

## Research Article

# CRAM: A Conditioned Reflex Action Inspired Adaptive Model for Context Addition in Wireless Sensor Networks

Majid Hussain,<sup>1</sup> Muhammad Farrukh Shafeeq,<sup>1</sup> Sana Jabbar,<sup>2</sup>  
Ali Hammad Akbar,<sup>1,2</sup> and Shahzad Khalid<sup>3</sup>

<sup>1</sup>Department of Computer Science and Engineering, University of Engineering & Technology (UET), Lahore 54890, Pakistan

<sup>2</sup>Alkharizmi Institute of Computer Science, University of Engineering & Technology (UET), Lahore 54890, Pakistan

<sup>3</sup>Bahria University Wireless Research Center, Bahria University, Islamabad 44000, Pakistan

Correspondence should be addressed to Majid Hussain; majidhussain1976@gmail.com

Received 8 October 2015; Accepted 15 February 2016

Academic Editor: Yurong Qian

Copyright © 2016 Majid Hussain 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.

In Wireless Sensor Networks (WSNs), Context Awareness is typically realized through Context Aware Systems (CASs). Although almost each CAS follows sense-decide-act cycle, the notion of context is hardwired into the applications; that is, when an event is triggered, the sense-decide-actuate cycle runs and performs required actuation. In situations, for instance, whenever the same event is triggered, the cycle produces the same actuation through mechanical use of the same resources, posing the same processing and time. In this paper, we propose CRAM, a context added system in which actuations once performed by the system help it to internally evolve by serving as new contexts. As the system is exposed to more situations overtime, its context repository is enriched through such retrospective contexts, gradually letting it perform internal actuation through improved introspective contexts. This internal actuation leads the system towards the evolution of intelligent processing by reducing the independent function of decision in sense-decide-actuate cycle and merging it with new context. Finally, the system reaches a juncture where recurrence of each event proves to be a stimulus for the system to respond impulsively, through priming memory of introspective contexts, to achieve an imitation of learned reflex action resulting into reduced time and energy expenditure.

## 1. Introduction

Wireless Sensor Networks (WSNs) comprising a large number of interconnected sensing nodes have been the subject of intensive research for the last one and half decades. The growing applications of WSNs in more or less each aspect of human life have made them the up-to-the-minute topical research area. The broad canvas of WSNs novel applications ranges from volcano, glaciers, and aeroground monitoring, reptile tracking, maritime navigations, and studies of human anatomy at much larger scale to the surveillance and defense which conclusively illustrates the omnipresence of WSNs in human atmosphere. Visual Sensor Networks (VSNs)—special featured WSNs are the networks of visual devices, mostly cameras, equipped with enough onboard processing power to support collaborative image analysis. Being highly diverse in operations, VSNs are integrated with such capabilities where local processing controls video and image data acquisition,

removes cross-layer correlations, and aggregates such data to transmit only what is essential. To strike at such optimal trade-off amongst performance and QoS guarantees amidst time synchronization, storage capability, and multicamera collaboration necessitates autonomous reasoning and decision making in unison in a highly distributed manner.

To achieve such autonomous reasoning, VSNs must be empowered with visual context awareness. Regardless of the sensing type, context awareness is the ability of a system, artifact, or service to be aware of its physical environment and to respond intelligently. In particular case of VSNs, visual context aware systems augment sensing devices with such capability that comprehends the user's real time visual perception and defines corresponding interactions with its environs in a particular situation [1]. In order to accomplish the targeted task, these context aware systems follow a sense-decide-actuate and adapt cycle in which they acquire the context, dig out the situation, reason and decide the suitable

actuation, and then adapt to the resulting situation. A context aware system that has gone through multiple actuations in response to multiple and even so repeating events does not learn the pattern of the occurrences of events. It means that when an event is triggered, it is treated afresh and the system always responds by executing the sense-decide-actuate cycle and carries out the requisite task. It is interesting to observe, however, that for scenarios in which the event is reiterated the sense-decide-actuate cycle is executed in its entirety once again. In other words, a context aware system deals with the same events again and again but it does not have the capability to establish an association between such recurrences of events in order to perform adaptation of its internal functionality. This limitation of context aware systems is exacerbated in VSNs because the recurrence of a visual event in itself is always treated as the arrival of a new scene or a video. Since a context has not been predefined for such recurrence of visual events as a metaevent, a context aware system becomes static and performance limited. The same amount of image processing is replicated at each cooperating node during each new occurrence of the same event and the accumulated processing results are considerably higher. This overprocessing resulting in corresponding time and energy expenditures can be restricted through the knowledge, that is, stored context with associated actuation learned by the system based on its prior encounter with the same event. This associative learning as a metaevent context helps the system to respond impulsively against the recurrence of an event emulating a conditioned reflex arc (reflex arc is the impulsive response of a biological system that bypasses the brain through conditioned behavior). Simultaneously, the actuation once performed by the system helps the system to internally evolve by acting as a new context. As the system is exposed to more situations where the same context occurs over time, its context repository is enriched through such retrospective contexts, gradually letting it perform internal actuation through improved introspective contexts which are learnt-and-stored actuations. Finally, it reaches a juncture where a new situation demands minimal external actuation, hence transforming it into a *context added system*, an ultimate imitation of an autonomic system.

In this paper, we propose CRAM, a novel cut-through processing paradigm for contemporary visual context aware systems. CRAM proposes memory-based visual context addition through a learnt reflex arc implementation. The reflex action is realized through a tenon-mortise layered architecture that uses *priming* memory to associate prior experiences with newer contexts. Using an analytical model and an underlying testbed based on VISTA by Jabbar et al. [2], we show that CRAM results in significant reduction in image processing load, provides energy efficiency, and improved compliance to end-user delay bounds and visual accuracy requirements.

The organization of this paper is as follows: Section 2 briefly presents the contemporary related work on the field. Section 3 describes in detail the proposed architecture. Section 4 deals with the memory hierarchy of CRAM and its similitude with human memory. Section 5 shows performance evaluation through simulation based results

while Section 6 presents an analytical model to validate this research work. Section 7 concludes the paper.

## 2. Literature Survey

This section is tetrapartite. The first part presents context and situation. The second part presents context aware systems in general and visual context aware systems in particular. The third part presents building blocks of context aware systems. And finally the last part explores contemporary applications of bioinspired reflex arc on computing systems.

*2.1. Context and Situation.* One of the prime challenges which have been faced by the researchers is to define the context and exhibit it as a different expression from situation. Baldauf et al. [3], Ryan et al. [4], Abowd et al. [5], and Cheverst et al. [6] refer to context as user's identity, current location, environment, and time illustrating an explicitly non-all-inclusive depiction of context. Dey [7] describes the context as the user's focus of attention, emotional state, date and time, location, and orientation as well as bits-and-pieces and people in user's vicinity. It opens up a new omnidimensional nature of context where sensing, perception, and measurement of sentiments and conjecturing the focus of intention are of elementary concern. Brown [8] defines context as the elements located in the user's environment about which the computer has the information. These sorts of definitions are often too wide-ranging. Conceivably, the most often used definition has been offered by Abowd et al. [9]. These authors refer to the context as "any information that can be used to characterize the situation of entities (i.e., a person, place, or object) that are considered relevant to the interaction between a user and an application including the user and application themselves." It is generally agreed upon that no single definition of the context even in the contemporary research has been reached. Hull et al. [10] attempt to establish relationship between the context and situation with the former being the characteristics of the latter. Since the concept of situation is very closely related to the notion of context, various authors have continued to relate the two of them. Zimmermann et al. [11] suggest that situation can be taken as an instantaneous and structured representation of a part of the context which may be directly compared with the snapshot taken by a camera. Loke [12] remarks that the situation can be viewed as being at a higher level of abstraction than context. Baker et al. [13] describe the situation as "the complete state of universe at an instance of time." Other studies state that sensors can be used to capture the context and construct high level context models of part of the real world. These models can further be used for the recognition of situation and its corresponding reasoning.

*2.2. Context Awareness and Context Aware Systems.* With a background of user's increasing expectations from its environment to adapt to itself, context aware systems have emerged as powerful means to synergize context generating sensors, wireless networks, and the consequent context utilization. From an interactive perspective, as mentioned by Schmidt [14], context aware systems show either of the

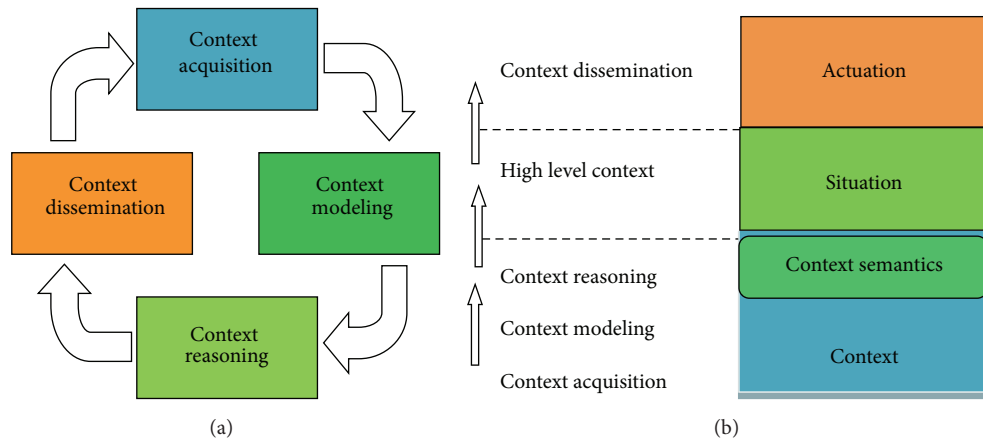


FIGURE 1: (a) Context based building blocks of CAS. (b) Conceptual building blocks of CAS.

two behaviors, adaptive or proactive. Adaptive CASs act on behalf of users, try to adapt to user's context, and are often single trigger based, while, on the other hand, proactive CASs require user's involvement and interaction as they are multitrigged. From an architectural perspective, the CASs can be standalone, centralized, or distributed. Standalone architecture is the simplest and easily deployable but it does not allow internode sharing of context limiting it only for small domain specific applications. In centralized systems, the whole range of sensors and devices is connected to a context server equipped with necessary computing and storage capacity making it simple to add and exchange context offering devices. Chatterjea et al. and Jin and Li [15, 16] propose that while being a single point of failure, the centralized approach is particularly not suited for CASs because the delay involved in context update might change the context itself at the source, whereas Zafeiropoulos et al. [17] take the view that distributed systems offer to use locality-based context acquisition and processing. A comprehensive classification has also been performed by Žontar et al. [18] on the basis of such kinds of architecture.

**2.3. Building Blocks of Context Aware Systems.** A CAS is a complex system that is often defined and composed to meet the desired level of context processing. Usually, context as the pivotal information is the main emphasis of CASs for acquisition, modeling, reasoning, and dissemination. What differs in various approaches is the operational flow and extent of these.

Perera et al. [19] present a cyclic pathway to traverse the context based hierarchy of the building blocks as illustrated in Figure 1(a). Context can be acquired through a diversity of sources. These sources include physical, virtual, and logical sensors. Physical sensors are tangible sources for context acquisition which provide low-level context which is raw, less meaningful, and noisy. Virtual sensors are not direct resources of context but they retrieve diverse types of data from different sources (e.g., phone directories, social websites, profiles, and databases) and publish it in the form of context. The logical sensors (software sensors) combine the physical and virtual sensors to produce high level and more

meaningful context (e.g., weather forecasting web service combines data from temperature, humidity, and wind sensors and corresponding weather maps and seasonal calendars to produce the weather information). Context modeling is the process of defining the context in terms of its characteristics, aspects, quality, and interrelationship with previous contexts and with the set of queries it corresponds to. Context reasoning includes preprocessing, fusion, and inference of context. It handles imperfection and uncertainty of raw data in order to deduce information and in best cases knowledge as high level context [20]. Finally, context dissemination or distribution is its propagation to other entities.

A number of researchers [13, 21, 22] have followed a layered approach to elaborate the fundamental elements of CASs as shown in Figure 1(b). The primary processes of context acquisition, modeling, and reasoning have jointly been incorporated in context and its semantic layers. Context and situation work in cohesion to realize context awareness. Finally, situation as high level context is disseminated for realizing context awareness through actuation layer.

