

RESEARCH

Open Access

VISTA: achieving cumulative Vision through energy efficient Silhouette recognition of mobile Targets through collABoration of visual sensor nodes

Sana Jabbar^{1*}, Ali Hammad Akbar¹, Saima Zafar², Muhammad Mubashir Quddoos² and Majid Hussain³

Abstract

Visual sensor networks (VSNs) are innovative networks founded on a broad range of areas such as networking, imaging, and database systems. These networks demand well-defined architectures in terms of sensor nodes and camera deployment, image capturing and processing, and well-organized distributed systems. This makes existing VSN architectures deficient because these are limited in approach and in design. In this paper, we propose VISTA, a distributed vision multi-layer architecture aimed at constructing the cumulative vision of mobile objects (MOs). VISTA realizes silhouette recognition of mobile targets through (a) pre-meditated deployment of sensor nodes (SNs) that are equipped with sonar sensors and fixed view (FV) on-board cameras present at the periphery of region of interest (RoI) and SNs with only on-board cameras within RoI, (b) pre-distribution of silhouettes of known objects across SNs, (c) sonar-based presence detection of MO at the outskirts of RoI, (d) MO silhouette capturing and matching at interior node to determine the % age match, (e) subsequent activation of next interior cameras in order to improve % age match, and (f) terminating further activation upon threshold recognition of MO. Experimental evaluation of our image processing algorithms against baseline algorithms with respect to execution time and memory shows significant reduction in image data and memory occupancy. Also, experiments show that true match is achieved fully under broad daylight conditions and large backgrounds when our proposed background subtraction and pixel reduction techniques are used. The mobility-driven behavior of associated network layer algorithms of VISTA is simulated in a network simulator (NS2) by representing the surety of MO identification as a function of number of cameras, database size and distribution, MO's trajectory, stored perspectives, and network depth. The simulation results show that doubling and, in some situations, manifold increase is observed in the surety of the target with an increase in the number of silhouettes deployed against the baselined database size and mobility model. The results substantiate that VISTA is a suitable architecture for low-cost, autonomous and efficient human and asset monitoring surveillance, friend-or-foe (FoF) identification, and target tracking systems.

1 Introduction

Visual sensor networks (VSN) forms the crossroads of networking, image capturing, processing and rendering techniques, and distributed systems. These innovative networks are emerging as an important research challenge and gaining notice of both research community and

applications developers. The contemporary VSN architectures are limited in approach and in design. For instance, none of these architectures takes into account civil infrastructure and geographical information in the placement and simultaneous activation of sensors or cameras or both. Likewise existing VSN architectures focus on capturing images in entirety, which tends to be redundant and at times even detrimental to user application requirements. Also, these schemes tend to overlook the constrained ambulatory behavior of mobile objects (MOs) such as varying mobility behavior in the interior and at

*Correspondence: sana.jabbar@kics.edu.pk

¹ Al-Khwarizmi Institute of Computer Science, University of Engineering and Technology (UET), Lahore 54890, Pakistan

Full list of author information is available at the end of the article

the exterior of region of interest (RoI). Furthermore, these architectures do not reflect on features and attributes of captured images as means for defining the camera activation schedule and coordination between sensor nodes (SNs). Finally, hardware choices for VSNs are either limited to cameras mounted onto mobile assemblies or cameras using pan-tilt-zoom (PTZ) assemblies, both involving mechanical motion. All in all, existing work makes strong assumptions about the presence and availability of video-customized hardware and codecs, bandwidths of the orders of megabit per second (Mbps), and mains power supply or unconstrained battery sources, all defining VSN design in concordance. In this research, we adopt contra-concordance by redefining and restricting VSN features to meet the limited capabilities of real wireless sensor networks (WSNs), which have limited form factors in computation and memory and are equipped with wireless transceivers. We propose VISTA, an architecture that involves redefining the video capturing capability of VSNs. The hardware for VISTA is deployed considering the civil infrastructure of RoI to be monitored. VISTA proposes a deployment scheme in which SNs are placed at optimal positions in order to make communication effective. Camera at the next hop SN is activated when MO comes in its range, such that redundant image is avoided. Only the SNs at the boundary of the RoI are activated to avoid unnecessary consumption of energy of interior SNs in the network.

This research includes the following:

1. Comprehensive layered architecture for achieving cumulative vision.
2. Elaboration of the role played by each layer in order to accomplish the goal which includes hardware role, image processing details, and final task accomplishment.
3. Description of operations concerning the edge nodes (ENs) and the inner nodes (INs).
4. Simulations in NS2 in order to validate our assertion regarding the goals achieved by the proposed architecture.

The remainder of the paper is organized as follows. In Section 2, related work is discussed. Section 3 presents the VISTA architecture in detail. Section 4 presents the experimental results based on NS2 simulations. Section 5 presents the discussion on the operation and performance of VISTA.

2 Literature review

The challenges in the effective realization of VSNs is a contemporary research problem. Research is being carried out in diverse directions of VSNs that include camera calibration, image processing algorithms, hardware

architecture, communication protocols, and applications. The research work that we explored covers multiple domains of VSNs. The objective of this research is to realize energy efficient solution for VSNs in terms of computation and communication recovering the shortcomings of previous efforts in this direction. Literature review of VISTA is based on three domains that are mentioned below.

Chen et al. [1] focus on capturing images from SNs and reducing these images to object of interest (OoI) through mobile agents in VSN. In this way, volume of image data at each SN in target region is reduced. Though a degree of compression is achieved through segmentation, and transmission of OoI only, whole process remains an image processing and transmission scheme. Compression simply reduces the image size to be transmitted. However, image transmission is impractical for long-lived VSNs because VSNs once deployed are sporadically used over very long times. Nelson and Khosla [2] describe a number of criteria that assist in improving visual resolution of a MO. These criteria are used to control the focus and motion of single or multiple cameras. They address camera resolution by suggesting that cameras can actually be moved. Since in WSNs, energy per SN is very limited; therefore, this idea is impractical for VSNs. In [3], Navarro-Serment et al. present their work that is related to the inspection of moving targets in RoI by activating multiple cameras which distributes the collective tasks of identification and prevent energy consumption on a single robot. They address the problems of scheduling and maneuvering cameras to observe targets based on their present positions. Likewise Capezio et al. [4] develop Cyber Scout, an autonomous surveillance and investigation system to detect and track OoI. They use a network of all-terrain vehicles and focus on vision for inspection, autonomous navigation, and dynamic path planning. In [5], a motion segmentation algorithm is proposed for extracting foreground objects with a PTZ camera. Image mosaicing technique is used to build a planar background. The object is detected by comparing current camera image with the corresponding background indexed from the mosaic. In [6], a novel method is proposed by Saptharishi et al. for temporally and spatially moving objects by automatically learning the relevance of the object's appearance features to the task of discrimination. This method is proposed for distributed surveillance systems. Ukita and Matsuyama [7] perform multi-target tracking by active vision agents (AVAs) that is a network-connected computer with an active fixed-view pan-tilt-zoom (FV-PTZ) camera. Multiple FV-PTZ active cameras are required for detailed measurements of 3D objects. However, their idea for surveillance and tracking is not implementable in VSNs due to maneuvering cameras. Similarly in [8], Matsuyama gives the overview of cooperative distributed

vision (CDV). The goal of CDV is to embed network-connected mobile robots with active cameras in a real world and realize wide-area dynamic scene understanding and visualization. However, all of the above ideas for surveillance and tracking are not implementable in VSNs due to maneuvering cameras and significant power consumption of devices.

For image recognition, Tien et al. [9] propose a novel method based on non-uniform rational B-splines (NURBS) and cross-ratios. They propose a method that utilizes both memory and computation time, but the resources required are less as compared to those of the curve matching method. They use a small database to save memory. But matching by using the NURBS curves first and then applying cross-ratios is still expensive in terms of time and computation for a VSN system. In VISTA, a rich database is proposed, i.e., more aspects of an object deployed in the nodes but avoiding computationally expensive algorithm for matching.

In [10], Soro and Heinzelman take into account the unique characteristics and constraints of VSN that differentiate VSNs from other multimedia networks as well as traditional WSNs. They outline all areas of VSNs such as applications, signal processing algorithms, communication protocols, sensor management, hardware architectures, middleware support, and open research problems in VSNs by exploring several relevant research directions. They argue that traditional WSN protocols do not provide sufficient support in VSNs. Hence, there is a need to propose new communication protocols and vision algorithms suitable for resource-limited VSN systems.

Background subtraction is an important step in image matching using low-power devices. In [11] Stauffer and Grimson present a background subtraction method which involves thresholding the error between an estimate of the image lacking moving objects and the current image. The background model used in this work models each pixel as a mixture of Gaussians with an on-line approximation used for updating the model. The Gaussian distributions of this mixture model are evaluated to classify pixels which most likely fall in the background process. Since in reality, multiple surfaces show in the view frustum of a pixel along with changes in lighting conditions; therefore, multiple adaptive Gaussians are required. A mixture of adaptive Gaussians is used in this approximation such that as the parameters of the Gaussians are updated, the Gaussians are estimated based on a simple heuristic to find out those which are part of the 'background process'. In [12], the mixture of Gaussians (MoG) concept has been used in a number of sensor network problems. A related work in this context is by Ihler et al. [12]. Their work addresses the problem of automatic self-localization of sensor nodes. The authors redefine the sensor localization problem within a graphical model framework and present

