweight methods for smart cameras [27–30] to reduce resource needs. However, these ad hoc methods compromise the overall quality and do not extend easily. If there is any improvement in the original algorithm, the customized version may or may not agree to the improvement.

Based on the discussion above, smart cameras provide processing and bandwidth scalability only when

- (i) the processing tasks are not repeated; that is, if the foreground detection is done at the camera, it should not be repeated at the central unit;
- (ii) the data communicated from a smart camera is much less than the original data captured.

In the following section we show the effects of these constraints on the research in other interdisciplinary areas of video analysis systems. Subsequently, we propose a cloud entity, SmartHub, and demonstrate how it provides both scalability and synergistic integration with other research areas.

3. Limitations of Smart Cameras

The most important limitation of smart cameras is the limited opportunity for information fusion. In the process of video analysis, information fusion can take place at the following three levels [31].

- (i) *Data Level.* In this type of fusion, pixel values are directly compared to come to a conclusion; therefore it requires image data for fusion.
- (ii) *Feature Level.* In a more popular approach, features are extracted from the image and compared for detection. If features are not heavily compressed, they require significant bandwidth for transmission.
- (iii) *Decision Level.* This type of fusion is the most economical in terms of bandwidth overhead. The detection task is performed for individual cameras, and only final decisions from each video are fused together.

The smart camera systems only allow fusion at the decision level. In current multicamera systems, however, generally the cameras are densely placed with overlapping views which require data and feature level fusion [32–37]. To enable feature level fusion in smart camera systems, feature compression techniques have been proposed [25, 26]. However, feature compression is an ad hoc process and compromises the overall accuracy of the analysis task. We argue that the features are already compressed and further compression is unfavoured for future analysis techniques.

To assess the smart camera systems with overlapping views, we consider a scenario in which 4 cameras with overlapping views are tracking a person at a subway station. If the cameras do not communicate with each other to save bandwidth, one object is being tracked by 4 cameras, which is a redundancy rate of 75%. There have been research works in which only one master camera with the best view tracks the object [19, 21, 38]. In this case we have 4 hardware units capable of tracking but only one unit is being used at a time. The other 3 units are underutilized, which increases the overall cost per object of the system.

In Figure 3, we show the processing times of the steps of a typical video analysis system. Four videos with overlapping views from a multicamera video dataset [39] are used for this evaluation. While the foreground detection only depends on the frame resolution, the processing times of detection and tracking are proportional to the computational load of each step. It is evident from the figure that tracking is a computationally intensive task. It would require expensive hardware to track objects using state-of-the-art tracking methods, such as particle filters [40].

TABLE 2: Brief summary of smart camera works.

Work	Tasks
Moorhead and Binnie [18]	Edge detection
Wolf et al. [3]	Human gesture recognition, region extraction, contour detection, and template matching
Lin et al. [23]	Gesture recognition
Muehlmann et al. [50]	Real-time tracking
Heyrman et al. [51]	Motion detection
Bramberger et al. [9]	Traffic surveillance, multicamera object tracking
Chen and Aghajan [19]	Gesture recognition using smart camera network
Quaritsch et al. [21]	Multicamera tracking, camShift
Rinner and Wolf [4]	scene abstraction
Aghajan et al. [20]	Human pose estimation
Sankaranarayanan et al. [24]	Object detection, recognition, and tracking
Tessens et al. [30]	Foreground detection, subsampling
Wang et al. [22]	Tracking, event detection, and foreground detection
Casares and Velipasalar [28]	Foreground detection, tracking feedback
Sidla et al. [52]	Traffic monitoring
Pletzer et al. [53]	Traffic monitoring, vehicle speed, and vehicle count
Wang et al. [29]	Foreground detection, contour tracking
Cuevas and Garcia [54]	Single camera tracking, background modelling

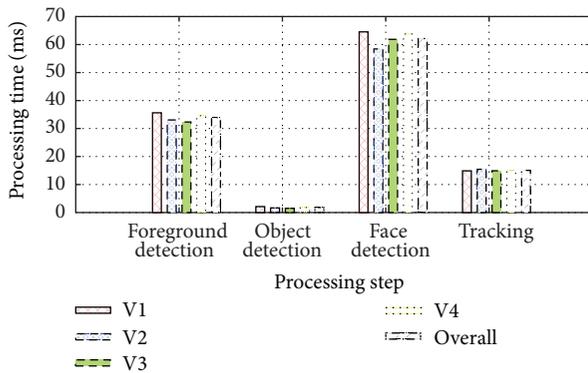


FIGURE 3: Processing times of individual steps.