Based upon the above discussed hierarchy, the following CASs are presented. Chen et al. [23] present Context Broker Architecture (CoBrA) to support context aware systems in smart and active spaces. CoBrA maintains a model of current context as *repository of knowledge* shared amongst all the components of the same smart place. The coupling of this shared model with reasoning provides the presence of a user in the smart meeting room. Román et al. [24] come up with a metaoperating GAIA to support the development and execution of portable applications for active spaces. It implements Context File System (CFS) which uses modeling and reasoning by federating application-defined properties and environmental context information to realize active spaces. MobiLife [25] follows similar conceptual framework to realize context aware platform to contact anyone, anytime and anywhere. SPICE [26] demonstrates a tetralayered architecture including a specific layer that they term knowledge layer providing rich set of mechanisms for context acquisition and knowledge processing. Knowledge layer ensures realization of context aware services by making this knowledge available. Open Platform for User-Centric Service

Creation and Execution (OPUCE) [27] emphasizes designing an architecture based on context cycle for integration of communication features in social networking applications and to realize interoperation of Telco-IT applications in a seamless event-driven way. Lamorte et al. [28] demonstrate a similar kind of platform for enabling context awareness in telecommunication services. Baker et al. [13] elaborate the mapping of context life cycle in actual conceptualization of context based situation, perception, decision, actuation, and adaptation process. They further discuss the role of context awareness and CASs based on this hierarchy in future Internet to realize the intelligent society and to address its implications. Barrenechea et al. [29] come forward with a context aware and adaptive approach in distributed event-based systems to model and implement context aware proactive applications involving the combination of context and distributed events. Perera et al. [19] present an IoT paradigm-based detailed analysis of CASs and underscore that each context aware system follows a sense-decide-actuate cycle.

**2.4. Context Aware Systems and Bioinspired Reflex Arc.** It is intuitive to note that there may exist an analogy between context aware systems and reflex actions that are involuntary and automatic responses of living organisms provoked by a sensory stimulus. Although the idea has started to gain strength, there is no contemporary work to the best of our knowledge that truly tries to emulate natural reflex arc in entirety. A recent research project titled “reflex-tree” [30] tries to implement reflex arc for gas pipeline maintenance in urban environments. The authors present a four-tier hierarchy each being part of the reflex arc. The paper however falls short of elaborating how exactly this reflex arc mimics the natural reflex arc. They do not shed any light on the nature of biological reflex arcs if theirs is *instinctive* or *learned*. A deep insight into various studies [31–33] on human and animals provides converging evidence for the existence of two foremost genera of reflexes in human beings: firstly, inborn or intrinsic reflexes which execute their required functions without underlying foundations of memory or prior experience, and these instinctive reflexes are never learned by the subject consciously or unconsciously, and, secondly, *learned* or *conditioned reflexes*, which carry out their requisite operations on the basis of some preceding experience or on the basis of the information items stored in the nondeclarative or implicit unconscious portion of the memory. This specific portion of implicit memory serves as the *priming memory* in which the store is “primed” through repetition of experiences. This primed store helps them to respond very promptly.

On the basis of a thorough review of above-cited literature, our corresponding intuition about context aware systems as sense-decide-actuate cycle, and associative learning-based conditioned reflex action, we conclude the following:

- (1) Each CAS operates through a sense-decide-actuate cycle in which the actuation once performed by the system against a specific event does not help the system to treat the reappearance of the same event intelligently through *on-the-fly* metacontext.

Consequently, CAS executes the same cycle to gain the same actuation resulting in suboptimal performance leading to corresponding time and energy expenditures.

- (2) In contemporary CAS, there is no such mechanism that supports retention and recollection of actuations performed thus far. This limitation restricts the system to adopt the *self*-associative learning ability and therefore cannot react reflexively in case of recurrence of an event.

In this paper, we propose CRAM, a novel cut-through processing paradigm for contemporary visual context aware systems. CRAM provides a cohesive approach to integrate visual context addition and associative learning-based conditioned reflex action to reduce visual context processing so as to evolve CAS into the autonomic state of minimalist actuation. The reflex action is physically realized through supplementation of a memory module into CAS that gets *primed* every time an event occurs. Both context addition and conditioned reflex implemented as an overlay on CAS result in the reduction of node and network-level image processing, increased network-wide energy efficiency, and delay cutbacks.

### 3. CRAM Architecture

In this section, we present conceptualized behavior based upon context addition and reflex arc that forms the basis of the architecture subsequently presented. The layered implementation of the architecture through tenon-mortise is then given.

**3.1. Conceptualized Behavior.** The envisaged behavior of four layers of CASs is shown in Figure 2.

CRAM provides mobile object (MO) detection, tracking, and recognition mechanism through a distributed context added system. The sensing layer being the first and the lowest layer is comprised of pertinent sensing devices. Sensing process takes place whenever a MO is detected. The sensing process always occurs at every node in entirety and it does not undergo any change over time. However, the behavior of other layers changes over time as per Figure 2.

The processing layer that typically implements context semantics in traditional CASs changes its behavior in CRAM with the incorporation of context addition. When MO is detected at the first node, its context is processed in entirety. As MO moves on to the subsequent node in its trajectory, context processing starts to reduce. It is due to the fact that the actuation performed by each predecessor node serves as a metacontext to its successor. The processing and decision-making gradually reach asymptotic minima.

Over time, the corresponding actuation once performed by the system helps the system to internally evolve by serving as a new context. As the system is exposed to more situations in due course of time including recurrence of events, its context repository is enriched through such retrospective contexts, gradually letting it perform internal actuation through improved introspective contexts. Finally, it reaches

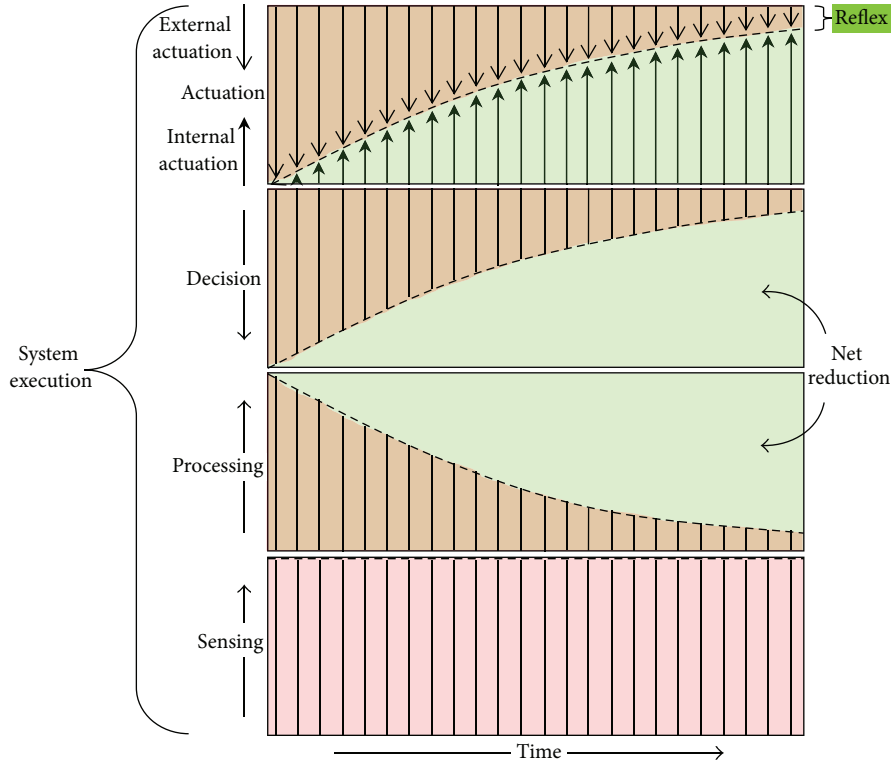


FIGURE 2: Conceptualized behavior of CRAM.

a juncture where a new situation demands minimal external actuation, truly realizing the operation of a *context added system*. When a context added system fully realizes internal actuation, it starts to reflexively respond to recurrence of MO.

**3.2. Tenon-Mortise Architecture.** This section is bipartite. The first part elaborates camera deployment and mobility model that determines the physical topology and resulting mobility pattern of MO in *Region of Interest (RoI)*. The second portion is comprised of layered implementation architecture of CRAM that realizes context addition and reflex arc through cross-layer modules.

**3.2.1. Camera Deployment and Mobility Model.** We formulate the following assumptions for the realization of tenon-mortise layered model:

- (i) The blueprint of RoI is predefined in which each external or edge node (EN) is equipped with sonar and camera while each inner node (IN) is provided with camera only.
- (ii) All SNs have preprogrammed locations in RoI such that each node is aware of its location and the relative locations of each of its one-hop neighbors.
- (iii) All internal nodes have the same computational and memory resources. The external nodes have been provided with an additional memory module that serves as priming (imitation).

TABLE 1: EN states.

State	Sonar	Timer	Camera	Transceiver
Active	1	1	1	1
Quasi sleep	1	0	0	0
Sleep	0	0	0	0

TABLE 2: IN states.

State	Timer	Camera	Transceiver
Active	1	1	1
Demin sleep	0	0	1

(iv) ENs exhibit three states based upon energy consumption with regard to sonar, camera, timer, and transceiver operations as shown in Table 1, while INs exhibit two states of activation with respect to camera, timer, and transceiver as depicted in Table 2.

(v) The fields of view of EN sonar and camera are calibrated to be exactly the same.

Since we present a mobile object detection, tracking, and recognition system, it is important to lay out the physical deployment of sonars and cameras and consequent mobility considerations. As MO mobility is constrained to the road segments only (Figure 3), we find Manhattan mobility model as the most suitable one for MO in RoI.

CRAM is comprised of two sensor node stratum. The outer stratum consists of two layers of ENs. The twofold

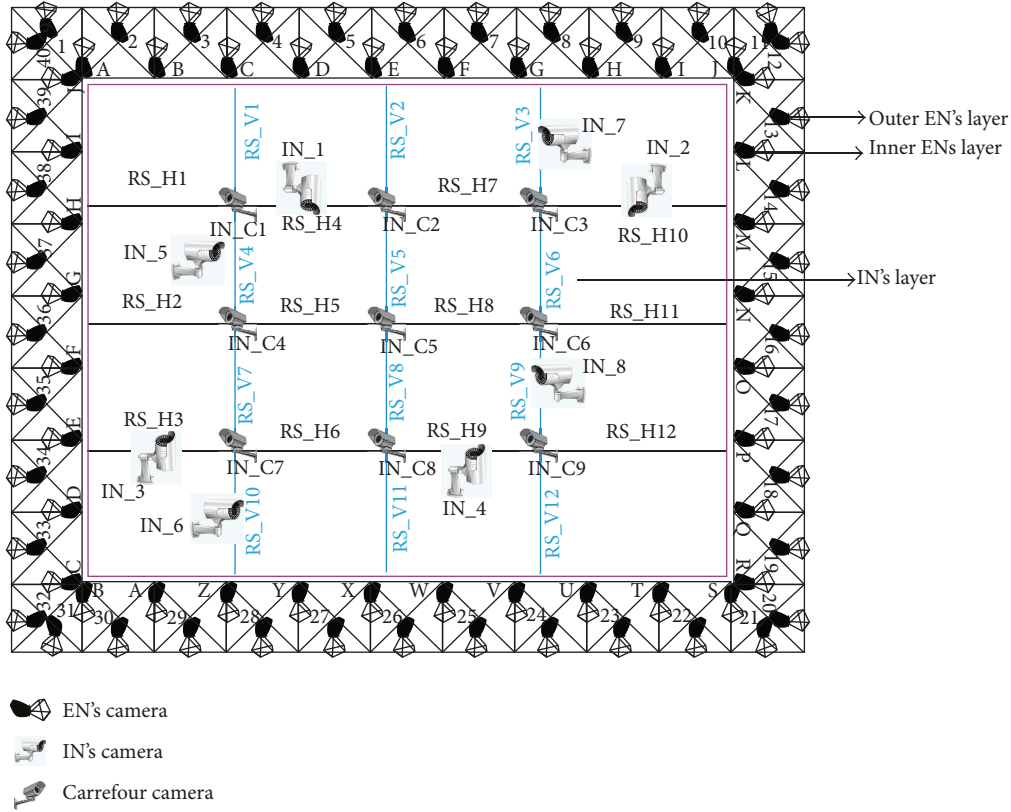


FIGURE 3: Camera deployment and MO mobility pattern.

edge node hierarchy is deployed to assure the detection and recognition of intruding MO in a reliable and energy-efficient manner. The operation of sonars in sets of duo with overlapping (FoV) is presented to ensure coverage while the operation of sonars in triplets achieves energy efficiency.