the use of a recent generalization of particle filtering for approximation of sensor locations. In this technique, each message is depicted using either a sample-based density estimate (as a mixture of Gaussians) or as an analytic function. The messages along observed edges are represented by samples and the messages along unobserved edges are described as analytic functions. First, the samples are drawn from the estimated marginal and then these samples are used to approximate each outgoing message. Another paper which employs mixture of Gaussian distributions in sensor networks is by Rabbat and Nowak [13]. They address the problem of in-network data aggregation in sensor networks which comprise sensor nodes capable of sending sensed data to a base station. Normally, it is required to derive an estimate of a parameter or function from the collected data which is huge and redundant. This paper investigates the distributed algorithms for data processing prior to its transmission to a central point which results in reducing the amount of energy spent in obtaining accurate estimate. This estimation problem is defined as the incremental optimization of a cost function concerning collected data from all nodes such that each node adjusts the estimate based on its local data and transmits it to the next node. In distributed expectation-maximization (DEM) algorithm, the measurements are modeled as samples extracted from a mixture of Gaussian distributions with unknown means and covariances, the mixture weights being different at each sensor in the network. Initially, the parameters of the global density are estimated, which are passed through the network such that each sensor node detects the component of the density which best fits its local data. Cho et al. [14] present a smart video surveillance system by deploying visual sensor network. It comprises of an inference framework in which autonomous scene analysis is carried out using distributed and collaborative processing among camera nodes and an effective occupancy reasoning algorithm. For each node in the network, they define a potential function representing how the global inference is coherent with the local measurement on that node. Next, a multi-tier architecture is built and one node in each cluster is chosen as an anchorage node for global inferences. The amount of overlapping between two nodes is used as a basis when constructing a work tree for distributed processing within a cluster. The existence probabilities for each camera are predicted using the binary images from the background subtraction. A modified mixture of Gaussian (MOG) algorithm is used. In [15], Tsai and Lin deal with contextual redundancy linked with background and foreground objects in a scene. They propose a scene analysis technique that classifies macroblocks based on contextual redundancy. Only specific context of macroblock is analyzed for motion which involves salient motion through an object-based coding architecture. The context of a scene is defined as

the association of a pixel in a scene with static or moving background or moving foreground with/without illumination change based on the observation on a number of recent frames. The context of a macroblock is modeled by an estimated background image. In the scene analysis method, most representative Gaussian is selected from mixture of Gaussians. In [16], Ellis presents a multi-view video surveillance system with algorithms for detecting and tracking moving objects. The scene-dependent information is depicted by creating models of the scene based on observations obtained from the camera network. In order to cater for the background changes, the probability of detecting a pixel value is modeled by a mixture of Gaussians based on color and monochrome pixel values. In [17], Paletta et al. present a video surveillance system for monitoring passenger flows at public transportation junctions based on a network of video cameras. In background modeling and motion detection module, they employ an adaptive model for background estimation applying mixture of Gaussians and appearance patterns, thereby presenting a stable and robust background model.

Kumar [18] demonstrates the importance of various features in image matching. A framework is proposed consisting of hardware cameras and accompanying software. The software manages processing of image and satisfies queries from other cameras over the network or by the camera itself. The software logic is implemented over the publisher-subscriber model. To satisfy queries, different handlers are registered to publisher-subscriber block. It is asserted that scale invariant feature transform (SIFT) features do not work well when there is large orientation change and low resolution. It is also shown that SIFT features do not work efficiently across cameras that are far in terms of time or location. Therefore, it is always efficient to use more than one identification feature in different scenarios. In [19], Margi et al. study Meerkats project and observe the trade-off between power efficiency and performance and realize verifications through a test-bed based on the Crossbow Stargate platform. They observe energy consumption of activities such as processing, image acquisition, flash memory access, and communication over the network. They also report steady-state and transient energy consumption behaviors. They prove that transients are not at all negligible, neither in terms of power nor in terms of delay incurred. They conclude that delay and energy measurements are very important for performance, and transients play a significant role in terms of delay and energy. In [20], Margi et al. present power consumption analysis and execution time for the elementary tasks such as sensing, processing, and communication. These tasks compose duty cycle of a VSN node based on Crossbow Stargate board. They also predict the life time of a VSN system

by considering energy consumption characterization and draw attention to the fact that activation/deactivation of the hardware and transition between different states of a SN requires non-negligible amount of time and energy. They illustrate that SN performs the same functionality but with different energy requirements, depending upon the SN's current state. They also prove that on-board detection always plays a significant role in energy saving even if the rate of event detection is high. To determine event detection, SN requires blob detector which further decides whether image should be transmitted or not. However, blob detection is a power-consuming process that must be run in either case. Even in the case of event detection, image is compressed by the node and sent to sink or any other node. Image compression saves energy, but when blob detector detects larger blobs in acquiring image, it takes sufficient amount of energy to send that image. Image compression only reduces the size of the image, and small blobs require high energy and long time for transfer. In order to overcome the abovementioned problems, we introduce a novel idea in VISTA, where we identify MO on a node and only send the information about MO without sending image data. Qureshi and Terzopoulos [21-24] present work related to smart camera networks which consists of static and active cameras that provide coverage of environment with minimal reliance on human operator. They propose a distributed strategy in which nodes are capable of local decision making and inter-node communication. Each camera node has an autonomous agent to communicate with nearby nodes. When the node is in idle state, the camera does not perform any task. Upon receiving message, node calculates its relevance to the task by employing low-level visual routines (LVR). Supervisor node decides whether or not to include node in the group by observing its relevance value. A visual routine occurs every time when a node receives the message. This means that every time, the node bears the burden of running LVR and calculates its relevance to the task. Contrary to this, in our approach, we invoke only that node which can participate efficiently in identifying MO. An overview of VSNs along with research challenges in this area is given in [25]. The need for tight coupling between communication protocols and vision techniques for effective object monitoring and tracking is also highlighted. Many surveys about VSNs are published in the past in which VSN characteristics, its corresponding layers, and open research issues are discussed in detail. An extensive survey of wireless multimedia sensor networks is provided in [26], where Akyildiz et al. discuss various open research problems in multimedia research area, including networking architectures, layers, and protocols. Similarly in [10], the authors overview the current state-of-the-art in the field of visual sensor networks, by exploring several relevant research directions.

All these authors agree about the fact that previous architectures cannot fulfill the need of this new smart visual sensor network era. They suggest that development of some new energy-efficient architectures for sending visual information is the need of the day.

In paper [27], the authors highlighted the major wireless visual sensor network approaches for energy efficiency. They analyze the already proposed strategies in this domain. They suggested that enhancement should be done in LANMAR [28] and in G-AODV [29] to increase the energy efficiency of VSNs in future. They also suggested that due to the different elements that enter into the design of visual sensor networks, multidisciplinary research is essentially needed to design future VSN that provide an effective trade-off between the energy associated with the VSN and the QoS received by the end user. Paper [30] focuses on the functionality of VSNs as intelligent systems capable of operating autonomously and in a wide range of scenarios. The authors feel the need of extensive research regarding the placement of these nodes, coverage of blind areas and how to localize and calibrate camera nodes within the network.

To overcome the above mentioned energy consumption and architectural problems, a novel architecture ‘VISTA’ is proposed through which SNs energy can be saved by pre-planned database and camera scheduling.

3 VISTA architecture

This chapter elaborates the VISTA design that comprises layered architecture. First, the assumptions are formulated and then we discuss details of layers and modules that comprise this architecture.

3.1 VISTA assumptions

The following assumptions are taken for VISTA:

- All nodes in VISTA are deployed in a pre-engineered topology.
- At the time of network initialization, there is no mobile object inside RoI. Even if there is an already present MO, it is not the scope of VISTA to track it.
- On ENs, sonars are mounted along with cameras.
- INs are equipped with cameras only.
- Each SN location is pre-programmed in RoI.
- All SNs have same computational resources and memory.
- ENs exhibit three different states of activation with respect to (w.r.t.) to sonar, timer, camera, and transceiver as shown in Table 1.
- Similarly, INs exhibits two different states of activation w.r.t. timer, camera, and transceiver as shown in Table 2.
- Field Of view (FOV) of edge nodes’ sonars and cameras is calibrated to be exactly the same respectively.

Table 1 States exhibited by ENs

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

3.2 VISTA network model

VISTA sensor node functionality and their operation for mobile object recognition can be thoroughly illustrated by deploying specific mobility models. Freeway and Manhattan [31] are the most suitable mobility models for VISTA sensor nodes because the mobility of the target in the RoI is constrained on road segments only. We take Manhattan mobility model as an example to demonstrate the VISTA SN behavior for mobile object recognition in region of interest. VISTA assumes two types of SNs called ENs and INs. Edge nodes cover the region of interest boundary both through sonars and cameras that detect and recognize any entering MO, while INs only have cameras for recognizing it. We propose two edge nodes’ layers that cover RoI border for reliability and for energy efficient MO detection. The operation of two layers of ENs is realized by forming a triplet in which two edge nodes are in foreground and one edge node in the background. The operation of triplet is explained in Section 3.3.2. In this section, we propose considerations for camera deployment at both types of sensor nodes.

3.2.1 Considerations for camera deployment of ENs

Case 1: number of cameras at outer EN layers. In order to secure the entire perimeter, each side of length ‘L1’ can be covered by $n1$ ENs of width ‘W1’ of FOV such that $L1/W1 = n1$.

Case 2: number of cameras at inner EN layers. In order to secure the entire perimeter, each side of length ‘L2’ can be covered by $n2$ ENs of width ‘W2’ of FOV such that $L2/W2 = n2$, where $n2 = n1 - 1$ such that $W2 = 2W1$.

3.2.2 Considerations for camera deployment of INs

INs are deployed in such a way that all aspects of a MO image could be captured by their cameras. Two types of INs are deployed for Manhattan models; their camera considerations are given below.

Table 2 States exhibited by INs

State	Timer	Camera	Transceiver
Active	1	1	1
Demi-sleep	0	0	1

Case 1: number of IN cameras for covering straight road segments. IN cameras with their FOVs are deployed in such a manner that each camera covers 'n' horizontal road segments or 'm' vertical road segments in RoI, where 'n' or 'm' depends on the total number of rows and columns of RoI. For example, IN_3 covers three horizontal road segments such as RS_H1, RS_H2, and RS_H3, while IN_7 covers three vertical road segments such as RS_V1, RS_V2, and RS_V3 as shown in Figure 1. The total number of INs required to cover all the horizontal and vertical road segments is given below:

$$\text{Total number of INs required} = (r - 1) + (c - 1), \quad (1)$$

where r = total rows of RoI, and c = total columns of RoI.