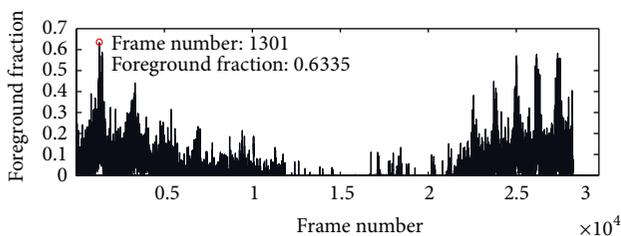


FIGURE 4: The amount of foreground in real surveillance footage of 24 hours.

In order to decide the best view, smart cameras need to share foreground information with each other. Figure 4 shows the fraction of image area that belongs to the foreground for real surveillance footage of 24 hours. We can see that

the foreground area may vary from nothing to 63%. Sharing such a large amount of foreground information would use a great deal of bandwidth. Furthermore, with the overlapping views, the best view can change between consecutive frames. This would require frequent changes in the role of the master camera. Frequently changing the master camera would cause additional bandwidth and processing overhead.

4. SmartHub System

We propose to use normal IP cameras to capture video and delegate all video analysis tasks to the private cloud. In our discussion, the private cloud is defined as the distributed computing nodes on the same Local Area Network (LAN) to which the cameras are connected. The block diagram of the proposed SmartHub system is shown in Figure 5. The cameras only perform the basic tasks of video compression and streaming. Video analysis tasks of detection, recognition, and tracking are performed by the SmartHub cloud entity. Note that a normal IP camera with basic encoding and streaming capabilities is approximately 10 times cheaper than a smart camera capable of detecting activities.

SmartHub fuses visuals from multiple cameras and provides a set of services to the central unit such as object detection, face detection, and tracking. The central unit can query SmartHub to receive continuous information (video streams) or event information in terms of detected objects. Because the information fusion takes place at SmartHub, it does not need to send video from all cameras to the central unit but only the most informative view. Furthermore, SmartHub can create a synthetic view (e.g., a 3D model) of the scene and send that information.

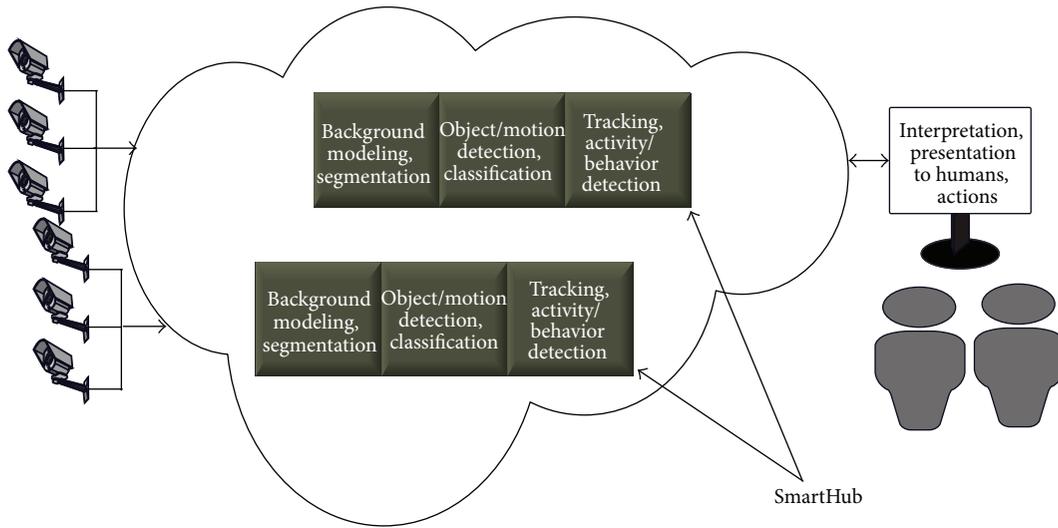


FIGURE 5: Cameras capture video and send it to the SmartHub nodes in the cloud. SmartHubs process the data as a service to the central security unit. Based on the situation, the central unit can enable or disable a particular service.