*Considerations for Camera Deployment of ENs.* For the length of each side of exterior layer as  $L_1$ , the total number of nodes  $N_1$ , each with Field of View (FoV) of width  $W_1$ , required to cover is given by  $N_1 = L_1/W_1$  and the total number of ENs required to secure the complete perimeter =  $4N_1$ .

Similarly, for interior layer,  $N_2 = L_2/W_2$ . The total number of ENs required to secure the complete perimeter =  $4N_2$ .

It is obvious that FoVs of  $W_2 = 2W_1 \Rightarrow N_2 = N_1 - 1 \Rightarrow N_1 = N_2 + 1$ .

*Considerations for Camera Deployments of INs.* In order to detect and track MOs that follow Manhattan model, two types of INs are deployed. The first type of nodes are cameras nodes which are deployed in such a way that each camera covers “ $n$ ” number of horizontal road segments and “ $m$ ” number of vertical road segments in RoI. For instance, as in Figure 3, visual coverage of IN\_1 is three horizontal road segments RS\_H4, RS\_H5, and RS\_H6. Similarly, IN\_8 provides visual coverage of RS\_V7, RS\_V8, and RS\_V9. These road segments all together form rows and columns across RoI as a grid-like structure. The total number of INs required to provide complete RoI coverage is given as  $(IN)_T$ :

$$(IN)_T = (R - 1) + (C - 1), \quad (1)$$

where  $R$  is the total number of rows and  $C$  is the total number of columns.

The second type of INs is Carrefour nodes that are deployed at the Carrefour (Carrefour is French word for intersection). Carrefour nodes provide coverage for MOs approaching or departing an intersection of four road segments. The total number of Carrefour cameras required to cover whole RoI can be given as

$$(IN)_c = (R - 2)(C - 2). \quad (2)$$

*3.2.2. The Layered Architecture.* We present tenon-mortise modules as the realization of the implementation architecture of CRAM based upon cross-layered approach as shown in Figure 4.

Physical layer deals with necessary hardware infrastructure and provides signaling information to the network and processing layers using sonar and timer and shuttered-in frames through cameras. The network layer is just above the physical layer. It implements sonar and timer modules to manage sensing, sleeping, and synchronization operations. It also manages the transceiver through routing decision modules. Memory management module at the network layer allows the management of memory either at the same SN or amongst a set of SNs. Image processing layer, being the highest layer, performs image detection, recognition,

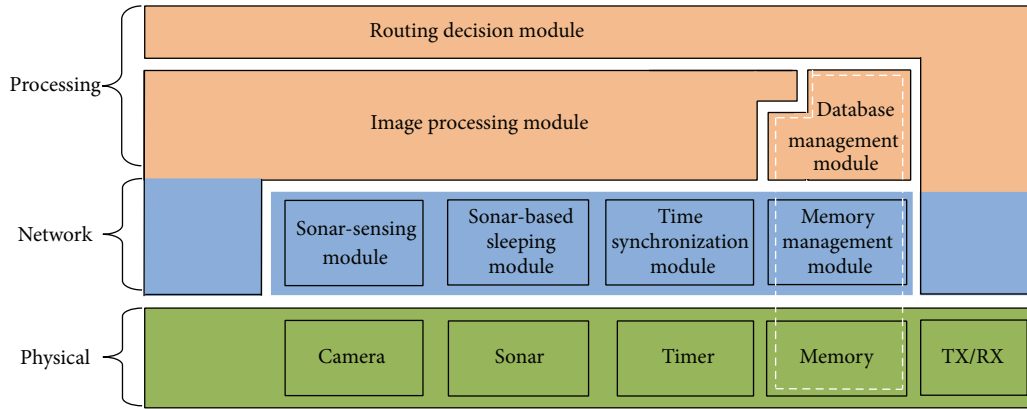


FIGURE 4: Tenon-mortise layered model.

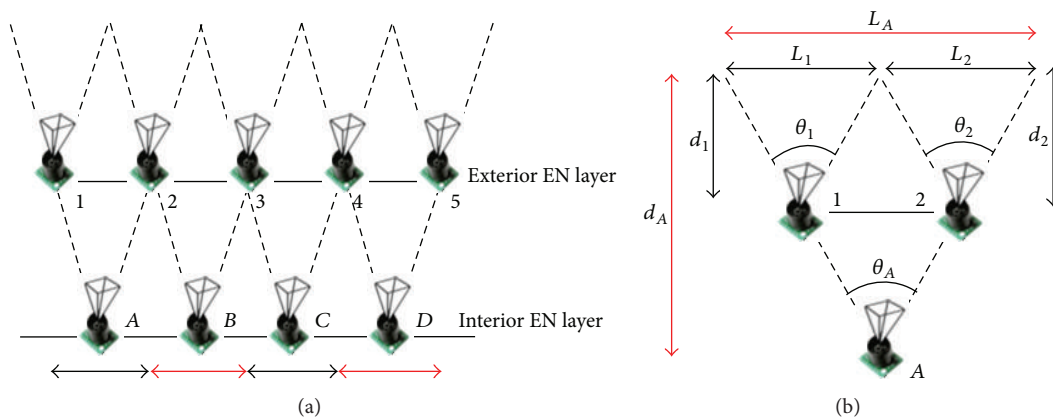


FIGURE 5: (a) Double layer sonar deployment. (b) Triplet formation of sonars.

and tracking through an interplay of database management, image processing, and routing decision operations.

*Physical Layer.* The physical deployment of hardware devices such as sonar, timer, camera, and transceiver is defined at physical layer:

- (i) ENs monitor the existence of an intruding MO through *sonar*. Consequently, cameras are only activated after sonars have detected MO.
- (ii) *Timers* are used to disseminate timing information for synchronization of image acquisition activity at neighboring SNs through in-time activation of next expected nodes.
- (iii) The key function of *camera* is to capture the image of an intruding MO into RoI. The camera is turned on just at the arrival of MO in SNs and is turned off soon after the image acquisition for conserving energy. A multitude of image processing algorithms are then applied.
- (iv) *Memory* provides workspace and storage capabilities. Nonvolatile memory is utilized to store and update topology tables, silhouette tables, and other types of prestored information. Volatile memory provides

run-time environment to perform more intelligence-intensive operations at the upper layers.

- (v) *Transceivers* are used for the transmission and reception of packetized information messages from upper layers.

*Network Layer.* The network layer is comprised of sonar-sensing, sonar-based sleeping, time synchronization, and memory management modules. Memory management module manages the CRAM memory hierarchy with a prime focus to deal with the explicit and implicit memory items. This module ensures the placement of image extracted during the run-time processing of MO in the priming memory that further plays the main role in realization of reflex action (Section 4).

*Sonar-sleeping module* is incorporated with focus on the fact that CRAM assumes to have sonars only at ENs. It becomes improbable to detect MO again in RoI if sonars fail or laxly do so. In order to provide stronger means of detection, two layers of sonars are deployed as shown in Figure 5(a). ENs in both exterior and interior layers are positioned in such a way that one background node resides behind and between two foreground ENs forming a triplet such that FoV of an inner EN is double that of an outer EN (Figure 5(b)).

TABLE 3: CRAM failure resilience through triplet formation.

Malfunctioning triplet	Triplet nodes	Failing nodes	Alternatively working nodes
1	1, 2, A	1 or 2 or 1 and 2 or A	2, A or 1, A or A or 1, 2
2	2, 3, B	2 or 3 or 2 and 3 or B	3, B or 2, B or B or 2, 3
3	3, 4, C	3 or 4 or 3 and 4 or C	4, C or 3, C or C or 3, 4
4	4, 5, D	4 or 5 or 4 and 5 or D	5, D or 4, D or D or 4, 5
5	5, 6, E	5 or 6 or 5 and 6 or E	6, E or 5, E or E or 5, 6

Such triplet formation results in improvements in fault tolerance and failure resilience. For example, when one triplet fails, two neighboring triplets automatically provide alternate coverage. In order to ensure this coverage, intersonar distances of background ENs follow the following well-known solid angle relationship:

$$\begin{aligned} L_1 &= d_1 \theta_1, \\ L_2 &= d_2 \theta_2, \\ L_A &= d_A \theta_A. \end{aligned} \quad (3)$$

As  $L_A = 2L_1$  or  $L_A = 2L_2$ ,  $d_A = 2d_1$  or  $d_A = 2d_2$  such that when IN is on, its coverage should be equal to the coverage of two ENs.

For power consumption analysis, we suppose that the number of ENs at each layer of outer layer is  $n$  (as for very large networks  $n = n + 1$ ). If  $p$  is the amount of power consumed by each node, the average power consumed by outer stratum exterior layer is  $P_{\text{Ext}} = np$ .

Since in each triplet one background node operates against two foreground nodes, the total number of ENs at interior layer for corresponding  $n$  nodes of exterior layer in triplets must be  $n/2$ . Now if  $p$  is the amount of power consumed by each node, then the total amount of power consumed by all nodes of triplets at interior layer can be given as  $P_{\text{Int}} = (n/2)p$ .

If the distance between inner ENs is doubled (requisite for a triplet), the power consumption becomes fourfold. Then, the above relation can be modified as  $P_{\text{Int}} = (n/2)4p \Rightarrow P_{\text{Int}} = 2np$ . And, finally, if the average power consumed by two layers in triplet form is  $P_{\text{Avg}}$ , then it can be demonstrated by the following relation:  $P_{\text{Avg}} = (P_{\text{Ext}} + P_{\text{Int}})/2 \Rightarrow P_{\text{Avg}} = (np + 2np)/2$ .

Hence,

$$P_{\text{Avg}} = 1.5np. \quad (4)$$

So the power consumption difference between nontriplet format and triplet format is  $0.5np$ . Clearly, the triplet formation bears a power tax of  $0.5np$ , but this power added levy provides us with almost complete border breach avoidance system [34] in which the system performs equally well even if 50% of nodes fail. Table 3 shows failure resilience of CRAM through triplet formation.

The power expenditure at ENs can be efficiently managed through a distributed sleep scheduling among the foreground and background nodes of the triplet in such a way that if a foreground EN fails, the background node can be activated

instead. Vice versa, two foreground ENs can be activated in case of background EN malfunction. The sleeping schedule among the foreground and background nodes is shared through SMAC protocol [35] in which background nodes play the role of synchronizers and foreground ones play the role of followers in each triplet.

*Sonar-sensing module* identifies the entrance of the MO in the RoI after receiving Mobile Object Detected (MOD) message from sonar. At the deployment time, distance-based fingerprinting of reflected *Received Signal Strength Indicator* (RSSI) is computed and stored as  $\text{RSSI}_{\text{THRESHOLD}}$  at each EN. The EN at the boundary of RoI periodically sends out beacons to sense possible presence of MO. An EN responds to detection only when the measurement of received RSSI is greater than that of  $\text{RSSI}_{\text{THRESHOLD}}$ . Upon reception of an echo to the beacon at exterior layer EN, the duos of foreground ENs, on the basis of the received RSSI on both of the duo nodes, communicate with each other for the selection of an apt EN to kick off the MO detection job. If the signal is received by the background EN of the triplet, it proceeds with the task of MO recognition itself. For assuring that MO is present and is detected in RoI, we propose that three readings must be taken and analyzed according to Table 4 before camera activation takes place.

Unnecessary camera activation is avoided by anticipating MO trajectory in the network which leads to network longevity. For instance, if a mobile object approaches a sonar and then turns back or moves away from its defined trajectory, no camera is activated. This module, after sensing the existence of any MO in the RoI, generates MOD message to activate the next hop IN.

*Time Synchronization Module.* Each IN sends MO image related information to next hop IN with an associated local time of its acquisition which provides time reference for synchronization [35]. Such synchronization requires accuracy and timeliness so that other INs are activated only and definitely when MO is in this proximity. In order to realize tightly coupled synchronization of INs timer clocks, we present *Camera-Activation Delay Avoidance Time Synchronization* (CADETS) scheme in the time synchronization module (Figure 6) which is tailored to the unique sequence of camera activation and image processing. Here, synchronization activity is initiated in the relevant section of RoI upon the detection of MO by the ENs through beacons to successive INs to update the time information. This time information is further used by INs as reference to synchronize their time clocks and those of subsequent INs in their radio ranges well



TABLE 4: Object movement and EN response.