Case 2: number of IN cameras for covering Carrefour. IN cameras with their FOVs are deployed in such a manner that each camera covers a single Carrefour in RoI as shown in Figure 1. Thus, the total number of Carrefour INs required is given below:

$$\text{Total Carrefour INs required} = (r - 2).(c - 2), \quad (2)$$

where r = total rows of RoI, and c = total columns of RoI. This camera deployment also affects the database at each SN that will be discussed in the database section.

3.3 VISTA-layered tenon mortise architecture

The proposed VISTA-layered tenon mortise architecture consists of three layers as shown in Figure 2. The modules in these layers work collaboratively in order to accomplish silhouette-based recognition and tracking of MO.

The hardware specifications are defined at the physical layer. Physical layer provides hardware entities to the network and processing layers. Sonar and timer give the input signals to the network layer, while camera gives input image frame to the image processing (IP) module that is the part of processing layer. The network layer that is above the physical layer implements timer and sonar modules. The network layer can also be executed by the IP module and routing decision module to send and receive the packetized messages. The processing layer is at the top level incorporating basic functionalities for detection-, recognition-, and perspective-based MO tracking. Database management module also resides at this layer.

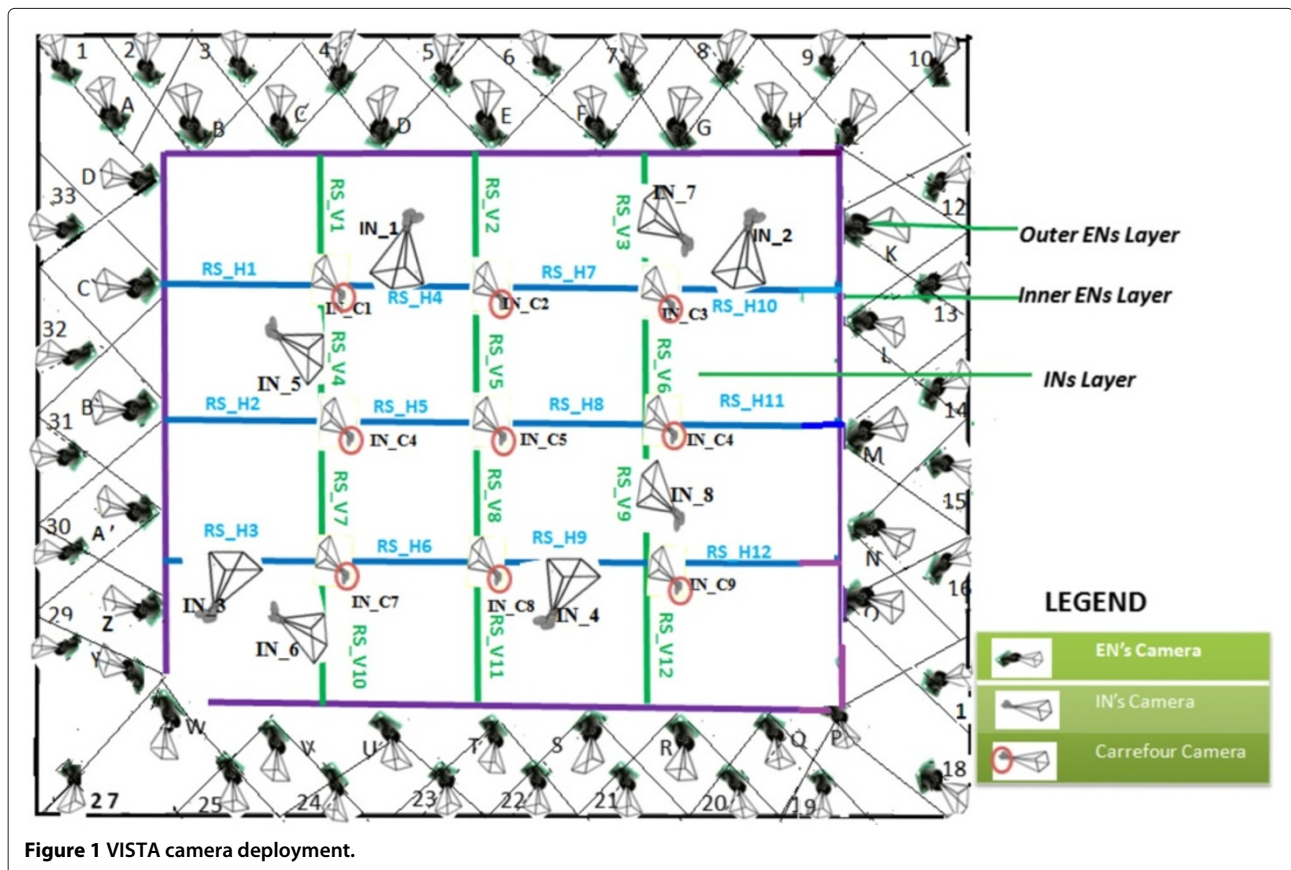
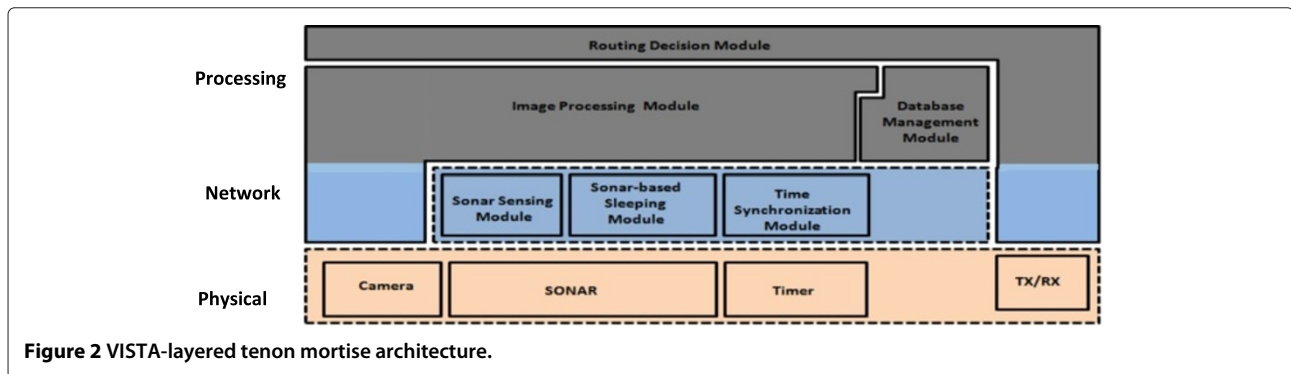


Figure 1 VISTA camera deployment.



3.3.1 Physical layer

The physical layer comprises hardware devices such as sonar, timer, camera, and transceiver.

- Sonar is used for proximity detection of MOs. Firstly, sonar detects the arrival of MO in RoI and then cameras are switched ON to further analyze it.
- Timers are used to provide time information of MO detection event in the RoI, MO image capturing activity at SNs and to activate other SNs in the network in time.
- Camera is activated only to capture an incumbent MO and then it is turned off immediately to conserve energy. The captured image is used to recognize the MO by applying different image processing algorithms.
- Transceiver is used to send and receive packetized messages. Typical examples could include ZigBee- and Wi-Fi (802.15.4)-based transceivers.

3.3.2 Network layer

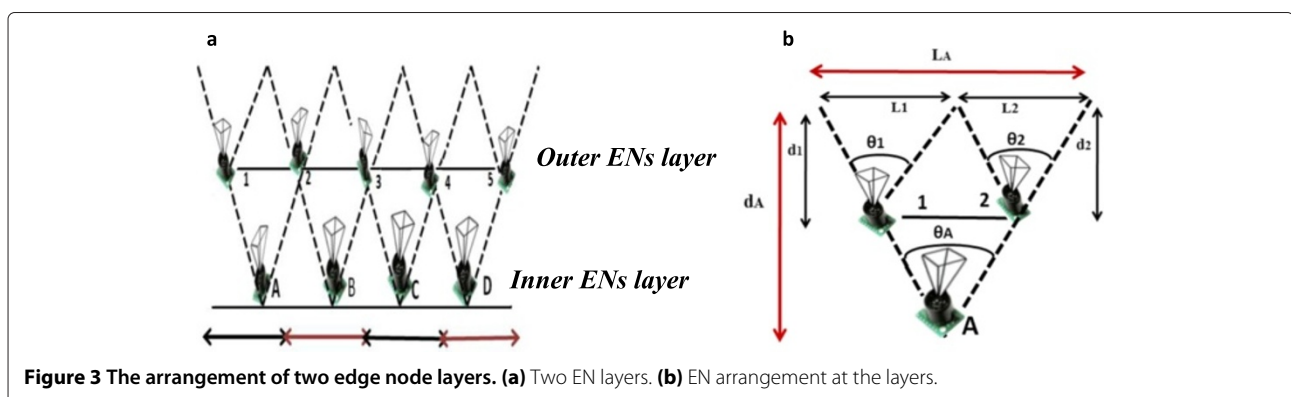
This layer is concerned with sonar-sleeping, sonar-sensing, and time synchronization modules that are discussed below:

Sonar-sleeping module We propose two EN layers to ensure the presence of any MO in RoI because if one edge node layer fails to detect the presence of any MO

in region of interest due to any technical failure, then it would never be detected by VISTA again because VISTA assumes sonars only at edge nodes. Figure 3a shows two edge node layers, i.e., the outer edge node layer and the inner edge node layer, in which the outer layer edge nodes are called foreground edge nodes, while the inner layer edge nodes are called background edge nodes. The arrangement of edge nodes in these two layers forms a federation such that one background edge node lies between two foreground edge nodes, which is present in the form of a triplet, as shown in Figure 3b. The entire perimeter is covered through such triplets. In a triplet, a background EN has the coverage equivalent to the sum of the coverage of both foregrounds edge nodes since its FOV width is equal to the sum of the FOV widths of both edge nodes. The main purpose forming triplets is failure resilience.

$$\begin{aligned} \text{Power consumption of outer EN layer} &= (n + 1) \times P \\ &= n \times P + P \\ &\approx n \times P, \quad (3) \end{aligned}$$

where P is the power consumed by a single EN. When the inner EN distance is doubled (to that of outer ENs), the power consumption increases fourfold. So,



$$\begin{aligned} \text{Power consumption of inner EN layer} &= (n/2) \times 4P \\ &\approx 2 \times n \times P \end{aligned} \quad (4)$$

$$\begin{aligned} \text{Average power of two layers} &= n \times P + 2 \times n \times P/2 \\ &= 1.5 \times n \times P. \end{aligned} \quad (5)$$

So, our proposed two layers use power = $1.5 \times n \times P$.