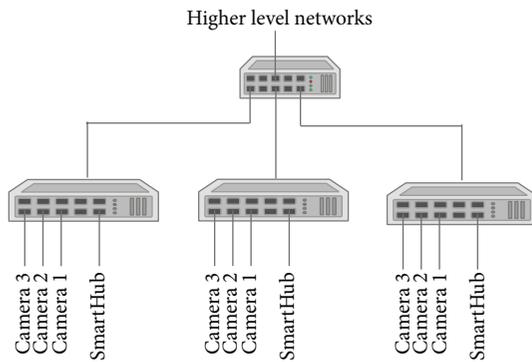


FIGURE 6: The cameras and the corresponding SmartHub processing node are connected to the same switch.

In this system, the cameras that are likely to coordinate are connected to one processing node (SmartHub), which creates an abstract understanding of the coverage area and shares the coordinated and synchronized information with the other processing nodes and central unit.

To avoid the bandwidth bottleneck, we exploit the organization of LAN infrastructure. A LAN consists of multiple switches, arranged in a hierarchical fashion. The cameras are connected to the lowest level of switches. The data going out of the switch has a bandwidth limitation depending on the number of other switches in the network and data flow. For communication within a switch, however, almost full Ethernet bandwidth is available.

We propose that the cloud processing nodes should be connected to the switch to which the corresponding cameras are connected. With that setup, we would be able to send high quality video to SmartHub for information fusion, and abstract information can be forwarded to the central unit through higher level switches. The proposed scheme is shown in Figure 6.

A SmartHub based system has various merits over both centralized system and smart camera based system. These merits and salient features of SmartHub based system are discussed below. The topics for the discussion have been chosen in consideration of the current state of research focusing on design and quality issues.

4.1. Storage Scalability. A SmartHub with storage capabilities can provide an excellent distributed data storage architecture. Storage at each smart camera is costly, whereas unified storage at the central location is not scalable. Hence, SmartHub can provide a midway solution for storage.

Storing video at smart cameras is costly because smart cameras generally use flash memory. Adversely, SmartHubs can employ disk memory on the cloud. Table 4 compares the price of hard disk and flash memory. We see that the cost per GB for hard disk memory is from 6.25 to 8.9 cents, whereas flash memory prices can range from 183 to 210 cents. Furthermore, the price of compact memory used in smart cameras increases more rapidly with capacity.

4.2. Reduced Processing Repetition. Video analysis involves low level processing (background-foreground classification) and high level processing (blob detection and tracking). In smart camera systems, the low level steps of background and foreground detection are repeated both at the camera and at the processing node that fuses data from multiple cameras. In SmartHub, since object detection and fusion are performed at the same place (cloud), there is no repetition of processing. We see in Figure 3 that SmartHub needs 30% less processing for the same task, as foreground detection is only done once.

4.3. Lower per Sensor Cost. The per sensor cost of the overall system is very high in smart camera systems due to the enhanced capabilities of smart cameras. On the other hand, SmartHub offers reduced cost as there is only one hardware

TABLE 3: The comparison of SmartHub, smart camera, and centralized systems.

	Processing and storage scalability	Sensor coordination and synchronization difficulty	Bandwidth requirement	Cost (per sensor)	Tracking accuracy in the presence of occlusions	Processing repetition
Smart camera	High	High	Low	High	Low	High
Centralized	Low	High	High	Medium	Medium	No
SmartHub	High	Low	Low	Low	High	Low

TABLE 4: Approximate current prices of flash and hard disk memory.

Seagate hard disk	Kingston flash drive
1 TB—\$89.99	8 GB—\$14.71
2 TB—\$119.99	16 GB—\$33.71
3 TB—\$179.99	32 GB—\$71.71
4 TB—\$249.99	64 GB—\$135.71
Cost per GB = 6.25 to 8.9 cents	Cost per GB = 183 to 210 cents

unit for a group of sensors. This topic has been included to emphasize that the smart cameras add to the cost of the system without providing equivalent benefits. SmartHub provides better performance with reduced cost. A normal IP camera costs around \$100, while a smart camera costs approximately 10 times more than a normal camera. If we consider a 4-camera system, building the system with smart cameras would cost at least \$4000. Alternatively, a cloud processor with processing power equivalent to 4 cameras would cost less than \$1000. Hence, a SmartHub system with 4 cameras would only cost \$1400, 65% lesser than the smart camera system. Furthermore, the processing power over cloud is available for other applications when the video processing workload is minimal [41].