Sonar reading	MO movement	EN response
$r_1 = r_2 = r_3 = 0$	MO not present	No MO detection by SN
$r_1 = r_2 > r_3$	MO present but moving away	MO detected but not recognized
$r_1 > r_2 > r_3$	MO present but moving away	No MO recognition
$r_1 < r_2 > r_3$	MO present, coming close, and now moving away	No MO recognition
$r_1 = r_2 = r_3$	MO present but standing still	MO recognition is performed at that SN
$r_1 < r_2 = r_3$	Came close and stopped	MO recognition is performed at that SN
$r_1 > r_2 < r_3$	Came close, went away, and came close	MO recognition is performed at that SN
$r_1 < r_2 < r_3$	Moving towards and entering RoI	MO recognition performed all along its trajectory
$r_1 = r_2 < r_3$	Moving towards RoI	MO recognition performed all along its trajectory

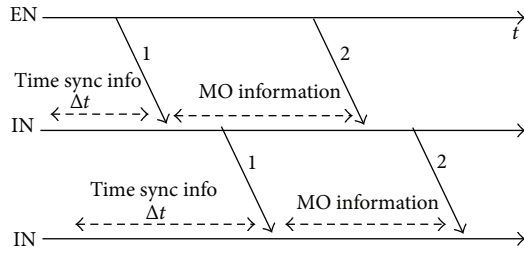


FIGURE 6: Time synchronization through CADETS scheme.

before image processing module disseminates upper layer messages.

*Memory management module* particularly manages the intercommunication of memory module and database management module. One of the main features of this module is to administer the memory module that consists of declarative memory in the form of  $RSSI_{THRESHOLD}$ , topology table, silhouette table, and the compressed images that resulted through SICS. The memory module is also included with nondeclarative memory (priming memory) in the form of priming table comprising aspect ratios of MOs in totality. Memory management module detailed working has been discussed in Section 4.

*Processing Layer.* It contains and executes the logical and algorithmic segment of CRAM to deal with the task of object tracking and identification. It consists of database management module, image processing module, and routing decision module. The operational contribution of each of these modules is elaborated below.

*Database Management Module.* A distributed database based on the regional aspects, application constraints, and requirements to correctly detect, track, and recognize the MO has been deployed over entire RoI.

*Database Organization.* Consider the following:

- (i) Each EN maintains a database record in its memory called topology table to store the neighbor SN's positions and their camera orientations. This table provides the key support in perspective-based mobile object tracking as shown in Table 5.

TABLE 5: SN positions and camera orientations.

SN_ID	SN position (cell, column/row)	SN camera orientation (direction)
IN_1	(II, I)	W
IN_2	(1, II)	E
IN_3	(12, III)	E

- (ii) Each SN maintains another table in its declarative memory called silhouette table. This table contains silhouettes of probable MOs with their respective identifiers (IDs), silhouettes aspect ratios with respect to their segments, classes assigned to these silhouettes, octets, views, angles, and sureties defined against these stored silhouettes as shown in Table 6. In CRAM, two classes have been assigned to the stored silhouettes. Class 1 represents vehicles while class 2 corresponds to humans. Each silhouette is segmented on the basis of its evident number of prominent visual features and then aspect ratios are computed for each segment of silhouette. This calculation has been discussed in detail at the end of this section. Octet is an ASCII code assigned to a recognized MO. Angle is the angle of camera with respect to MO while the view is the angle of MO with respect to camera. Surety represents the percentage match of acquired image with the stored silhouette and for each stored silhouette it can be given by the following relation:

Silhouette surety at observing SN

$$= \begin{cases} 0 & \text{if no silhouette is present} \\ \frac{1}{k} \times 100 & \text{otherwise,} \end{cases} \quad (5)$$

where  $k$  is the total number of silhouettes of a single object with different aspects distributed throughout the RoI.

- (iii) The static background image of the stored silhouettes is also stored in the database for the further use in the image processing procedures.

*Database Deployment.* Consider the following:

- (i) Topology table deployment: For successful tracking of MO, each SN is equipped with a topology table which

TABLE 6: Silhouette table.




ID	Class	Silhouette	Silhouette aspect ratios with respect to segments		Octet	View	Angle	Surety
			Total segments	Aspect ratios				
1	0 (vehicles)		3	(Segment ID, aspect ratio)	Car	Front	0	12.5%
2			3	(Segment ID, aspect ratio)	Bus	Back	180	12.5%
3	1 (humans)		4	(Segment ID, aspect ratio)	Human	Front	90	25%

TABLE 7: Topology table for IN\_4.

SN_ID	SN position (cell, column/row)	SN camera orientation (direction)
IN_3	(12, III)	E
IN_5	(15, V)	W
IN_6	(10, VI)	E
IN_7	(18, VII)	S
IN_8	(19, VIII)	N
IN_9	(7, IX)	W

contains all neighboring SN's positions and orientations. Table 7 shows the deployment of topology table at IN\_4.

- (ii) Deployment of silhouette table: Storage of silhouette table at each SN depends primarily on its locality. For instance, IN\_1, IN\_2, and IN\_5 are expected to cover a solo side of the MO, so one silhouette of requisite dimension for each expected MO is enough to be stored in its silhouette table as shown through Table 8 for IN\_2. Conversely, as a Carrefour camera is positioned to acquire image through its multiple (sides, front, back, and tilted views) aspects, its table will be provided with all possible dimensions of probable MO silhouettes as shown in Table 9 for IN\_9.
- (iii) It has been erudite that the number of silhouettes in entire network rises with the increase in RoI if the SNs are mounted at a constant distance from each other while if deployed at variable distance, the total number of silhouettes depends upon the number of SNs deployed in RoI.
- (iv) Silhouette table is affected by few factors in which *camera orientation* becomes at primary level. Each SN is stored with the silhouettes which have the higher matching probability with acquired MO silhouettes as for IN\_2 side views of MO have the highest probability to match. Secondly, with the increase in type of MO to be passed through RoI, silhouette table amplifies in its size. Thirdly, in case Gauss's Markov mobility model is used, it will require storing silhouettes of all possible aspects over the entire network and correspondingly

the surety depends upon the number of silhouettes stored for a single object as demonstrated in Table 10.

*Silhouette Segmentation and Aspect Ratio Calculation.* The silhouettes of multiple MOs with the highest matching probabilities are stored in SNs that are used to identify MO by matching them with run-timely acquired ones. Silhouette segmentation process is based on the total number of prominent features; for example, the front view of a human is segmented with five prominent features as head, neck, shoulders, torso, and lower limbs. Figure 7(a) illustrates the different segmented views of a human. Similarly, Figure 7(b) shows the prominent features based segmented images of hatchback and saloon cars with different views.

The computation of aspect ratios is carried out by taking the width-to-height ratios of segmented parts. For instance, Figure 8 explicates the segmentation of human in five parts based on prominent features and calculation of aspect ratios stored in the database table.

*Image Processing Module.* IP module is initiated directly by sonar interruption at EN or upon reception of MOD message at any SN. Being the central part of the CRAM, IP module plays the key role in image capturing and processing. It captures the MO instantaneous image, processes it through different image processing algorithms to convert it into MO silhouette, matches this extracted silhouette with the stored one, recognizes the MO, and presents its outcomes in the form of percentage surety. It operationally proceeds with the following assumptions:

- (i) The prestored and extracted silhouettes are of identical scales.
- (ii) The background subtraction algorithm is restricted to be applied only in the condition when the distance between road segments and the SNs remains the same.

Image processing module contains some submodules whose role is elaborated as follows.

*Image capturing submodule* is responsible for acquisition of instantaneous image. Upon reception of MOD message, the corresponding SN triggers its camera for  $\Delta t$  time and captures fixed size MO image. It is of the fundamental concern that in presence of variance in MO arrival time  $\Delta t$  affects the total "Shutter ON" time. The image acquisition frequency

TABLE 8: Silhouette table of IN\_2.

















ID	Class	Silhouette	Aspect ratios with respect to segments		Octet	View	Angle	Surety
			Segments	Aspect ratios				
1	0		3	(Segment ID, aspect ratio)	Bus	Left side	90	12.5%
2			3	(Segment ID, aspect ratio)	Tank	Left side	90	12.5%
3			4	(Segment ID, aspect ratio)	Car	Right side	270	25%
4			4	(Segment ID, aspect ratio)	Jeep	Right side	270	25%
5	1		4	(Segment ID, aspect ratio)	Human	Left side	90	25%

TABLE 9: Silhouette table of IN\_3.






ID	Class	Silhouette	Silhouette aspect ratios with respect to segments		Octet	View	Angle	Surety
			Total segments	Aspect ratios				
1	0 (vehicles)		3	(Segment ID, aspect ratio)	Car	Front	0	12.5%
2			3	(Segment ID, aspect ratio)	Car	Right side	270	12.5%
3			3	(Segment ID, aspect ratio)	Car	Back	180	12.5%
4			2	(Segment ID, aspect ratio)	Bus	Left side	90	12.5%
5			3	(Segment ID, aspect ratio)	Bus	Back	180	12.5%
6			3	(Segment ID, aspect ratio)	Jeep	Right	270	12.5%
7			3	(Segment ID, aspect ratio)	Jeep	Front	0	12.5%
8			3	(Segment ID, aspect ratio)	Car	Tilt	315	12.5%
9			3	(Segment ID, aspect ratio)	Car	Tilt	225	12.5%
10	1 (humans)		3	(Segment ID, aspect ratio)	Man	Left side	90	25%
11			3	(Segment ID, aspect ratio)	Man	Front	0	25%

must be 25 frames per second in vehicle traffic areas [36]. Out of  $n$  frames captured by SN, every  $i$ th frame is processed for MO recognition and tracking. Figure 9 demonstrates the timeline of SNs sequential camera activation for  $\Delta t$  seconds.

After acquisition by image capturing submodule, the captured image is processed through the *image change detection module* to find any change in prestored image with static background to detect the presence of MO. We use Gaussian Mixture Model (GMM) for this change detection for given number of Gaussian components for “evolving” background. The process of change detection can be influenced by

a number of factors at each node which include swaying background objects, slow moving foregrounds, and shadowing or illuminating of light sources with their localized distinctiveness. This emphasizes the adaptation of the image change detection module on each IN to become more sensitive in case of activation of localized background process and less sensitive otherwise. More sensitivity leads to counter the effects of active backgrounds processes through higher number of Gaussian components in mixtures of Gaussians and through correspondingly lesser number when background is more stationary. However, the utilization of large number

TABLE 10: Silhouette table with different aspect ratios stored in database.