The power consumption difference between triplet formation and one-layer deployment is $0.5 \times n \times P$. The triplet formation bears an overall power tax of $0.5 \times n \times P$. However, such added power tax gives us complete border breach avoidance [32] system even if 50% of the network nodes fail as described in the Table 3 below. The two neighboring triplets support the failing triplet. To provide such triplet coverage, we propose the deployment distance for all the background ENs in the form of equations.

Since

$$L = d\theta, \quad (6)$$

so,

$$L_A = d_A\theta_A \quad (7)$$

$$L_1 = d_1\theta_1 \quad (8)$$

$$L_2 = d_2\theta_2. \quad (9)$$

Since we want to keep $L_A = 2L_1$, so the deployment distance should be $d_A = 2d_1$. Similarly, $d_A = 2d_2$, such that when an inner EN is ON, its coverage is equal to the coverage of two outer ENs. The power consumption at edge nodes can be reduced by altering the sleeping schedule between the foreground and background edge nodes of the triplet, thus achieving overall network longevity. If any foreground edge node in a triplet fails, then the background edge node can be invoked instead. Similarly, the foreground edge nodes in a triplet can be invoked in case of background edge node failure. Edge nodes' layers exchange their sleeping schedules through SMAC

protocol [33], in which the background edge nodes are considered as a synchronizer, while the foreground edge nodes are as followers in triplets.

Sonar-sensing module MO entrance in RoI is detected by this module upon receiving MO detection signal from sonar. At the time of deployment, distance-based fingerprinting of reflected received signal strength indicator (RSSI) is performed and stored as RSSITHRESHOLD in each edge node. Edge node periodically sends out beacon, looking for possible mobile object presence at the boundary of RoI. An EN only reacts to a detected mobile object, only if the measured RSSI is greater than RSSITHRESHOLD. When a sonar signal is received by the outer EN layer, then the foreground duo edge nodes in a triplet communicate with each other to decide the appropriate edge node for initiating mobile object detection activity. This decision is taken by seeing the measured RSSI strength at both edge nodes in such duo. If mobile object detection signal is received by inner edge node layer, then the background EN in triplet would proceed with the recognition of mobile object. We proposed that three readings from sonar are analyzed to be assured about the MO presence and assess its trajectory across RoI. By assessing the mobile object movement, only a decision should be taken about the camera activation at the boundary. All the possible trajectories as assessed by the three readings are shown in Table 4. Network longevity is also achieved by avoiding unnecessary camera activation in the network by accessing mobile object movement across RoI. For example, if a mobile object comes closer to sonar and then moves away, no cameras are activated for mobile object recognition. This module is also responsible for generating mobile object detected (MOD) message to activate next most optimal hop SN after detecting mobile object presence in RoI.

Time synchronization module The information about mobile object image is meaningless without the time information at which the image is captured [10] because mobile object might have moved away from the located position when this information reaches the sink. Such time information should be accurate to subsequently transmit mobile object information and to activate other SNs in the network in time. To give accurate time information, timers are synchronized through time synchronization module upon receiving mobile object detection signal from sonar. We propose camera-activation delay avoidance time synchronization (CADETS) scheme to synchronize timers' clocks of SNs. Our proposed scheme for SN timers' synchronization is tailored to the unique sequence of camera activation and image processing.

When the presence of any mobile object is detected by ENs, time synchronization activity starts in the relevant

Table 3 An example of a three-supporting triplet

Malfunction triplet	Failing nodes (malfunction/battery outage)	Alternative working nodes
Triplet 1: '1', '2', 'A'	If '1' and '2' fail	'A' works
Triplet 2: '2', '3', 'B'	If '2' and '3' fail	'B' works
Triplet 3: '3', '4', 'C'	If '3' and '4' fail	'C' works
Triplet 1: '1', '2', 'A'	If 'A' fails	'1' and '2' work
Triplet 2: '2', '3', 'B'	If 'B' fails	'2' and '3' work
Triplet 3: '3', '4', 'C'	If 'C' fails	'3' and '4' work
Triplet 2: '2', '3', 'B'	If '2','3', and 'B' fail	'A' and 'C' work

Table 4 MO movement and VISTA SN behavior

Sonar reading	MO movement	VISTA SN behavior
$r1 = r2 = r3 = 0$	MO not present	No MO recognition done at SN
$r1 = r2 > r3$	MO present but moving away	No MO recognition done at SN
$r1 > r2 > r3$	MO present but moving away	No MO recognition done at SN
$r1 < r2 > r3$	Come close and moving away	No MO recognition done at SN
$r1 = r2 = r3$	MO stands still	MO recognition done only at that SN
$r1 < r2 = r3$	Come close and stop	MO recognition done only at that SN
$r1 > r2 < r3$	Come close and went away and come close	MO recognition done only at that SN
$r1 < r2 < r3$	Moving towards and enter in RoI	MO recognition done over the entire network
$r1 = r2 < r3$	Moving towards	MO recognition done over the entire network

part of RoI through the CADETS scheme. Time synchronization is carried out by ENs, sending beacons to subsequent INs to update time information. Inner nodes use this time information as a point of reference for synchronizing their clocks. In turn, the inner nodes synchronize other inner nodes in their transmission ranges. Time synchronization is performed as a moving localized activity such that the inner nodes are well synchronized before image processing module passes messages among SNs as shown in Figure 4. In the figure, first, the beacon gives the time synchronization information before the second packet carrying MO information is transmitted.

3.3.3 Processing layer

The logical and algorithmic execution of VISTA is part of the processing layer through which it deals with object identification and tracking. This layer constitutes the following modules: database management module, image processing module, and routing decision module. In the following subsections, the role of each of the abovementioned modules is elaborated in realizing the objective of VISTA.

Database management module Our database is deployed in a distributed manner over the entire network, keeping in view the regional aspects, application constraints, and requirements to correctly recognize and track the MO.

Database organization

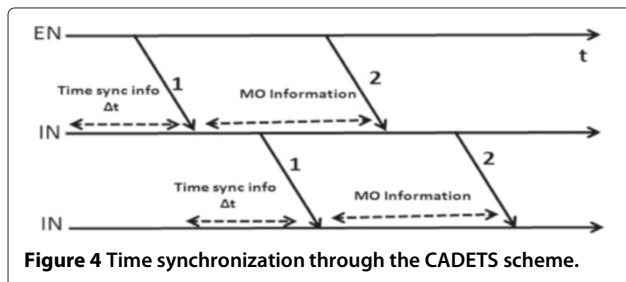


Figure 4 Time synchronization through the CADETS scheme.

- A database table is maintained by each SN to store the neighbor SNs' positions and their camera orientations in its memory. This database helps in perspective-based MO tracking by seeing its neighbor SNs' positions and camera orientations as shown in Table 5.
- Another database table is maintained for the silhouettes of the expected objects with their respective identifiers (Ids), classes, silhouette aspect ratios w.r.t. their segments, octets, views, angles, and sureties as shown in Table 6. Classes distribute stored silhouettes into two categories: class 1 for vehicles and class 2 for humans. Aspect ratios are calculated for each segment of a silhouette, while a silhouette is segmented by seeing the number of features in a silhouette. Silhouette segmentation criteria and their aspect ratio calculation are discussed in the end of this section. Here, octet is an assigned code for identified object in ASCII. View is the angle of MO with respect to camera. Angle defines camera's angle with respect to the MO. Surety of each stored silhouette is defined by the equation below:

Silhouette surety at observing SN

$$= \begin{cases} 0 & \text{if no silhouette is present out of } K \\ 1/K \times 100 & \text{otherwise} \end{cases} \quad (10)$$




where K is the total number silhouettes of a single object with different aspects distributed overall the network.

- The database also stores static background image for further use in image processing activities.

Table 5 SNs' positions and camera orientations

SN_Id	SN position (cell, column/row)	SN camera orientation (direction)
IN_1	(11, I)	(W)
IN_2	(1, II)	(E)
IN_3	(12, III)	(E)

Table 6 Expected MO silhouette with specifics

Id	Class	Silhouette	Aspect ratios w.r.t. segments		Octet	View	Angle	Surety (%)
			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 (human)		4	(segment id, aspect ratio)	Human	Front	90	25

Database deployment criteria

1. SNs' positions and camera orientations table.

- Each SN stores neighbor SNs' positions and their camera orientations to track the MO in RoI. IN_4 is taken as an example from Figure 1 to show the neighbor SNs' positions and their camera orientations as shown in Table 7.

2. Deployment of Silhouette table. All possible aspects of MOs' silhouettes helpful for object identification are distributed throughout the network without redundancy. Expected silhouette table that is deployed on entire network depends on

- Type of SNs: Silhouette table stored at each SN depends on its type. A simple IN stores only one silhouette of a MO in silhouette table. In this case, only one expected silhouette against one MO is enough to recognize the MO. For example, IN_2 silhouette table is shown in Table 8 where only one silhouette is stored against one object. A Carrefour IN stores multiple aspects of silhouettes because a Carrefour IN's camera can not only capture side, front, or back views but also tilted views of a MO. Table 9 shows the multiple aspects of MO silhouettes stored at Carrefour IN_C3.
- Total RoI: When SNs are deployed at constant distance in RoI, then stored silhouettes over the

entire network increases as the total RoI increases. When SNs are deployed at variable distances in RoI, then silhouettes stored over the entire network depends on the total number of SNs that are deployed in RoI. Silhouette table that is deployed on a SN depends on the factors listed below.