4.4. Others. Sensor coordination and synchronization is also difficult in centralized and smart camera based systems due to random network delays at intermediate nodes. For a fixed bandwidth, SmartHub will provide best tracking performance as high quality video from overlapping view cameras is available at one node without causing additional bandwidth overhead. Similarly, to achieve the same level of tracking accuracy, centralized system will need high quality video from multiple cameras causing large bandwidth overhead. A summary of the above discussion is provided in Table 3.

5. Design and Analysis

The main question in designing the system is the number of cameras to be connected to a SmartHub. To determine a suitable number, we conduct a task based analysis. Consider a video analysis task of human detection and matching. We chose this task because it is a very common task for video based applications [42, 43]. In this task, we detect the humans in each camera and match them across cameras to obtain the best view of a person.

The task can be accomplished in both centralized and distributed architectures. However, each architecture will have different overheads in accomplishing the task. The overheads are abstracted in two categories: communication overhead and processing overhead.

5.1. Communication Overhead. For human detection and matching, high quality image regions need to be transmitted to other processing nodes over the network. For cameras with overlapping views, it is very useful to share the facial data to match humans and track across obstacles. For nonoverlapping cameras, the human data is only required when tracking a person over a larger territorial region or when there is a specific threat generated at one camera and the person needs to be detected at all possible places. Therefore, we assume that information from each pair of cameras needs to be fused with a nonzero probability. This implies that in a purely distributed smart camera network every pair of cameras needs to communicate with each other probabilistically.

We have modelled the communication overhead in terms of camera overlap and data sharing requirements. Let $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$ be the set of cameras where n is the number of cameras. Let $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ be the set of areas covered by the corresponding camera. Let \bar{w} and \bar{h} be the average width and height of the human region in number of pixels. Now, the total communication overhead for a smart camera network is calculated as

$$\omega^s = \sum_{i=1}^n \sum_{j=1, j \neq i}^n o_{ij} * \bar{w} * \bar{h}, \quad (1)$$

where o_{ij} is the normalized intersection of the areas covered by the i th and j th cameras; that is,

$$o_{ij} = \begin{cases} \frac{a_i \cap a_j}{a_i \cup a_j} & \text{if } \frac{a_i \cap a_j}{a_i \cup a_j} > 0.01, \\ 0.01 & \text{otherwise.} \end{cases} \quad (2)$$

In a SmartHub based architecture, the communication overhead is mainly due to the communication among SmartHubs. Let \mathcal{S}_i be the set of cameras connected to the i th SmartHub. The total amount of data flowing out of the i th SmartHub is calculated as

$$\omega_i = \sum_{i \in \mathcal{S}_i} \sum_{j=1, j \notin \mathcal{S}_i}^n o_{ij} * \bar{w} * \bar{h} \quad (3)$$

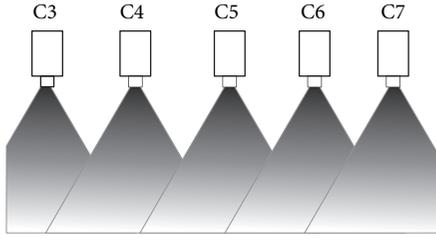


FIGURE 7: Camera placement scenario.

and the total overhead is the sum of the overheads due to all SmartHubs; that is,

$$\omega = \frac{1}{\omega^s} \sum_{i=1}^{n^s} \omega_i, \quad (4)$$

where ω_i is the overhead due to the i th SmartHub, ω is the overhead in SmartHub based architecture, and n^s is the number of SmartHubs. If η is the number of cameras connected to a SmartHub, the total number of SmartHubs is calculated as

$$n^s = \left\lceil \frac{n}{\eta} \right\rceil. \quad (5)$$

Note that the communication overhead depends mainly on the communication requirements between the processing nodes and on the camera placement.