ID	Class	Silhouette	Aspect ratios with respect to segments		Octet	View	Angle	Surety
			Segments	Aspect ratios				
1			4	(Segment ID, aspect ratio)	Car	Front/left	45	Depends on stored silhouettes
2			1	(Segment ID, aspect ratio)	Car	Left side/top	90	
3	0		3	(Segment ID, aspect ratio)	Jeep	Left side/tilt	90	
4			3	(Segment ID, aspect ratio)	Car	Right side/back	225	
5			3	(Segment ID, aspect ratio)	Car	Right side/tilt	270	

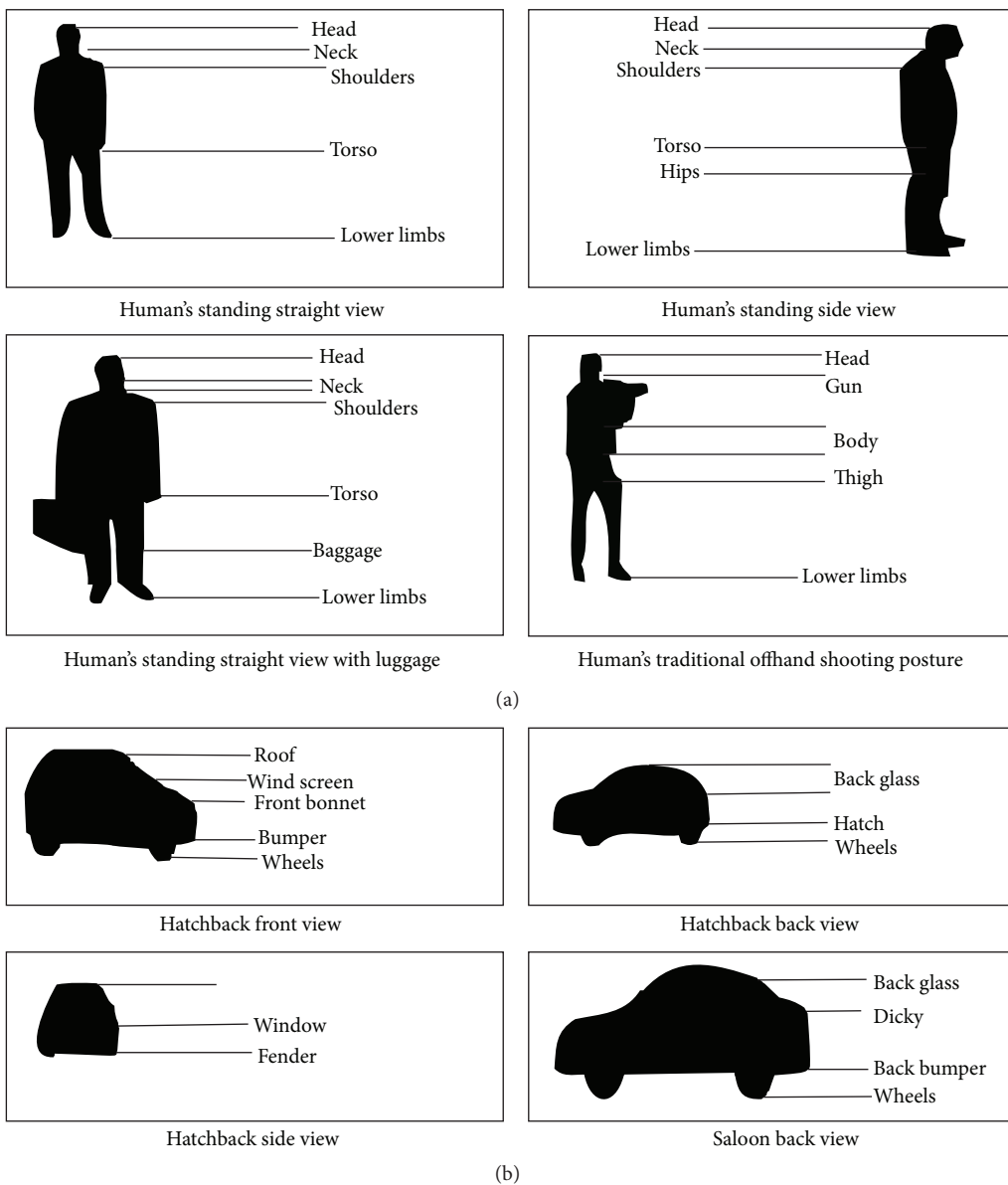


FIGURE 7: (a) Human segmentation with different views based on prominent features. (b) Car segmentation with different views based on prominent features.

TABLE 11: SICS surety based transcoded image storage and corresponding power consumption analysis.

Surety level (%)	Image transcoding level	Image quality level (%)	Power consumption in transcoding operation (mW)
0–25	Level 0	100	Nil
26–50	Level 1	75	68.8
51–75	Level 2	50	44.1
76–99	Level 3	25	24.7

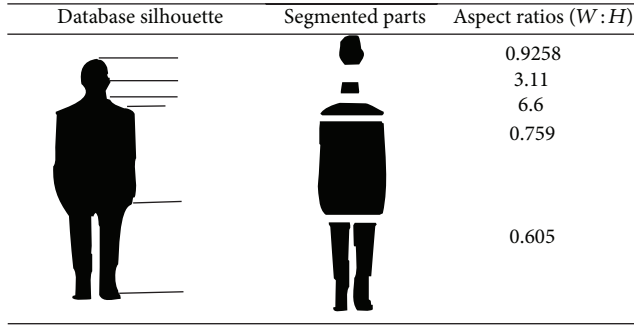


FIGURE 8: Aspect ratios calculation of human silhouette.

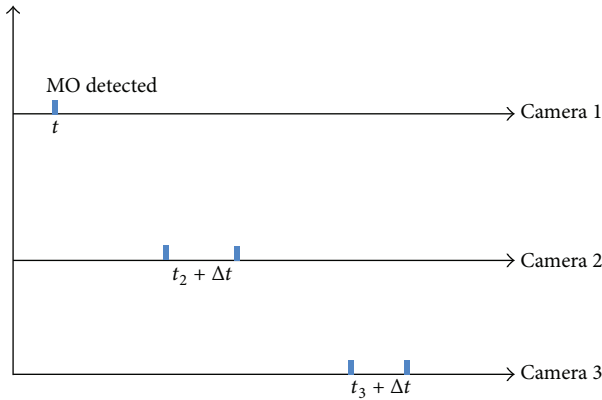


FIGURE 9: Timeline of camera activation.

of Gaussian components is not suggested in CRAM because of higher quantity of consequential energy drain with added complexity.

For optimization of image change detection, we propose an adaptive change detection scheme which is driven by a feedback loop bridged between an up-trajectory and down-trajectory node. The up-trajectory node refers to an EN which activates an IN or an IN which activates another IN while down-trajectory node stands for an IN which is activated by an IN or EN. In the operational execution of this scheme, when an object is sensed by an up-trajectory node, it forwards a MOD message to its down-trajectory neighbor expecting that its counterpart shall detect it. On successful detection, the down-trajectory node replies with positive feedback implying that the change detection at up-trajectory node is sufficiently sensitive or adequate number of Gaussian components are being used at it. In a situation where the down-trajectory node does not detect the MO, it responds

negatively signifying the fact that up-trajectory node is not aptly sensitive or a higher number of Gaussian components are needed to be fuelled. These false positives can also be triggered by malfunctioning of an IN or Carrefour node and can be alleviated by practicing a consensus and voting algorithms among a group of neighboring inner nodes-based hysteresis loop which intertwines recursively in whole network to avoid such malfunctions [37].

The image captured by image capturing module is refined and fine-tuned by image change detection module and is compressed and stored by *Image Compression and Storage Module*. In order to optimize the compression and storage, we use Quality-Aware Transcoding [38] which offers a quality-versus-size trade-off based scheme for dynamically changing the image size. We further propose a compression and storage scheme in which the image is compressed and stored based on its corresponding MO's surety level named as *Surety Based Image Compression and Storage (SICS)*. In this scheme, as a MO attains more surety levels, its image is transcoded to more elevated levels and is stored at lower image quality levels to diminish the energy expenditure as the power consumption of transcoding operation decreases with decline in image quality level. We examine that as the low-quality stored images are sufficient enough for further image processing and MO recognition, the MO recognition process is not manipulated by this low-quality image storage. Moreover, these low-quality stored images occupy less space in the memory which ultimately consumes less computational power for further image processing and MO recognition processes. We justify the proposed method through Table 11.

When a MO penetrates into the RoI for the first time, its image is not transcoded as per the suggestions of SICS and is stored with its original size for better identification. Subsequently, when it passes through more SNs hops (IP) in RoI, it consequently acquires elevated surety levels. At a situation when the MO attains surety level more than 25%, it is assigned with the corresponding transcoding level. For instance, at 50% surety, it achieves transcoding level 1, at 75% surely, it is transcoded at level 2, and when it reaches more than 75% surety, it gets transcoded level 3 and is stored at 75, 50, and 25 percent image quality levels, respectively. The entire power expenditure through image capturing, compression, and storage modules can be given as

$$P_T = P_A + P_C + P_S, \quad (6)$$

where  $P_T$  is the total power consumed while  $P_A$ ,  $P_C$ , and  $P_S$  are powers consumed by image capturing, compression, and storage modules, respectively. Each of these constituents

of power consumption is independently determined by the algorithms used below.

*Image subtraction submodule* is used to extract the MO silhouette by subtracting the silhouette from its image with static background stored at SN through background subtraction submodule. Background subtraction can be optimized in terms of power and time cost by applying Don't Care operation on background image without affecting silhouette extraction. In order to apply Don't Care operation on selected parts of background image, silhouette information of acquired-and-then-stored image is required. The changed and unchanged regions can be then detected by applying the following equation as suggested by Xu et al. [39]:

$$Dx_{ij}^k = x_{ij}^k(t_2) - x_{ij}^k(t_1). \quad (7)$$

The pixel-by-pixel change detection process is executed once at first time entry of the MO in RoI while the change is detected on successive hops. As RoI is based on Manhattan model, the possible division of background image in four portions is proposed in which few are Don't Cared and few are used for background subtraction. It is found that there possibly exist four cases which include the variation of change detection regions and positions of changed regions as shown in Table 12. Based on the total number of changed regions and their positions, we assign a distinctive code to each possible combination which is further used to transfer background subtraction information to neighbor SNs. The background subtraction operation is applied to the one, two, or all parts at all SNs in which change is detected. We also suggest ID-based split background image subtraction which applies the background subtraction operation on some portions of background image on the basis of ID of previous SN. These portions are selected on the basis of scene entry region information where scene entry information refers to the mobile object navigation on last SN where the MO is lastly seen and provides the entry direction of MO at a SN. SNs store this scene entry region information in their topology tables with reference of their neighbor SNs IDs. We present the IN\_4 from Figure 3 as an example to demonstrate the adequate section of background image approaching it from neighbor SNs while supposing that the change is detected in one portion so the subtraction shall take place at corresponding single portion (Table 13).

Further analysis shows that the total time consumed during the image subtraction of one portion is 4.5 times less than the time it takes to subtract whole image.

*Silhouette comparator submodule* compares the finally extracted silhouette with prestored silhouettes in the silhouette table. In order to optimize energy utilization, we present feature-dependent silhouette segmentation (FRILL) procedure. In this segmentation technique, the acquired silhouette is segmented in correspondence with the stored silhouette such that both possess absolutely identical and the same number of segments. Further, the aspect ratio of each segment of extracted silhouette is computed and is compared with the aspect ratios of corresponding segments

TABLE 12: Total possible changed regions with their positions.

Case number	Total number of changed regions	Code number	Changed region
1	4	0000	
2.1	1	0001	
2.2	1	0010	
2.3	1	0011	
2.4	1	0100	
3.1	2	0101	
3.2	2	0110	
3.3	2	0111	
3.4	2	1000	
4.1	3	1001	
4.2	3	1010	
4.3	3	1011	
4.4	3	1100	









of stored silhouette. The silhouette comparison procedure is done through the following expression:

$$\text{Diff} = \text{Min} \left( \sum_j AR_e^i - AR_s^i \right) \forall_k, \quad (8)$$

where  $AR_e$  is aspect ratio of the extracted silhouette,  $AR_s$  is aspect ratio of the stored silhouette,  $j$  is total number of silhouette segments, and  $k$  is total number of silhouettes deployed on a SN.

Intuitively, this relation works out the extracted and stored silhouettes' aspect ratio difference one by one and then returns the least difference value. The stored silhouette with which it shows minimum difference is declared similar to

TABLE 13: Split background table of IN\_4 based on previous SN.ID.

Previous Node_id	Scene entry region information	Background section
IN_3	Northwest	
IN_5	Southeast	
IN_6	Northeast	
IN_7	Southeast	
IN_8	Southwest	
IN_9	Northwest	
IN_10	Northeast	
IN_11	Northwest	

extracted one while the extent of similitude in percentage is calculated through the following relationship:

$$\text{percentage match} = \frac{\sum_{i=0}^{\text{total segments}} (\text{AR of extracted silhouette})}{\sum_{i=0}^{\text{total segments}} (\text{AR of stored silhouette})} \left(\frac{1}{k}\right) \quad (9)$$

$$\times 100.$$

The proposed mechanism is presented with an exemplar scenario in Figure 10. The silhouette of an unknown MO is extracted and matched up with four stored silhouettes of human with dissimilar views. The run timely harvested silhouette is segmented in the same correspondence with stored ones in the database. For instance, when this is matched up with standing man's side view, it is segmented in six parts and consequently in five parts while being compared with man's armed view with the same width and height ratios.

After getting done with the matching up task, a packet is generated by IP module in which all the fields are extracted from database except surety as illustrated in Figure 11.

*Routing Decision Module.* This module destines the packets generated by IP modules. The sonar-sleeping, sonar-sensing, time synchronization, and image processing modules are responsible for invoking it. Table 14 shows a variety of decisions this module can take.

When the routing decision module is summoned by sonar-sleeping module, it disseminates the background EN's

TABLE 14: Decision table of routing and decision module.