- Camera orientation: Silhouette table also considers camera's orientations at SN and stores only those silhouettes of MOs' in the table that have a greater probability of matching with captured MO silhouette. For example, IN_2's camera has the probability to capture the side views of MOs. Therefore, the side views of MO silhouette are stored in the database as shown in Table 8.
- Type of MOs: Silhouette table increases in size at each SN as the type of MO increases that can be entered in the RoI as shown in Table 8.
- If MO adopts Gauss's Markov mobility model patterns rather than the Manhattan model, then all aspects of MOs' silhouettes should be stored in silhouette tables over the entire network in which some are shown in Table 10. In this case, surety depends on the number of stored silhouettes against a single object.

Table 7 SNs' positions and camera orientations of 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)
IN_10	(5, X)	(S)
IN_11	(3, XI)	(N)

Silhouette segmentation and aspect ratio calculation. Silhouette segmentation and aspect ratio calculation is an *a priori* activity to store silhouettes of multiple objects at each SN. These stored silhouettes are used for identifying run time extracted silhouette of a captured mobile object. Silhouettes are segmented by seeing the total number of prominent features. For example, the front view of a human can be segmented into five parts, i.e., head, neck, shoulders, torso, and lower limbs as shown in Figure 5a. Figure 5a also shows the prominent features of a human's side view, loaded view, and armed view, respectively. Similarly Figure 5b shows the features of a hatchback and saloon car with different views.

Aspect ratios can be calculated by taking the width-to-height ratios of the segmented parts. When a human is segmented in five parts by seeing its prominent features, then aspect ratios of all five of these segmented parts are calculated and stored in the database as shown in Figure 6.

Table 8 IN_2 silhouette database






Id	Class	Silhouette	Aspect ratios w.r.t. 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 IN_C3 silhouette database

















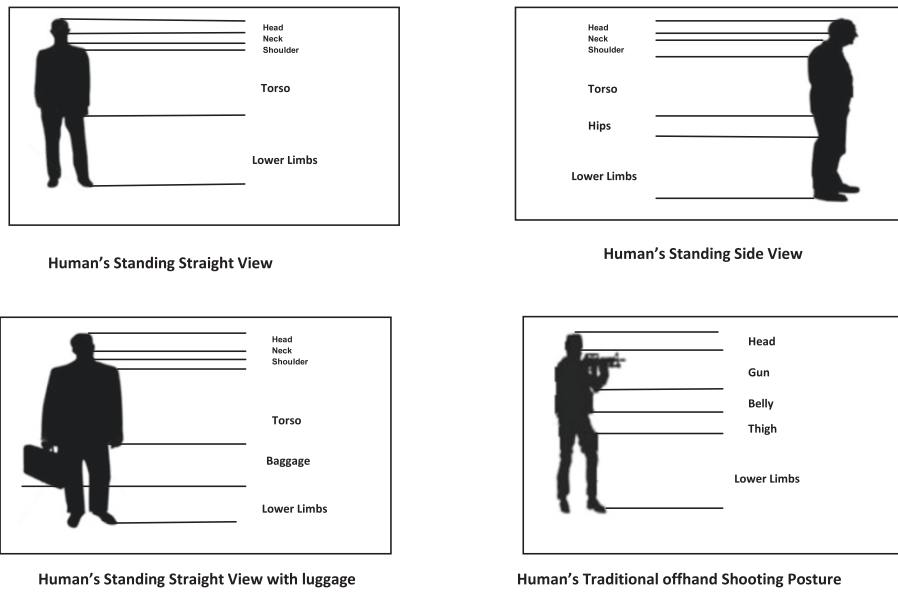
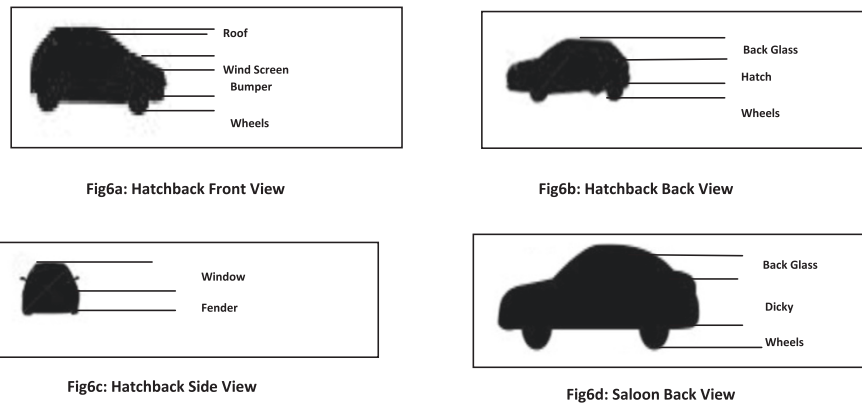
Id	Class	Silhouette	Aspect ratios w.r.t. segments		Octet	View	Angle	Surety (%)
			Segments	Aspect ratios				
1	0		3	(segment id, aspect ratio)	Car	Front	0	12.5
2			3	(segment id, aspect ratio)	Car	Right side	270	12.5
3			3	(segment id, aspect ratio)	Car	Back	180	12.5
4			3	(segment id, aspect ratio)	Bus	Left side	90	12.5
5			2	(segment id, aspect ratio)	Bus	Back	180	12.5
6			3	(segment id, aspect ratio)	Jeep	Right side	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		3	(segment id, aspect ratio)	Human	Left side	90	25
11			3	(segment id, aspect ratio)	Human	Front	0	25

Table 10 Different Aspects of silhouette store in database

Id	Class	Silhouette	Aspect ratios w.r.t. segments		Octet	View	Angle	Surety
			Segments	Aspect ratios				
1	0		4	(segment id, aspect ratio)	Car	Front/left	45	
2			1	(segment id, aspect ratio)	Car	Left side/top	90	
3			3	(segment id, aspect ratio)	Jeep	Left side/tilt	90	Depends on stored silhouettes
4			3	(segment id, aspect ratio)	Car	Right side/back	225	
5			3	(segment id, aspect ratio)	Car	Right side/tilt	270	



a: Human Silhouettes segmentation by seeing the prominent features



b: Car's Silhouettes segmentation by seeing the prominent features

Figure 5 Human and car silhouettes segmentation by seeing the prominent features.

Database silhouette	Segmented parts	Aspect Ratios (W:H)
		0.9259
		3.11
		6.6
		0.759
		0.605

Figure 6 Aspect ratio calculation of a human silhouette.

Image processing module IP module is at the heart of VISTA architecture and plays role in image capturing and processing. Image processing module captures the MO image first and then different image processing algorithms are applied on this captured image to convert it from an image to MO silhouette. Image processing module matches this extracted MO silhouette to the stored silhouettes at SN. Image processing module then recognizes the MO at SN and gives result in terms of surety. Image processing module is initialized by receiving MOD message at SNs or it can be directly initialized by sonar interruption at ENs.

IP module assumptions

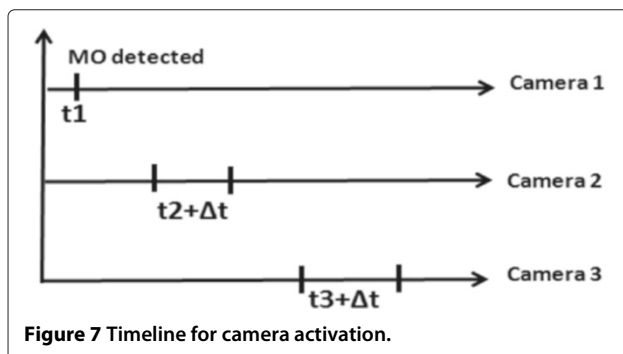
- Captured MO image and stored SN background image are of same size because the FV cameras capture typically well-known objects of known scales which appear on road.
- Background subtraction algorithm is only applicable where the distance between SNs' cameras and road segments remains unchanged.

Image capturing sub-module. An instantaneous image capturing of MO is done by the image processing module as a first step. When a SN receives a MOD message, its camera is invoked for ' Δt ' s and captures fixed-size MO images where Δt affects the total 'shutter ON' time of camera for accommodating variance in expected arrival. A minimum of 25 fps is recommended for motor vehicle traffic areas [34]. A total of ' n ' fps is captured by the sensor node's camera while every ' i th' captured frame is processed for MO recognition and tracking. Figure 7 shows the timeline in which SNs' cameras are sequentially invoked for ' Δt ' s.

Image change detection sub-module. The captured MO images are then passed through image change detection sub-module to detect the presence of mobile object. Using Gaussian mixture model (GMM) for detecting change in the background (or equivalently validating the presence of foreground) for a given number of Gaussian components is known to yield desirable results in the presence

of moving or more appropriately 'evolving' backgrounds. However, for limited form-factor nodes used in multi-camera networks such as VISTA, blanket application of a fixed number of components in GMM as in [11] is not possible. Such impossibility owes to the fact that there exist differences in the fields of view (frustums) for all the camera due to their unique spatial deployment resulting into unique confusing background processes (including swaying background objects, slow moving foregrounds, and shadowing/illuminating effects of light sources) that are disparate and have localized uniquenesses. It is, therefore, needed to adapt the operation of image change detection sub-module at each IN to become more sensitive when the localized background processes are active and less sensitive otherwise. More sensitivity implies that the effects of active background processes are countered through the use of a higher number of Gaussian components in the mixture of Gaussians. When the background is more stationary, the IN can switch back to a lesser number of Gaussians. Although an offline initialization of the number of Gaussians and the identification of a suitable increment size in the number of Gaussians can be performed using information theoretic Bayesian information criterion (BIC) [35], using an intuitively large number of Gaussians is not advisable in VISTA because it results into overfitting at the cost of more energy drain and added complexity. In other words, such adaptation should initialize modestly and subsequently work incrementally (or decrementally) as 'what you see is what you do' (WYSIWYD).