5.2. Processing Distribution. In a smart camera network, all the processing is pushed to the edge. The processing tasks are completely distributed among processing units. With the introduction of SmartHubs, we bring the processing one level higher. In a completely centralized system, all the processing is done at a single node. This introduces a processing bottleneck and a single point of failure. Therefore, distributed processing is a desired characteristic of an architecture and it is measured as processing distribution (γ).

If ρ is the amount of processing required to complete the task, the processing load on a single node in a smart camera network is ρ . In a SmartHub based architecture, the processing load (Ψ) on a SmartHub is

$$\Psi = \rho * \eta. \quad (6)$$

Consequently, the processing distribution is calculated as

$$\gamma = \beta e^{-\Psi/n}, \quad (7)$$

where β is a normalizing coefficient, which is chosen to be $e^{\rho/n}$ to limit the maximum value of processing distribution to 1 for the smart camera case. The minimum value of distribution is β/e^ρ .

To measure the most adequate number of cameras for a SmartHub, we define an optimization function as follows:

$$\Gamma = \gamma - \omega \quad (8)$$

to emphasize that we wish to reduce the communication overhead while still being able to distribute processing.

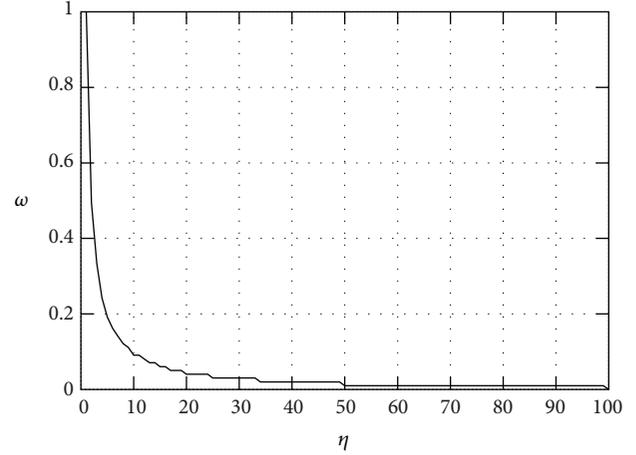


FIGURE 8: Communication overhead versus number of cameras per SmartHub.

5.3. Experimental Results. In this section we obtain the number of cameras to be connected to a SmartHub in a given scenario. While the framework can be applied for any task and any given scenario, we consider a camera placement scenario as given in Figure 7 for experimental purposes.

For the experiments, we considered 100 cameras ($n = 100$) and assumed that the amount of processing required for the detection and recognition tasks is unity; that is, $\rho = 1$. The height and width of the human blob are also assumed to be unity ($\bar{w} = 1, \bar{h} = 1$). For the scenario given in Figure 7, consecutive cameras have a fractional overlap of 0.16, otherwise no overlap; that is,

$$o_{ij} = \begin{cases} 0.16, & |i - j| = 1, \\ 0.01, & \text{otherwise.} \end{cases} \quad (9)$$

The resulting communication overhead for the given placement is shown in Figure 8. We see that the overhead initially reduces rapidly until $\eta \approx 4$ and then becomes horizontal. Thus, beyond a point, adding more cameras to a SmartHub does not provide any significant advantage in reducing the bandwidth requirement.

The processing distribution decreases linearly with the number of cameras connected to a SmartHub as shown in Figure 9. The combined optimization function is plotted in Figure 10. With the help of this figure, we conclude that 4 to 15 cameras could be connected to a SmartHub for adequate trade-off between communication overhead and scalability.

5.4. Limitations. While there are multiple advantages of SmartHub in multicamera systems, these are tightly coupled with network topology. The proposed SmartHub architecture assumes that the network follows tree topology. Experiments also reveal that the benefits of SmartHub are significant only when the number of cameras is large and the task at hand requires fusion of video from multiple cameras.