External module	Packet type	Destination decision
Sonar-sleeping module	Sleeping schedules	Send to ENs in a triplet
Sonar-sensing module	Sleeping schedule	Send to next hop only
Time synchronization module	Beacon/time stamps	Send to subsequent node only
Image processing module	MO recognition information	Send to next hop SN/base station

generated sleeping schedules with destined foreground ENs in a triplet. It further decides about the sonar-sleeping module generated MOD destination where the MOD calls up the camera of most optimal SN for timely acquisition of image frame and MO recognition.

#### 4. CRAM Memory Management and Processing

In this section, we present a mapping between human memory structure and CRAM memory composition. The humans are in-built with a three-stage memory formation as illustrated in Figure 12. The first stage of memory in human anatomy that interacts with the outside world through the sensory receptors is sensory memory.

In CRAM, analogous to haptic and iconic sensory memory, we use sonar as an electromechanical receptor for sensing of a MO in RoI. As discussed in sonar-sensing module (Section 3.2.2), sonar sends beacons continuously in the requisite region but reacts only when the MO presence is detected on the basis of the  $RSSI_{\text{THRESHOLD}}$  information to either commit the event for further processing or pay no heed otherwise.

Short term memory is the second stage of human memory structure which holds items that are of further interest extracted from sensory memory. CRAM uses similar approach for short storage when SICS-based transcoded images are extracted by Image Compression and Storage Module (Section 3.2.2).

For the permanent storage of items, the human anatomy defines long term memory. Long term memory is further subdivided into declarative (explicit) and nondeclarative (implicit) memories. The declarative memory or explicit memory refers to those memories which can be consciously recalled such as facts and knowledge [40]. Declarative memory is further comprised of semantic and episodic memories [41]. Episodic memory is a major constituent of declarative memory that is the collection of previously experienced events with their incidence at particular place, time, coupled emotions, and additional context to figuratively remember the event that took place at certain time and place [42]. Analogous to episodic memory in humans, CRAM uses prestored topology and silhouette tables with their IDs, positions, camera orientations, classes, octets, views, angles,

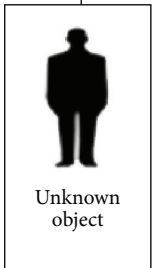

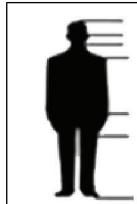


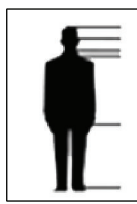

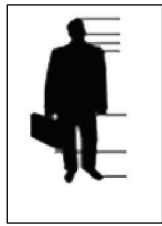
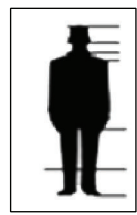



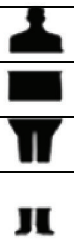
	Database silhouettes	Extracted silhouettes	Segmented parts	H & W of segmented parts	Diff Eq working		
					AR of extracted silhouette	$AR_e - AR_s$	
 <p>Unknown object</p>	 <p>Man's stooping view</p>			Part 1 H = 0.09'' Part 1 W = 0.13''	1.44	1.44 - 2.22 = 0.78	
				Part 2 H = 0.03'' Part 2 W = 0.12''	4	4 - 4.16 = 0.16	
				Part 3 H = 0.09'' Part 3 W = 0.29''	3.2	3.2 - 3.14 = 0.06	
				Part 4 H = 0.29'' Part 4 W = 0.33''	1.1	1.1 - 1.07 = 0.03	
				Part 5 H = 0.12'' Part 5 W = 0.34''	2.83	2.83 - 1.84 = 0.99	
				Part 6 H = 0.32'' Part 6 W = 0.24''	0.75	0.75 - 1 = 0.25	
						$(\sum AR_e^i - AR_s^i) = 2.27$	
	 <p>Man's standing straight view</p>			Part 1 H = 0.1'' Part 1 W = 0.12''	1.2	1.2 - 1.33 = 0.1	
				Part 2 H = 0.05'' Part 2 W = 0.12''	2.4	2.4 - 1.55 = 0.8	
				Part 3 H = 0.24'' Part 3 W = 0.03''	8	8 - 8 = 0	
				Part 4 H = 0.39'' Part 4 W = 0.3''	0.76	0.76 - 0.815 = 0.05	
				Part 5 H = 0.39'' Part 5 W = 0.26''	0.66	0.66 - 0.815 = 0.15	
						$(\sum AR_e^i - AR_s^i) = 1.1$	
	 <p>Man's loaded (baggage) view</p>			Part 1 H = 0.08'' Part 1 W = 0.14''	1.75	1.75 - 1.33 = 0.42	
				Part 2 H = 0.05'' Part 2 W = 0.13''	2.6	2.6 - 1.6 = 0.94	
				Part 3 H = 0.05'' Part 3 W = 0.3''	6	6 - 5.16 = 0.84	
				Part 4 H = 0.37'' Part 4 W = 0.35''	0.94	0.94 - 1.02 = 0.08	
				Part 5 H = 0.21'' Part 5 W = 0.25''	1.19	1.19 - 1.73 = 0.54	
				Part 6 H = 0.15'' Part 6 W = 0.24''	1.6	1.6 - 1.33 = 0.27	
						$(\sum AR_e^i - AR_s^i) = 2.55$	
	 <p>Car's side view</p>			Part 1 H = 0.26'' Part 1 W = 0.31''	1.192	1.192 - 2.04 = 0.848	
				Part 2 H = 0.16'' Part 2 W = 0.31''	1.937	1.937 - 3.93 = 1.99	
				Part 3 H = 0.24'' Part 3 W = 0.31''	1.291	1.291 - 2.8 = 1.5	
				Part 4 H = 0.16'' Part 4 W = 0.22''	1.375	1.375 - 4.3 = 2.9	
					$(\sum AR_e^i - AR_s^i) = 7.19$		
$Diff = \text{Min}_j (\sum AR_e^i - AR_s^i) \forall_k = 1.1$ Result = match with human straight view							

FIGURE 10: Matchup of an unknown MO extracted silhouette with the prestored silhouette in database.



Sequence number	SN_ID	Octet	Angle	Percentage match	Time stamp
-----------------	-------	-------	-------	------------------	------------

FIGURE 11: Resultant MO information packet generated by IP module.

TABLE 15: System specifications for performance evaluation.

Features	Specifications
Matlab (version)	R2012b
Processor	Intel corei5
Memory	4 Gb
Camera	Web camera Logitech Tessar
Processor speed	1.80 GHz
Dimension of the image	240 × 240 pixels
MO traversal (in hops)	12

and sureties. This information is recalled and utilized with timing information as in CADETS through image processing module (Section 3.2.2). The resulting surety and identified class of the extracted image of MO are disseminated for actuation. Concomitant to the operation in the episodic (declarative) part of the memory, aspect ratio of the extracted silhouette becomes the item of interest to be subsequently passed on to and used by nondeclarative memory (implicit).

CRAM exploits the presence of nondeclarative memory in which previous experiences aid the performance of a task *without* conscious awareness of these previous experiences. Such implicit memory is *priming memory* in which the store is “primed” either through repetition of experiences called imitations or the most recent experience and it lets humans respond very promptly [43, 44]. As mentioned in the preceding discussion on episodic memory, CRAM simultaneously primes the nondeclarative memory (priming memory) with the aspect ratio of extracted silhouette. When a new MO is detected, the aspect ratio of its silhouette is compared with the stored aspect ratio of the previous MO in the priming memory. In case of a match, a recognition message is disseminated to the sink without further activating the downstream nodes. Such is the realization of reflex arc that lets the first node observing MO respond while only *referring* to the priming memory as shown in Figure 13.

Since, at the same time, episodic memory is also in the process of computing and comparing aspect ratios of individual segments of the newly detected MO, the results of declarative (episodic) and nondeclarative (priming) memories are evaluated for determining the margin of error. In case of discrepancy, nondeclarative memory processing may be adjusted to yield closer results to those of declarative memory.

## 5. CRAM Performance Evaluation

In this section, we evaluate the performance of CRAM with respect to context addition and reflex arc through prototype implementation and compare its performance with VISTA. It is important to note that an end to end comparison can only exist between VISTA and CRAM because of the common visual context cycle both of them execute. Table 15

TABLE 16: Parameters and situations for experiments.

Parameters	Situations
Lighting effects	Sunlight, artificial light, and shadow
Object types	Human, cars
Angle	Straight, side, and loaded
Background types	Uniform, nonuniform

shows the system specifications and features of the testbed. Table 16 outlines the parameters and situations under which the images were acquired.

*5.1. Reflex Performance of CRAM.* In order to analyze the context added behavior and impulse response of CRAM, a trajectory of 12 hops was implemented for MO. For the first appearance of MO, CRAM demonstrates context addition throughout the trajectory right from the second node, whereas, in case of reappearance of the same MO at the same node, CRAM takes the decision only on the basis of priming memory at the very first node illustrating an obvious imitation of learned reflex action. It is further observed in the particular case of reappearance that CRAM takes 11 times lesser time than VISTA (Table 17). It is due to the fact that VISTA does not declare its recognition results before complete processing of episodic memory for all the nodes in the trajectory.

*5.2. Accuracy of CRAM.* With the implementation of priming memory, it becomes important to be assured on the quality of recognition that reflex arc provides. Table 18 shows that, for multiple iterations of MO recognition, CRAM gave 80% accuracy, the same as VISTA. In both cases, 1 and 4 CRAM erred due to the presence of background objects adding noise. It may also be noted that CRAM performs better than VISTA in poor lighting conditions (case 9) because each preceding process in episodic memory adds its own image processing noise for the succeeding process.

## 6. CRAM Analytical Model

The analysis presented here develops an understanding of context aware systems with regard to the execution of processing and delays incorporated, etc. This section analyzes these aspects with the inclusion of context addition and reflex arc.

*6.1. CRAM Provides Sublinear Bounds on Delay.* In order to verify the time sensitive and delay effective approach of CRAM, we presume a trajectory of  $n$  nodes for a detected MO, where  $N = \{N_1, N_2, \dots, N_n\}$  as shown in Figure 14. Each node  $N_i$  contains a set  $S_i$  whose elements are pieces of silhouette information ranging up to  $m_i$  elements, for example, silhouette, aspect ratio, octets, angles, and surety values.  $A_i$  is subset of  $S_i$  whose elements range from 1 to  $\ell_i$ . Then,

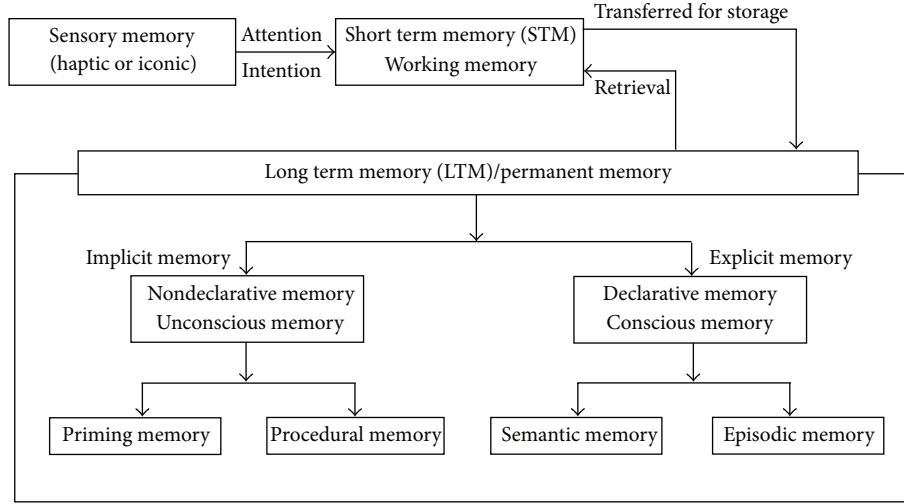


FIGURE 12: Human memory structure.

$$\begin{aligned}
 S_i &= \{1, 2, 3, \dots, m_i\}, \\
 A_i &= \{1, 2, 3, \dots, \ell_i\}, \\
 &\text{where } A_i \subset S_i, \ell_i \leq m_i.
 \end{aligned} \tag{10}$$

*Time Taken at the First Node.* Suppose, at node  $N_1$ , percentage surety level of a detected MO is calculated as a value  $V_1$ . For the calculation of  $V_1$ , the set  $S_1$  of silhouette information at  $N_1$  is traversed for comparison. Consider that a unit time  $t$  is consumed for each comparison.