The implementation of the adaptive change detection is proposed through a feedback loop that is established between an up-trajectory node (an edge node that activates an inner node or an inner node that activates another inner node) and a down-trajectory node (an inner node that is activated by an edge node or by another inner node). When the up-trajectory node detects a mobile object (foreground present), it generates mobile object detected message for its down-trajectory neighbor. Such up-trajectory node expects that the down-trajectory node would also detect this mobile object. When the down-trajectory node detects the same mobile object, it sends positive feedback to the up-trajectory node. Such positive feedback implies that the change detection at the up-trajectory node is rightly sensitive (the Gaussian mixture has the right number of components) and that the foreground was indeed present. In case the down-trajectory node sends negative feedback, it means that the change detection at the up-trajectory node is less sensitive and it has to become more sensitive to prevent such false positives. Likewise, the false negatives may also be triggered on reverse links through Carrefour camera nodes. The high-low sensitivity oscillations may result due to the feedback by a malfunctioning down-trajectory node



or a Carrefour node. This can be mitigated through the usage a more formidable hysteresis loop based on consensus and voting algorithms among a group of neighboring inner nodes [36]. This feedback (or hysteresis) loop is recursively intertwined throughout the network for any arbitrary up-trajectory node and its corresponding down-trajectory nodes.

Image compression and storage sub-module. The captured MO image is compressed and stored in SN memory by image compression and storage sub-module. Both compression and storage can be optimized using quality-aware transcoding which is an enabling technology for dynamically changing the image size using a quality-vs.-size tradeoff [37]. We propose surety-based image compression and storage (SICS) in which an image is transcoded and stored by seeing the surety level of that mobile object. Through SICS, the captured image can be transcoded at four different levels against four surety levels that can be stored at four different image quality levels as shown in Table 11. In SICS, an image is transcoded to more elevated levels as the mobile object achieves high surety levels in RoI. An image having highest transcoding level would have lowest image quality level. As the quality level of an image decreases, power consumption of transcoding operation also decreases. For example, at transcoding level 3, an image has only 25% quality level which consumes very low computational power as compared to low transcoding levels. Moreover, mobile object recognition is not affected by storing low-quality images because at high surety levels, low-quality images are adequate for further IP and mobile object recognition. Also, a low-quality stored image takes less memory to save and consumes low computational power in performing IP activities and MO recognition.

The proposed mechanism can be best described by illustrating the example scenario below. A mobile object has '0%' surety level, when it enters first time in the RoI. According to SICS, a MO image is not transcoded in this case and stored in its original size for better identification of the MO. As a MO is covered by more hops of IP by each passing-by SN in the RoI, it achieves high surety levels because of MO recognition at more SNs. When a mobile object achieves more than 25% surety level in the RoI,

then the transcoding level for compressed stored image is decided correspondingly. For example, if a mobile object is recognized by up to 50% surety level, then the captured image is transcoded at level 1 and stored at 75% quality level, and if a mobile object is recognized by more than 75%, then SICS compresses the MO image at the highest transcoding level (i.e., level 3) and stores at lowest image quality level (i.e., 25%).

The total power cost of image capturing and image compression and storage sub-modules is described in the equation below.

$$P = A + C + S, \quad (11)$$

where A is the power consumed by image capturer module, C is the power used in image compression, and S is the power used in storage. Power consumption of A is irrepressible, but C and S are controllable by applying SICS.

Image subtraction sub-module. The stored MO image is subtracted by the background subtracter sub-module to extract the MO silhouette from it. Silhouette is extracted by subtracting MO image from the static background image stored at SN. Background subtraction can be done in low computational cost and in less time by applying 'don't care' operation on some parts of the background image such that silhouette extraction is not affected. To apply the don't care operation on some parts of the background image, the information about changed and unchanged regions in MO image from its corresponding static background is very necessary. So, the regions of the captured image are not processed or 'don't cared' if unchanged. To select the appropriate section for background subtraction operation, the changed and unchanged regions can be detected by applying the equation below [38]:

$$Dx_{ij}^k = x_{ij}^k(t_2) - x_{ij}^k(t_1). \quad (12)$$

This pixel-by-pixel change detection method based on image subtraction is executed only once when a mobile object enters first time in RoI. When changed portions are detected from the background image on successive hops, we can avoid some portions of background image where change is not detected considering them don't care regions. Since we assume the Manhattan model, we propose that the background image is divided into four portions in which some regions are used for background subtraction, while some are ignored by applying don't care operation. Four cases of background subtraction possibly exist in which total number of change detected regions and positions of changed regions are varied as shown in Table 12. We assign a unique code to all possible combinations that are made by seeing the total number of changed regions and their positions. This unique code is

Table 11 SICS surety-based image compression power characteristics

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

Table 12 Total possible changed regions with their positions

Case no.	Total no. of changed regions	Code no.	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	

used to transfer information about background subtraction operation to neighbor SNs. If change is detected in all portions of background image, then the background subtraction operation would be done on whole background image at all sensor nodes. If change is detected in one, two, or three parts, then the background subtraction can be done in one, two, or three portions, respectively. We propose ID-based split background image subtraction in which background subtraction is done on some portions of the background image by seeing the previous sensor node ID. Scene entry region information is used to choose the appropriate portion for background subtraction. Scene entry region information is the information about mobile object navigation on the previous SN where a MO is lastly observed and gives the entrance direction of a mobile object at a SN. Sensor nodes keep this scene entry region information in the database table by using neighbor SNs' Ids. IN_4 is taken as an example from Figure 1 to show the appropriate section of background image for a MO coming from neighbor SNs to IN_4 as shown in Table 13. Here, we assume that changed is detected in only one portion. So, background subtraction would be done

Table 13 Spilt background table for 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	

only on one portion of the background image. Figure 8 shows the time elapsed during the image subtraction time. It shows that image subtraction takes 4.5 times less time if we subtract only one portion as compared to the whole image.

Silhouette comparator sub-module. The extracted silhouette is finally compared with the stored database silhouettes by silhouette comparator sub-module. We propose feature-dependent silhouette segmentation for low energy comparison (FRILL).

Through FRILL, the extracted silhouette is segmented corresponding to the number of segments of the stored silhouettes one by one, such that in each comparison, the number of segments of the extracted silhouette is equal to that of the stored one. Aspect ratios of each individual segment of the extracted silhouette are computed and then

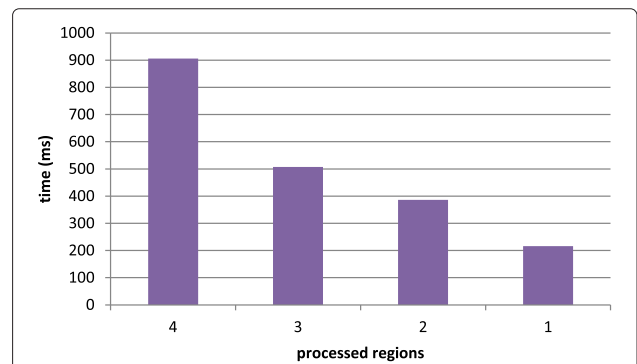


Figure 8 Image subtraction time by applying don't care operation on some portion of an image.

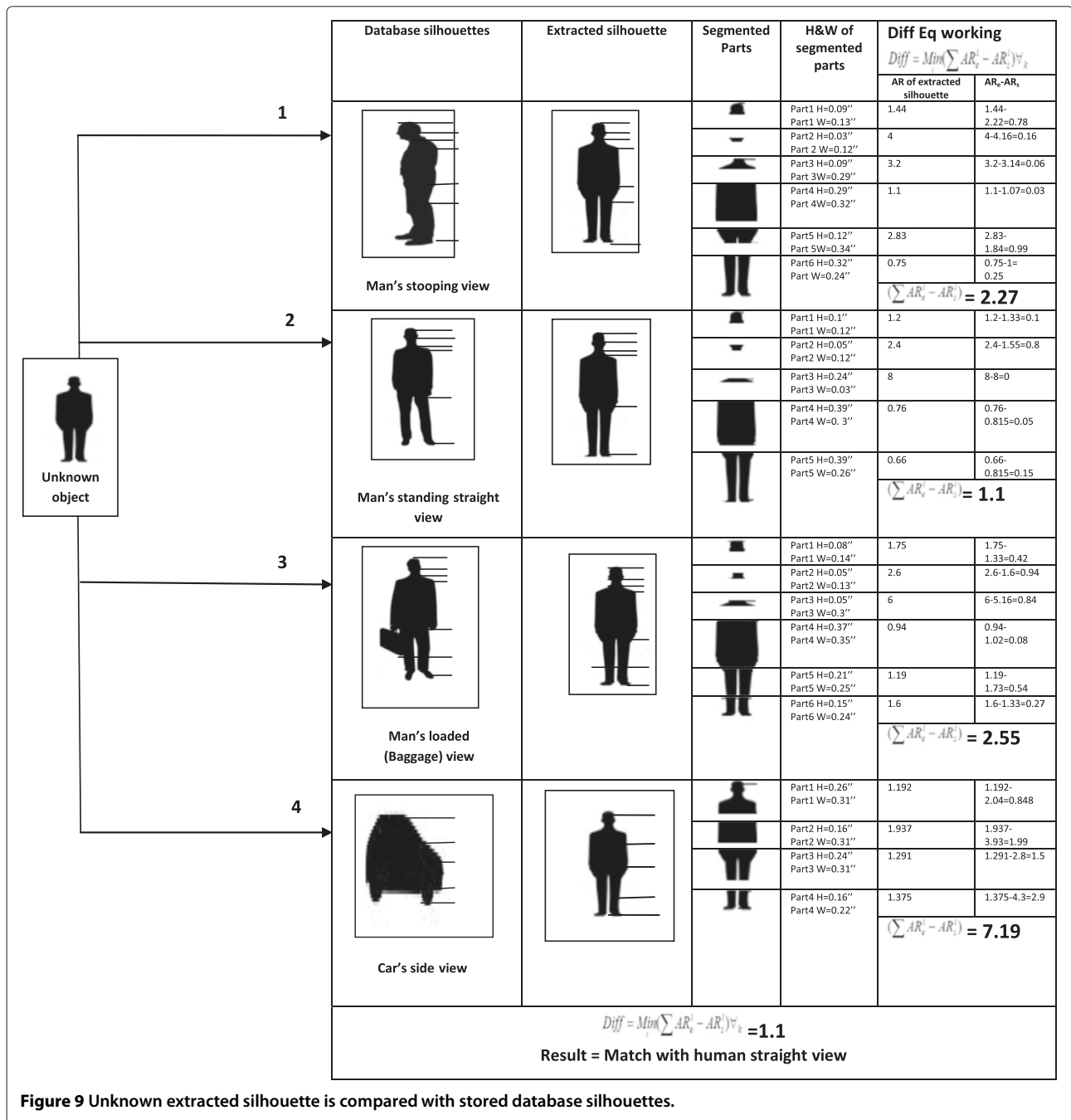


Figure 9 Unknown extracted silhouette is compared with stored database silhouettes.