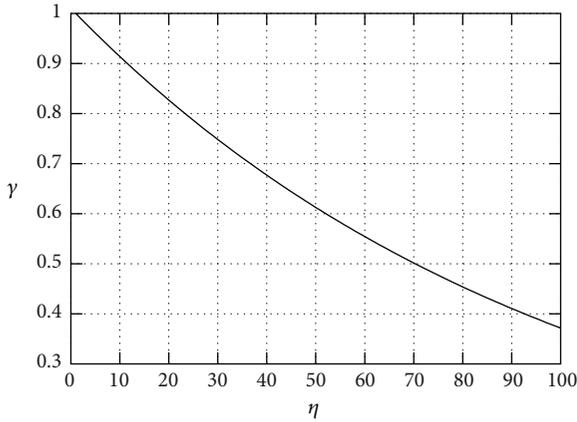


FIGURE 9: Processing distribution versus number of cameras per SmartHub.

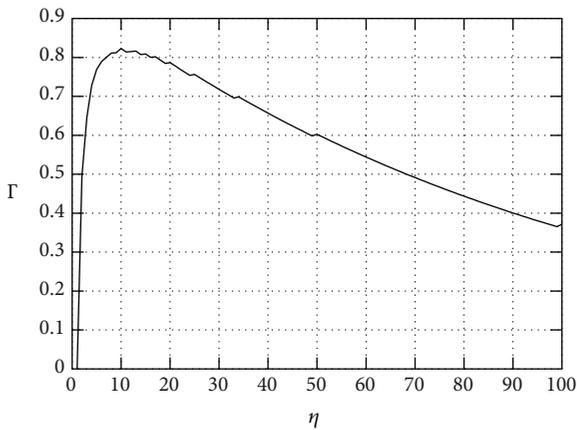


FIGURE 10: Optimization function.

6. Conclusions

Smart cameras are inefficient and costly in scenarios with multiple overlapping cameras. Such scenarios are common in setups using cheap video sensors. The scalability achieved by smart cameras puts additional constraints on the system which compromise the performance of vision algorithms. Similar scalability is achieved by processing data over cloud with SmartHub based architecture. The given framework can be used to calculate an adequate number of cameras to be connected to a SmartHub for a given camera placement. For a general placement with consecutive overlapping cameras, it is adequate to have 4 to 15 cameras per SmartHub. In the future, we intend to deploy SmartHub on dedicated hardware units and explore more design decisions.

Conflict of Interests

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

Acknowledgement

The work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Grant nos. 210345 and 371714.

References

- [1] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "Mesheye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 360–369, April 2007.
- [2] K. Sato, T. Maeda, H. Kato, and S. Inokuchi, "CAD-based object tracking with distributed monocular camera for security monitoring," in *Proceedings of the 2nd CAD-Based Vision Workshop*, pp. 291–297, February 1994.
- [3] W. Wolf, B. Ozer, and T. Lv, "Smart cameras as embedded systems," *Computer*, vol. 35, no. 9, pp. 48–53, 2002.
- [4] B. Rinner and W. Wolf, "An introduction to distributed smart cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1565–1575, 2008.
- [5] W. Wolf, "Distributed peer-to-peer smart cameras: algorithms and architectures," in *Proceedings of the 7th IEEE International Symposium on Multimedia (ISM '05)*, p. 178, December 2005.
- [6] J. Mallett and V. M. Bove Jr., "Eye society," in *Proceedings of the International Conference on Multimedia and Expo (ICEM '03)*, pp. 17–20, 2003.
- [7] J. Camillo Taylor and Babak Shirmohammadi, "Self localizing smart camera networks and their applications to 3D modeling," in *Proceedings of the Workshop on Distributed Smart Cameras (DSC '06)*, Boulder, Colo, USA, October 2006.
- [8] S. Hengstler, H. Aghajan, and A. Goldsmith, "Application-oriented design of smart camera networks," in *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '07)*, pp. 12–19, Vienna, Austria, 2007.
- [9] M. Bramberger, A. Doblender, A. Maier, B. Rinner, and H. Schwabach, "Distributed embedded smart cameras for surveillance applications," *Computer*, vol. 39, no. 2, pp. 68–75, 2006.
- [10] C. T. Ye, T. D. Wu, Y. R. Chen et al., "Smart video camera design-realtime automatic person identification," in *Advances in Intelligent Systems and Applications*, vol. 2, pp. 299–309, Springer, 2013.
- [11] X. Wang, "Intelligent multi-camera video surveillance: a review," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 3–19, 2013.
- [12] L. Tessens, M. Morbee, H. Lee, W. Philips, and H. Aghajan, "Principal view determination for camera selection in distributed smart camera networks," in *Proceedings of the 2nd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '08)*, September 2008.
- [13] C. S. Regazzoni and A. Tesei, "Distributed data fusion for real-time crowding estimation," *Signal Processing*, vol. 53, no. 1, pp. 47–63, 1996.
- [14] B. R. Abidi, N. R. Aragam, Y. Yao, and M. A. Abidi, "Survey and analysis of multimodal sensor planning and integration for wide area surveillance," *ACM Computing Surveys*, vol. 41, no. 1, article 7, 2008.
- [15] G. Scotti, L. Marcenaro, C. Coelho, F. Selvaggi, and C. Regazzoni, "Dual camera intelligent sensor for high definition 360 degrees surveillance," *IEE Proceedings on Vision, Image and Signal Processing*, vol. 152, no. 2, pp. 250–257, 2005.