The best-case time complexity is simply  $t$  if the detected MO matches with silhouette information of  $N_1$  in its very first comparison. It is the realization of reflex arc for recurrent systems where the elements of  $A_i$  are repetitive for successive MO detections.

The average-case time complexity is

$$\frac{1}{m_1} \sum_{i=1}^{m_1} t_i = \frac{m_1 + 1}{2} t, \tag{11}$$

where  $m_1$  is the last object of set  $S_1$ .

The worst-case time complexity of comparison is

$$t \sum_{i=1}^{m_1} 1 = m_1 t \tag{12}$$

if the detected MO matches with silhouette information of  $N_1$  in its very last comparison.

*Time Taken at the Second Node.* At node  $N_2$ , the silhouette information and surety value  $V_1$  calculated by node  $N_1$  is received as a new context according to our context addition model. Therefore, at this stage, the comparison operation does not traverse the entire silhouette information set  $S_2$  of node  $N_2$ . Rather, through new context added from the previous node  $N_1$ , only  $\ell_2$  comparisons are performed in the pertinent silhouette information part  $A_2$ , a subset of the entire set  $S_2$  at node  $N_2$  to yield  $V_2$ .

The best-case time complexity of comparison at this stage is again simply  $t$ , if the detected MO matches with silhouette information of  $N_2$  in its very first comparison.

The average-case time complexity is

$$\frac{1}{\ell_2} \sum_{i=1}^{\ell_2} t_i = \frac{\ell_2 + 1}{2} t, \tag{13}$$

where  $\ell_2$  is the last object of set  $A_2$ .

The worst-case time complexity of comparison is

$$t \sum_{i=1}^{\ell_2} 1 = \ell_2 t \tag{14}$$

if the detected MO matches with silhouette information of  $N_2$  in its very last comparison.

*Total Time Taken till the Second Node.* Concluding both nodes, the best-case time complexity is  $2t$ .

Similarly, the average-case time complexity is

$$\frac{m_1 + 1}{2} t + \frac{\ell_2 + 1}{2} t = \frac{m_1 + \ell_2 + 2}{2} t. \tag{15}$$

The worst-case time complexity is

$$m_1 t + \ell_2 t = (m_1 + \ell_2) t. \tag{16}$$

In rare cases, if the comparison operation yields incorrect result and a wrong object recognition message is sent to the second node, subsequent comparison to a subset  $A_i$  would yield a mismatch. Consequently, the second node once again has to traverse the entire silhouette information set  $S_2$  of node  $N_2$ .

Then, the worst-case time complexity in this situation is

$$\begin{aligned}
 t \sum_{i=1}^{m_1} 1 + t \sum_{i=1}^{\ell_2} 1 + t \sum_{i=1}^{m_2} 1 &= m_1 t + \ell_2 t + m_2 t \\
 &= (m_1 + \ell_2 + m_2) t.
 \end{aligned} \tag{17}$$

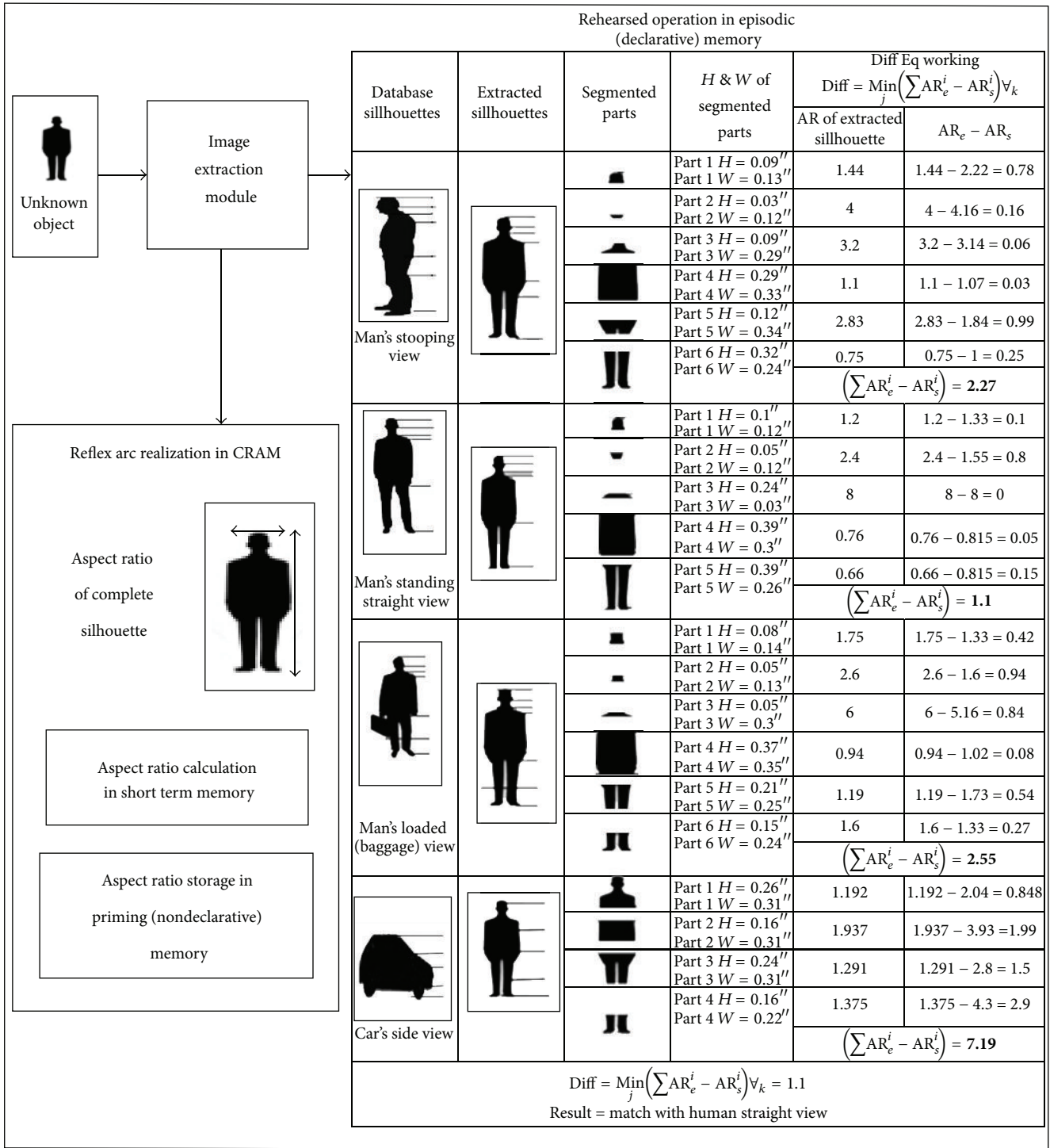


FIGURE 13: Function of episodic memory and realization of reflex through priming memory.

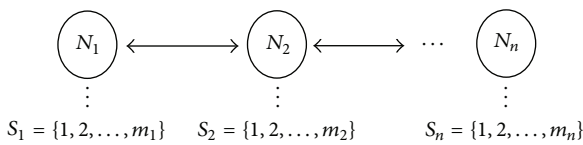


FIGURE 14: Trajectory of the detected MO.

Time Taken till nth Node. Similarly, for traversing  $n$  number of nodes' surety level as  $\sum V_i \geq V_{\text{Threshold}}$ , the time complexity of our model can be deduced through induction as follows.

TABLE 17: CRAM reflex response.

Captured image	Image	Elapsed time		Recognized image	
		VISTA	CRAM	VISTA	CRAM
Human standing		16.87 sec	2.02 sec	Human standing	Human standing
Car front view		24.40 sec	2.05 sec	Car front view	Car front view
Car side view		25.20 sec	2.28 sec	Car front view	Car front view

Best-case time complexity is  $nt$  if the defined silhouette information of all  $N_i$  is matched in  $S_1$  and all  $A_i$  ( $i$  iterates from 2 to  $n$ ) in their very first comparisons.

Average-case time complexity is

$$\begin{aligned} & \frac{m_1 + 1}{2}t + \frac{\ell_2 + 1}{2}t + \dots + \frac{\ell_n + 1}{2}t \\ & = \frac{1}{2} \left( m_1 + n + \sum_{a=\ell_2}^{\ell_n} a \right) t. \end{aligned} \quad (18)$$

Worst-case time is

$$\begin{aligned} & t \sum_{i=1}^{m_1} 1 + t \sum_{j=1}^{\ell_2} 1 + \dots + t \sum_{k=1}^{\ell_n} 1 \\ & = m_1 t + (\ell_2 + \ell_3 + \dots + \ell_n) t = \left( m_1 + \sum_{a=\ell_2}^{\ell_n} a \right) t. \end{aligned} \quad (19)$$

In very rare cases, if the comparison operations yield incorrect results and wrong object recognition messages are sent to every node, subsequent comparisons to pertaining subsets  $A_i$  would yield mismatches. Consequently, every node once again has to traverse the entire silhouette information set  $S_i$  of node  $N_i$ .

Then, the worst-case time complexity in this situation will be

$$\begin{aligned} & t \sum_{i=1}^{m_1} 1 + t \sum_{i=1}^{\ell_2} 1 + t \sum_{i=1}^{m_2} 1 + t \sum_{i=1}^{\ell_3} 1 + t \sum_{i=1}^{m_3} 1 + \dots + t \sum_{i=1}^{\ell_n} 1 \\ & + t \sum_{i=1}^{m_n} 1 = m_1 t + m_2 t + m_3 t + \dots + m_n t + \ell_2 t + \ell_3 t \\ & + \dots + \ell_n t = (m_1 + m_2 + \dots + m_n) t \\ & + (\ell_2 + \ell_3 + \dots + \ell_n) t = t \sum_{a=m_1}^{m_n} a + t \sum_{a=\ell_2}^{\ell_n} a. \end{aligned} \quad (20)$$

Equations (18) and (19) express the total time consumed by a context added system in average and worst environment, respectively, while (20) shows the time consumed by the system in a very rare case. However, in such a situation, the system will not generate a valid surety value. As we will see, it is almost the result in case of a contemporary context aware system under normal circumstances.

For a contemporary context aware system in average-case circumstances, the time complexity is

TABLE 18: CRAM accuracy.






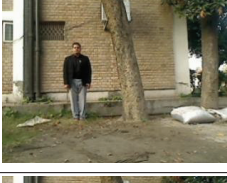
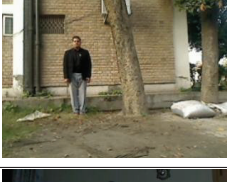

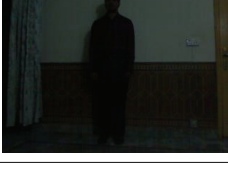
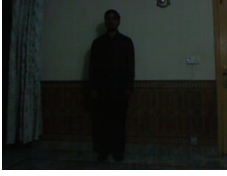
Case number	MO aspect	Image	Image recognized?	
			VISTA	CRAM
1	Car front view		✓	✓
2	Car front view		✓	✗ (Object identified as car back side view)
3	Car left side view		✓	✓
4	Car left side view		✗ Object identified as car right side	✗ Object identified as car right side view
5	Human straight view (good light conditions)		✓	✓
6	Human straight view (good light conditions)		✓	✓
7	Human straight view (good light conditions)		✓	✓
8	Human straight view (bad light conditions)		✓	✓
9	Human straight view (bad light conditions)		✗ Object identified as human loaded view	✓

TABLE 18: Continued.

Case number	MO aspect	Image	Image recognized?	
			VISTA	CRAM
10	Human straight view (bad light conditions)		✓	✓

$$\begin{aligned} & \frac{m_1 + 1}{2}t + \frac{m_2 + 1}{2}t + \dots + \frac{m_n + 1}{2}t \\ &= \frac{1}{2} \left( n + \sum_{a=m_1}^{m_n} a \right) t. \end{aligned} \quad (21)$$

For a contemporary context aware system in worst-case circumstances, the time complexity is

$$\begin{aligned} & t \sum_{i=1}^{m_1} 1 + t \sum_{i=1}^{m_2} 1 + \dots + t \sum_{i=1}^{m_n} 1 = (m_1 + m_2 + \dots + m_n) t \\ &= \left( \sum_{a=m_1}^{m_n} a \right) t, \end{aligned} \quad (22)$$

$$\frac{1}{2} \left( m_1 + n + \sum_{a=\ell_2}^{\ell_n} a \right) t < \frac{1}{2} \left( n + \sum_{a=m_1}^{m_n} a \right) t, \quad (23)$$

$$\left( m_1 + \sum_{a=\ell_2}^{\ell_n} a \right) t < \left( \sum_{a=m_1}^{m_n} a \right) t. \quad (24)$$

Equation (23) depicts the comparison of (18) and (21). Similarly, (24) depicts the comparison of (19) and (22). It is evident that the total time consumed by the context added system comprising  $n$  nodes to track and recognize a MO is much less than that of a context aware system.