compared to the aspect ratios of the corresponding stored silhouettes. For example, as seen in the database section, if an extracted mobile object silhouette is compared with a stored human silhouette, it is segmented into four parts before comparison. The comparison procedure is based upon the following equation:

$$Diff = Min_j (\sum AR_e^i - AR_s^i) \forall_k, \quad (13)$$

where AR_e = aspect ratio of the extracted silhouette, AR_s = aspect ratio of the stored silhouette, j = total

number of segments of a silhouette, k = total silhouettes deployed on a SN.

This equation computes the aspect ratio difference between the extracted silhouette and stored silhouettes

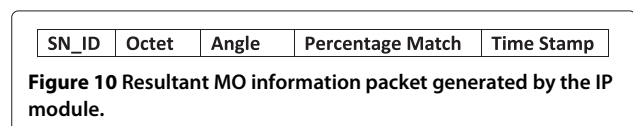


Table 14 Decision taken by seeing packet type

External module	Packet type	Destination decision
Sonar-sleeping module	Sleeping schedules	Send to ENs in a triplet
Sonar-sensing module	MOD message	Send to next hop SN only
Time synchronization	BEACON/TIME stamps	Send to subsequent INs only
Image processing module	MO recognition information	Send to next hop SN/base station

one by one and then returns the minimum difference value. This minimum difference shows that the extracted silhouette is similar to the stored silhouette from which it has minimum difference. The percentage match of extracted silhouette with the most matched silhouette is computed by applying the equation below:

Percentage match

$$= \frac{\sum_{i=0}^{\text{total segments}} (\text{AR of extracted silhouette})}{\sum_{i=0}^{\text{total segments}} (\text{AR of stored silhouette})} \times \left(\frac{1}{k}\right) \times 100. \quad (14)$$

The proposed scheme can be best described by an example scenario as shown in Figure 9. In this figure, an unknown extracted silhouette of a mobile object is compared against four different stored silhouettes of a human on a SN. The run time-extracted silhouette is segmented into the same parts and with the same aspect ratios one by one as the corresponding silhouette in the database. For example, when this unknown extracted silhouette is compared with a man’s standing side view, then the unknown object is segmented into six parts, and when this is compared by armed view, then it is segmented into five parts and with same width and height ratios.

After completing the silhouette comparison operation, the IP module creates the packet as shown in Figure 10. It may be noted that except for the surety, all the fields are retrieved from the database.

Routing decision module The routing decision module can be considered as a central unit as the subsequent behavior and desirable performance of network depend on the outcome generated by it. It is responsible for taking decisions regarding the destinations for the packets generated by the modules. The decision module is invoked

by sonar-sleeping, sonar-sensing, time synchronization, and image processing modules. The operations of decision module for respective instances of all these modules are discussed below. The nature of decisions taken by the module varies across the modules as can be seen in Table 14.

The decision module can be invoked by sonar-sleeping module and takes destination decisions for sharing the sleeping schedules generated by the background EN in a triplet for the foreground ENs. The decision module then takes decision about the destination of the MOD message that is generated by the sonar-sensing module. The MOD message invokes the most optimal SN’s camera in the network for in-time MO image frame capturing and recognition. The decision about the most optimal SN depends on the MO direction that is determined by capturing the MO image at Carrefour IN. The MOD message first goes to the Carrefour IN which then sends it to the next optimal sensor node by seeing the direction of the MO silhouette after recognizing it.

The decision module receives beacons from the time synchronization module, makes a decision about beacon destination, and sends it to the subsequent inner nodes by consulting the ‘SNs’ positions and camera orientations table’. The decision module is also responsible for taking the destination decision about the MO information packet generated by the image processing module. This packet is sent to the next hop sensor node or to the base station by seeing the percentage match of the mobile object. If the percentage match exceeds the user specified threshold, then this packet is sent to the base station, triggering termination of the successful recognition process.

4 VISTA performance evaluation

In this section, we evaluate the performance of VISTA with respect to the proposed IP algorithms. Simulations

Table 15 System specifications for performance evaluation

Feature	Specification
Processor	Atom
Processor model	Intel N2800
Memory	2Gb
Camera	Web camera Logitech Tessar
Processor speed	1.28 GHz
Dimension of the image	240 × 240 pixels

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, non-uniform

Case #	Original BG	Original FG	BG After DCT	FG After DCT	BG After BG Sub	FG After BG Sub	Extracted silhouette	Extracted silhouette parts
Case 1								
Case 2								
Case 3								
Case 4								
Case 5								
Case 6								

Figure 11 Evaluation of VISTA algorithms.

have also been carried out in NS-2 that observes the impact of database management and distribution on overall performance of VISTA.

4.1 VISTA IP algorithms performance on testbed

All the image processing algorithms for low form-factor SNs are implemented and tested using the Atom processor with Matlab (version R2012b) to emulate incremental change in computational load (i.e., added current load), similar to the work of [20]. The system specifications are given in Table 15. The experiments are done under various light conditions by using different background and foreground types. The parameters and situations in which experiments are done are given in Table 16. Figure 11

shows the results of VISTA algorithms executed in different situations. For example, case 1, case 3, and case 5 are executed with different backgrounds and object types, while case 3 and case 4 are executed with the same object type (human) but with different background types. Case 3 is executed for uniform background, while case 4 is executed for non-uniform background. Case 2 uses the same object type as case 1 but with different angle. The performance of VISTA is practically justified in all the above cases. Case 6 is executed with object type (human) and in bad light conditions (artificial light). In this case, VISTA algorithms could not segment the neck of the human because of low lighting conditions of the captured image that affects the overall performance.

Table 17 Memory consumption comparison after SICS

Elapsed time	DCT compression and reconstruction (seconds)	Memory consumption before compression (KB)	Memory consumption after compression (KB)
Time 1	1.242557	12	8
Time 2	1.242957	12	8
Time 3	1.286239	12	8

Table 18 Comparison of memory consumption after applying image subtraction algorithm

Case number	Detected region	Reduction in image dimensions	Elapsed time (s)
1	Change in whole image	240 × 240	0.438973
2	Change in upper half horizontal	240 × 120	0.111694
3	Change in second half vertical	120 × 240	0.101043

We have evaluated VISTA algorithms against baseline algorithms with respect to time and memory efficiency which subsequently lead to energy efficiency on battery-operated nodes. It is important to mention here that the proposed algorithms are compared against baselines because our algorithms are actually part of an integrated architecture in which each is tailored to achieve synergy, which is not the case in contemporary literature. Imagery data produced and subsequent memory usage is reduced significantly after applying compression through SICS algorithm as shown in Table 17. Image compression results into memory savings that deteriorate the image quality; however, the image stays extractable.

The image subtraction sub-module takes part in reducing image data and execution time as shown in Table 18. The image subtraction module reduces imagery data for further processing and controls energy consumption.

Table 19 shows the total time elapsed between pixel-by-pixel comparison and proposed FRILL scheme for MO recognition. It shows that there is a large difference in execution time when applying both schemes on the same silhouette.

4.2 VISTA performance evaluation using NS-2

4.2.1 VISTA simulation parameters

Simulations are carried out in NS2 in order to evaluate the performance of VISTA regarding the effects of network topology, mobility, and distribution of database. NS2 is a discrete event simulator targeted at networking research and provides substantial support for simulation of TCP, routing, and multi-cast protocols over wired and wireless networks [39]. Specifically, NS2 validates the performance of VISTA by representing the confidence (surety) of MO identification as a function of number of cameras, database size and distribution, MO's trajectory, stored

Table 19 Comparison of pixel-by-pixel operation and FRILL

Elapsed time	Pixel-by-pixel comparison (s)	Silhouette extraction comparison (s)	Difference (s)
Time 1	43.878	1.46	42.418
Time 2	42.108	1.23	40.878
Time 3	44.124	1.48	42.64

perspectives, and network depth. Some of the simulation parameters are described below:

1. Manhattan model: We use the Manhattan mobility model in a field of 60 m × 60-m with 16 nodes. There is a FV camera at each SN that is placed considering the location from where MO can be optimally observed.

2. Database distribution: The database is deployed at each SN considering the view of camera and application perspective. At each SN, relevant silhouettes along with their octets, angles, sizes, and surety levels are stored in the form of table. In our simulations, we consider four angles, i.e., left, right, front, and back and four sizes. We represent this using four bits for a total of 16 combinations. The surety level is affected by both parameters, i.e., angle and size.