- [16] H. Kruegle, *CCTV Surveillance: Video Practices and Technology*, Heinemann, Butterworth, Malaysia, 2011.
- [17] M.-J. Yang, J. Y. Tham, D. Wu, and K. H. Goh, "Cost effective IP camera for video surveillance," in *Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA '09)*, pp. 2432–2435, May 2009.
- [18] T. W. J. Moorhead and T. D. Binnie, "Smart CMOS camera for machine vision applications," in *Proceedings of the 7th International Conference on Image Processing and its Applications*, vol. 2, pp. 865–869, July 1999.
- [19] W. Chen and H. Aghajan, "Model-based human posture estimation for gesture analysis in an opportunistic fusion smart camera network," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS '07)*, pp. 453–458, September 2007.
- [20] H. Aghajan, C. Wu, and R. Kleihorst, "Distributed vision networks for human pose analysis," in *Signal Processing Techniques for Knowledge Extraction and Information Fusion*, pp. 181–200, Springer, 2008.
- [21] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof, and B. Strobl, "Autonomous multicamera tracking on embedded smart cameras," *EURASIP Journal on Embedded Systems*, vol. 2007, Article ID 92827, 2007.
- [22] Y. Wang, S. Velipasalar, and M. Casares, "Cooperative object tracking and composite event detection with wireless embedded smart cameras," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2614–2633, 2010.
- [23] C. H. Lin, T. Lv, W. Wolf, and I. B. Ozer, "A peer-to-peer architecture for distributed real-time gesture recognition," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, pp. 57–60, June 2004.
- [24] A. C. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa, "Object detection, tracking and recognition for multiple smart cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1606–1624, 2008.
- [25] A. Y. Yang, S. Maji, C. M. Christoudias, T. Darrell, J. Malik, and S. S. Sastry, "Multiple-view object recognition in smart camera networks," in *Distributed Video Sensor Networks*, pp. 55–68, Springer, 2011.
- [26] M. Makar, C. L. Chang, D. Chen, S. S. Tsai, and B. Girod, "Compression of image patches for local feature extraction," in *Acoustics*, in *IEEE International Conference on Speech and Signal Processing (ICASSP '09)*, pp. 821–824, IEEE, 2009.
- [27] M. Casares and S. Velipasalar, "Adaptive methodologies for energy-efficient object detection and tracking with battery-powered embedded smart cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1438–1452, 2011.
- [28] M. Casares and S. Velipasalar, "Resource-efficient salient foreground detection for embedded smart cameras by tracking feedback," in *Proceedings of the 7th IEEE International Conference on Advanced Video and Signal Based (AVSS '10)*, pp. 369–375, September 2010.
- [29] Q. Wang, P. Zhou, J. Wu, and C. Long, "Raffd: resource-aware fast foreground detection in embedded smart cameras," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '12)*, pp. 481–486, IEEE, 2012.
- [30] L. Tessens, M. Morbee, W. Philips, R. Kleihorst, and H. Aghajan, "Efficient approximate foreground detection for low-resource devices," in *Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '09)*, pp. 1–8, September 2009.
- [31] P. K. Atrey, M. S. Kankanhalli, and R. Jain, "Information assimilation framework for event detection in multimedia surveillance systems," *Multimedia Systems*, vol. 12, no. 3, pp. 239–253, 2006.
- [32] D. M. Gavrila and L. S. Davis, "3-D model-based tracking of humans in action: a multi-view approach," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pp. 73–80, IEEE, June 1996.
- [33] A. Sundareshan and R. Chellappa, "Model driven segmentation of articulating humans in Laplacian Eigenspace," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1771–1785, 2008.
- [34] Q. Delamarre and O. Faugeras, "3D articulated models and multi-view tracking with silhouettes," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, pp. 716–721, September 1999.
- [35] N. T. Nguyen, S. Venkatesh, G. West, and H. H. Bui, "Multiple camera coordination in a surveillance system," *Acta Automatica Sinica*, vol. 29, no. 3, pp. 408–422, 2003.
- [36] Y. Yao, C.-H. Chen, A. Koschan, and M. Abidi, "Adaptive online camera coordination for multi-camera multi-target surveillance," *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 463–474, 2010.
- [37] H. Mitchell, "Introduction," in *Data Fusion: Concepts and Ideas*, pp. 1–14, Springer, 2012.
- [38] M. Bramberger, M. Quaritsch, T. Winkler, B. Rinner, and H. Schwabach, "Integrating multi-camera tracking into a dynamic task allocation system for smart cameras," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS '05)*, pp. 474–479, September 2005.
- [39] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.
- [40] B. Ristic, S. Arulampalam, and N. J. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House Publishers, 2004.
- [41] M. Saini, X. Wang, P. K. Atrey, and M. Kankanhalli, "Adaptive workload equalization in multi-camera surveillance systems," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 555–562, 2012.
- [42] H. M. Dee and S. A. Velastin, "How close are we to solving the problem of automated visual surveillance?: a review of real-world surveillance, scientific progress and evaluative mechanisms," *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 329–343, 2008.
- [43] Q. Ye, Z. Han, J. Jiao, and J. Liu, "Human detection in images via piecewise linear support vector machines," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 778–789, 2013.
- [44] D. J. Cook, J. C. Augusto, and V. R. Jakkula, "Ambient intelligence: technologies, applications, and opportunities," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 277–298, 2009.
- [45] M. Skubic, G. Alexander, M. Popescu, M. Rantz, and J. Keller, "A smart home application to eldercare: current status and lessons learned," *Technology and Health Care*, vol. 17, no. 3, pp. 183–201, 2009.
- [46] A. Eleftheriadis and A. Jacquin, "Automatic face location detection and tracking for model-assisted coding of video teleconferencing sequences at low bit-rates," *Signal Processing: Image Communication*, vol. 7, no. 3, pp. 231–248, 1995.