**6.2. Context Addition Is Recursive.** In order to assess the behavior of context addition, we consider the trajectory of MO through the same set  $N$  of  $n$  nodes as discussed in previous section. Suppose that, at node  $N_1$ , the percentage surety level of the detected MO is calculated as a value  $V_1$ . At node  $N_2$ , its local surety value  $V_2$  is calculated by adding to it the received surety value  $V_1$  of node  $N_1$ . The context addition of our architecture proceeds in this recursive way. The recurrence equation for  $V_n$  received from  $n$  nodes is as the following:

$$V_n = \begin{cases} V_1, & \text{for } n = 1, \\ V_n + V_{n-1}, & \text{for } n \geq 2. \end{cases} \quad (25)$$

## 7. Conclusions and Future Work

In this paper, we present context aware systems that evolve into autonomic, intelligent processing systems through the

incorporation of context addition. We have presented conceptual, architectural, and deployment aspects of context added system. Through establishing an analogy between context added system and human anatomy of memory, we have proposed the incorporation of reflex arc into context aware systems. We have demonstrated that both context awareness and reflex arc can be embedded into visual context aware systems through prototype implementation.

As part of our future work, the recursive behavior of the CARM will be analyzed more specifically to calculate and compare the average performance variance of context added versus context aware systems and a possible trade-off between latency, precision, and autonomicity. Finally, the authors are quite optimistic to study and analyze the self-learning and reflex response features of the proposed model in the domain of Internet of Things to contribute towards the initiative of *Intelligent Civilization*.

## Competing Interests

The authors declare that they have no competing interests.

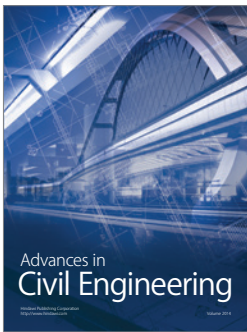
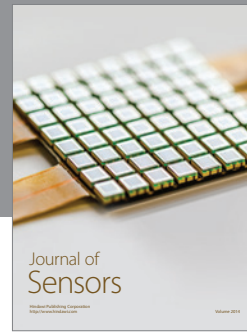
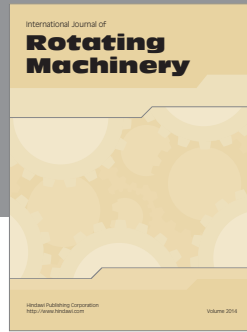
## References

- [1] S. Fayed, S. M. Youssef, A. El-Helw, M. Patwary, and M. Moniri, "Adaptive compressive sensing for target tracking within wireless visual sensor networks-based surveillance applications," *Multimedia Tools and Applications*, 2015.
- [2] S. Jabbar, A. H. Akbar, S. Zafar, M. M. Quddoos, and M. Hussain, "VISTA: achieving cumulative VISION through energy efficient Silhouette recognition of mobile Targets through collaboration of visual sensor nodes," *EURASIP Journal on Image and Video Processing*, vol. 2014, article 32, 24 pages, 2014.
- [3] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.
- [4] N. Ryan, J. Pascoe, and D. Morse, "Enhanced reality fieldwork: the context aware archaeological assistant," vol. 750 of *Bar International Series*, pp. 269–274, 1999.
- [5] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, "Cyber guide: a mobile context-aware tour guide," *Wireless Networks*, vol. 3, no. 5, pp. 421–433, 1997.
- [6] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstathiou, "Developing a context-aware electronic tourist guide: some issues and experiences," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 17–24, ACM, April 2000.

- [7] A. K. Dey, "Context-aware computing: the Cyber Desk project," in *Proceedings of the AAAI Spring Symposium on Intelligent Environments*, pp. 51–54, Menlo Park, Calif, USA, March 1998.
- [8] P. J. Brown, "The stick-e document: a framework for creating context-aware applications," in *Proceedings of the Electronic Publishing (EP '96)*, pp. 182–196, Palo Alto, Calif, USA, January 1996.
- [9] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Handheld and Ubiquitous Computing*, vol. 1707 of *Lecture Notes in Computer Science*, pp. 304–307, Springer, Berlin, Germany, 1999.
- [10] R. Hull, P. Neaves, and J. Bedford-Roberts, "Towards situated computing," in *Proceedings of the 1st International Symposium on Wearable Computers. Digest of Papers*, pp. 146–153, IEEE, October 1997.
- [11] A. Zimmermann, M. Specht, and A. Lorenz, "Personalization and context management," *User Modelling and User-Adapted Interaction*, vol. 15, no. 3–4, pp. 275–302, 2005.
- [12] S. Loke, *Context-Aware Pervasive Systems: Architectures for a New Breed of Applications*, CRC Press, New York, NY, USA, 2006.
- [13] N. Baker, M. Zafar, B. Moltchanov, and M. Knappmeyer, "Context-aware systems and implications for future internet," in *Towards the Future Internet*, G. Tselentis, J. Domingue, A. Galis et al., Eds., pp. 335–344, IOS Press, Amsterdam, The Netherlands, 2009.
- [14] A. Schmidt, "Context-aware computing: context-awareness, context-aware user interfaces, and implicit interaction," in *The Encyclopedia of Human-Computer Interaction*, The Interaction Design Foundation, 2nd edition, 2013.
- [15] S. Chatterjea, T. Nieberg, N. Meratnia, and P. Havinga, "A distributed and self-organizing scheduling algorithm for energy-efficient data aggregation in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 4, article 20, 2008.
- [16] S. Jin and K. Li, "LBCS: a load balanced clustering scheme in wireless sensor networks," in *Proceedings of the 3rd International Conference on Multimedia and Ubiquitous Engineering (MUE '09)*, pp. 221–225, Qingdao, China, June 2009.
- [17] A. Zafeiropoulos, P. Gouvas, A. Liakopoulos, G. Mentzas, and N. Mitrou, "NEURON: enabling autonomy in wireless sensor networks," *Sensors*, vol. 10, no. 5, pp. 5233–5262, 2010.
- [18] R. Žontar, M. Heričko, and I. Rozman, "Taxonomy of context-aware systems," *Elektrotehniški Vestnik*, vol. 79, no. 1–2, pp. 41–46, 2012.
- [19] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.
- [20] A. Bikakis, T. Patkos, G. Antoniou, and D. Plexousakis, "A survey of semantics-based approaches for context reasoning in ambient intelligence," in *Constructing Ambient Intelligence*, vol. 11 of *Communications in Computer and Information Science*, pp. 14–23, Springer, Berlin, Germany, 2008.
- [21] C. Venezia and C. A. Licciardi, "Improve ubiquitous web applications with context awareness," in *Proceedings of the 11th International Conference on Intelligence in Service Delivery Networks (ICIN '07)*, vol. 16, pp. 649–656, October 2007.
- [22] B. Moltchanov, M. Knappmeyer, C. A. Licciardi, and N. Baker, "Context-aware content sharing and casting," in *Proceedings of ICIN*, Bordeaux, France, October 2008.
- [23] H. Chen, T. Finin, and A. Joshi, "An intelligent broker for context-aware systems," in *Proceedings of the 5th International Conference on Ubiquitous Computing (UbiComp '03)*, vol. 3, pp. 183–184, Seattle, Wash, USA, October 2003.
- [24] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "A middleware infrastructure for active spaces," *IEEE Pervasive Computing*, vol. 1, no. 4, pp. 74–83, 2002.
- [25] P. Floréen, M. Przybilski, P. Nurmi et al., "Towards a context management framework for MobiLife," in *Proceedings of the 14th IST Mobile & Wireless Summit*, pp. 20–28, June 2005.
- [26] C. Cordier, F. Carrez, H. Van Kranenburg et al., "Addressing the challenges of Beyond 3G service delivery: the SPICE service platform," in *Proceedings of the Workshop on Applications and Services in Wireless Networks (ASWN '06)*, Berlin, Germany, May 2006.
- [27] J. C. Yelmo, J. M. del Álamo, R. Trapero et al., "A user-centric service creation approach for Next Generation Networks," in *Proceedings of the 1st ITU-T Kaleidoscope Academic Conference Innovations in NGN: Future Network and Services (K-INGN '08)*, pp. 211–218, IEEE, Geneva, Switzerland, May 2008.
- [28] L. Lamorte, C. A. Licciardi, M. Marengo et al., "A platform for enabling context aware telecommunication services," in *Proceedings of the 3rd Workshop on Context Awareness for Proactive Systems*, Guildford, UK, June 2007.
- [29] E. S. Barrenechea, P. S. Alencar, R. Blanco, and D. Cowan, "Context-awareness and adaptation in distributed event-based systems," Tech. Rep. CS-2011-14, David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada, 2011.
- [30] J. Kane, B. Tang, Z. Chen et al., "Reflex-tree: a biologically inspired parallel architecture for future smart cities," in *Proceedings of the 44th International Conference on Parallel Processing (ICPP '15)*, pp. 360–369, Beijing, China, September 2015.
- [31] P. K. Anokhin, *Biology and Neurophysiology of the Conditioned Reflex and Its Role in Adaptive Behavior*, vol. 3 of *International Series of Monographs in Cerebrovisceral and Behavioral Physiology and Conditioned Reflexes*, Elsevier, Philadelphia, Pa, USA, 2013.
- [32] I. P. Pavlov and G. V. Anrep, *Conditioned Reflexes*, vol. 614, Courier Corporation, 2003.
- [33] J. F. Iles, M. Stokes, and A. Young, "Reflex actions of knee joint afferents during contraction of the human quadriceps," *Clinical Physiology*, vol. 10, no. 5, pp. 489–500, 1990.
- [34] E. Onur, C. Ersoy, and H. Deliç, "How many sensors for an acceptable breach detection probability?" *Computer Communications*, vol. 29, no. 2, pp. 173–182, 2006.
- [35] Y.-X. Li, H.-S. Shi, and S.-P. Zhang, "An energy-efficient MAC protocol for wireless sensor network," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE '10)*, pp. V4619–V4623, IEEE, Chengdu, China, August 2010.
- [36] C. Ford and I. Stange, "A framework for generalizing public safety video applications to determine quality requirements," in *Proceedings of the IEEE Conference on Multimedia Communications, Services & Security*, Krakow, Poland, May 2010.
- [37] A. K. Bashir, A. H. Akbar, S. A. Chaudhary, C. S. Hussain, and K.-H. Kim, "Collaborative detection and agreement protocol for routing malfunctioning in wireless sensor networks," in *Proceedings of the 8th International Conference Advanced Communication Technology (ICACT '06)*, vol. 1, pp. 327–332, IEEE, Phoenix Park, Ireland, February 2006.

- [38] S. Chandra, C. S. Ellis, and A. Vahdat, "Managing the storage and battery resources in an image capture device (digital camera) using dynamic transcoding," in *Proceedings of the 3rd ACM International Workshop on Wireless Mobile Multimedia*, pp. 73–82, ACM, August 2000.
- [39] L. Xu, S. Zhang, Z. He, and Y. Guo, "The comparative study of three methods of remote sensing image change detection," in *Proceedings of the IEEE 17th International Conference on Geo Informatics*, pp. 1–4, 2009.
- [40] M. T. Ullman, "Contributions of memory circuits to language: the declarative/procedural model," *Cognition*, vol. 92, no. 1-2, pp. 231–270, 2004.
- [41] E. Tulving, "Episodic and semantic memory 1," *Organization of Memory*, vol. 381, no. 4, article e402, 1972.
- [42] D. L. Schacter, D. T. Gilbert, and D. M. Wegner, "Semantic and episodic memory," in *Psychology*, pp. 240–241, 2nd edition, 2011.
- [43] M. Hilker, J. Schwachtje, M. Baier et al., "Priming and memory of stress responses in organisms lacking a nervous system," *Biological Reviews*, 2015.
- [44] D. L. Schacter and E. Tulving, *Memory Systems 1994*, MIT Press, Cambridge, Mass, USA, 1994.





**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