- [47] T. Semertzidis, K. Dimitropoulos, A. Koutsia, and N. Grammalidis, "Video sensor network for real-time traffic monitoring and surveillance," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 103–112, 2010.
- [48] I. G. Georgoudas, G. C. Sirakoulis, and I. T. Andreadis, "An anticipative crowd management system preventing clogging in exits during pedestrian evacuation processes," *IEEE Systems Journal*, vol. 5, no. 1, pp. 129–141, 2011.
- [49] T. Gu, H. K. Pung, and D. Q. Zhang, "A service-oriented middleware for building context-aware services," *Journal of Network and Computer Applications*, vol. 28, no. 1, pp. 1–18, 2005.
- [50] U. Muehlmann, M. Ribo, P. Lang, and A. Pinz, "A new high speed CMOS camera for real-time tracking applications," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 5, pp. 5195–5200, May 2004.
- [51] B. Heyrman, M. Paindavoine, R. Schmit, L. Letellier, and T. Collette, "Smart camera design for intensive embedded computing," *Real-Time Imaging*, vol. 11, no. 4, pp. 282–289, 2005.
- [52] O. Sidla, M. Rosner, M. Ulm, and G. Schwingshackl, "Traffic monitoring with distributed smart cameras," in *IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics*, 2012.
- [53] F. Pletzer, R. Tusch, L. Boszormenyi, and B. Rinner, "Robust traffic state estimation on smart cameras," in *Proceedings of the IEEE 9th International Conference on Advanced Video and Signal-Based Surveillance (AVSS '12)*, pp. 434–439, IEEE, 2012.
- [54] C. Cuevas and N. Garcia, "Efficient moving object detection for lightweight applications on smart cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 1, pp. 1–14, 2013.