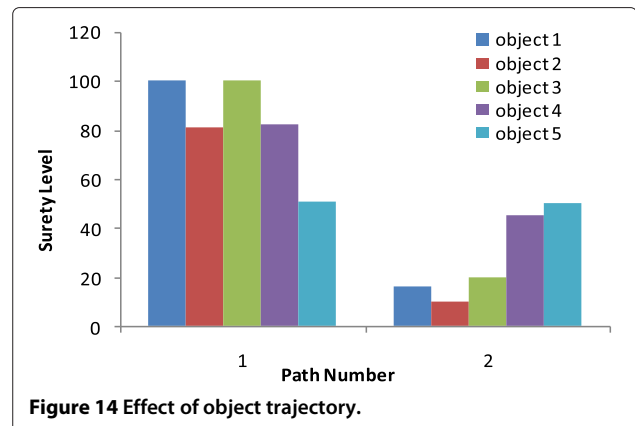
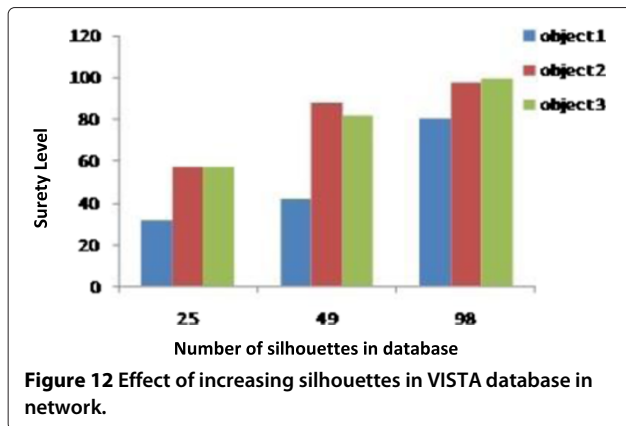
3. MO generation: A MO is generated and inserted into the RoI with fixed seed. The route of the MO which affects its surety level is programmed through a pseudo-random function that controls its direction and distance. When the MO enters the sensing range of a node, the node generates a packet containing octet, angle, size, and the surety level. If the resultant surety level is less than the threshold, the packet generated by the node is passed onto the next suitable node. The camera at the next node is similarly activated only when MO enters its range, so that redundant image capturing and processing is prevented.

4.2.2 VISTA results based on NS-2 simulation

Number of stored silhouettes in database

When the total number of stored silhouettes in network is less, the surety level of the detected object is also low. As the number of silhouettes stored in a database is increased, the surety level of object increases. This means that a large number of silhouettes stored in database results in a better chance of match at every SN. Initially, with only 25 silhouettes distributed across the network, it is observed that the maximum surety level of MOs varies between 30% and 60%. When number of stored silhouettes is increased twofold, the surety level increases to 40% to 80% and when the stored silhouettes are increased four times, the surety level increases to around 80% to 100%. Based on the MO trajectory, the variations in the surety level as a function of the number of silhouettes varies for three objects as shown in Figure 12. This shows that for MO 3, the surety level in case of 25 and 98 stored silhouettes is highest; the reason behind this is the default bias which exists in the used mobility model.

Depth of network in hop count In order to examine the impact of network depth on the percentage match of a MO, we observed the surety level while seeing the increase in the hop count as the MO follows the trajectory. The results as in Figure 13 show that as MO enters the RoI, the surety level of detection is low since a limited set of SNs



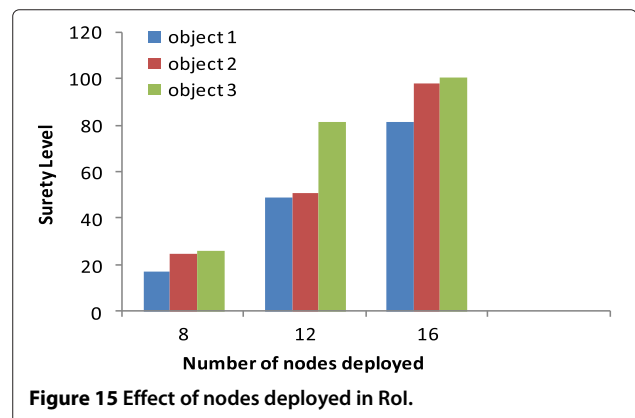
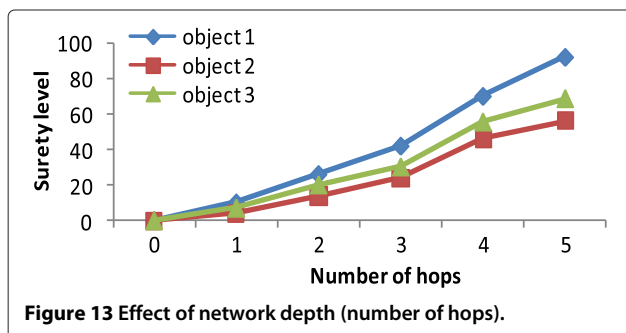
that observe the MO may or may not have its silhouettes in their databases. However, as MO goes deeper into RoI, the probability of finding an exact match to the silhouettes stored in distributed database increases since the number of SNs observing the MO increase. It is also important to note that though the graph is expected to be linear, it only tends towards linearity. This owes to the fact that the database is not uniformly distributed across the entire RoI.

Object trajectory Object trajectory or path affects the surety level of a MO. Interestingly, different paths can have different factors that directly affect the probability of object identification, such as the number of SNs and the number of silhouettes stored in the database. Due to this reason, when two mobile objects traverse the same path, the surety levels of these mobile objects can be different, depending upon different underlying factors for each MO. This effect can be observed in Figure 14, which shows the surety levels for five mobile objects traversing path 1 and then surety levels for these mobile objects traversing path 2. Path 1 comprises six hops, while path 2 comprises three hops. The results show that there is more chance of object identification while traversing path 1, i.e., a path with more number of SNs resulting in larger number of mobile object images captured. Moreover, it can also be observed that the surety level of mobile object 5 remains

unaffected by change in path, which owes to the underlying reason that path 2 although having less number of SNs, has a larger number of silhouettes stored for this mobile object.

Number of nodes deployed in a RoI When a large number of SNs are deployed in RoI, it implies that a larger area is covered by cameras, and there are more perspectives per RoI. This results in increased surety levels of MOs. It is shown in Figure 15 that there is an increase in surety levels of MOs with an increase in the number of SNs. Initially, with only eight SNs deployed in RoI, the surety is around 20% to 30%. Keeping the RoI constant, when the SNs are increased in number by 50%, the surety increases to 50% to 80%. Likewise, when SNs deployed in the same RoI are increased to 16, the surety is 80% to 100%.

Number of silhouettes of an object Figure 16 shows that as the number of stored silhouettes of an object increases across the network, the surety level also increases. A large number of stored silhouettes imply that the probability of matching for a MO increases. It is important to note that the nature of the MO itself is critical to define the number of its silhouettes that vary either in perspective or in



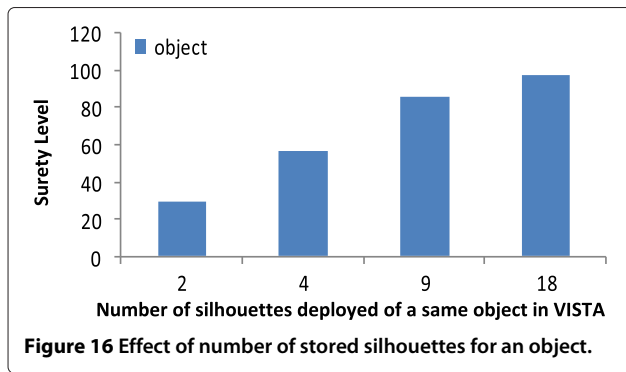


Figure 16 Effect of number of stored silhouettes for an object.

size or in both. For example, consider the MO to be a tank. Since in this case, the size of the MO is too large, deploying too many silhouettes of varying sizes is not needed because a tank at one edge of the road or at the other edge does not change in size considerably. However, multiple perspectives of tank are needed because it is highly agile in changing directions.

Centralized vs. distributed database There can be two types of database deployment strategies: centralized and distributed. In centralized distribution, images are equally deployed in the network without taking into consideration expected paths of different objects. In distributed strategy, the deployment of images is based on the type of object, i.e., if a path is more likely to be traversed by certain type of objects, the SNs in that path are deployed with more images of that type of object. In such case, the expected path complaint distribution of database distribution often serves to be of advantage. Also, there is a need to highlight that the object may not follow the expected trajectory. In that case, this strategy could be detrimental. Figure 17 shows that when an object traverses the expected path, the surety level is 100% (left blue). But when there is path violation, the surety level is 0% (missing red on the left side). On the contrary, pseudo path with independent deployment yields large surety level for random path traversal. In

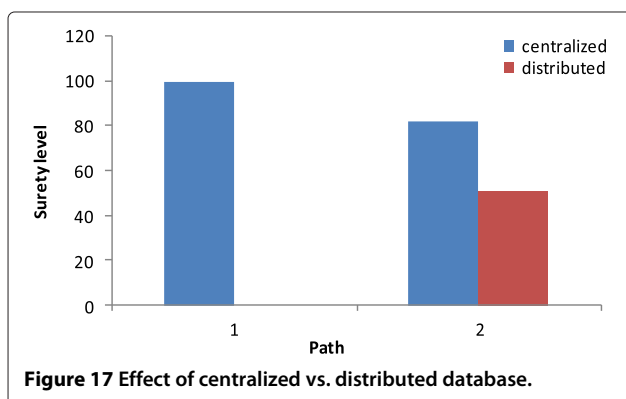


Figure 17 Effect of centralized vs. distributed database.

case the object travels the expected path, the surety level increases slightly.

5 Discussion

False alarms or no alarms (in case of MO present) can be generated by VISTA layers as shown in Figure 18. Four combinations of events and alarms can be generated that are described in Table 20 showing different situations in which generation of false alarms can affect the performance of VISTA from two aspects, namely, the target and the camera.

When the target reduces or increases its speed suddenly, either it stops moving further or it gets a turn of 180° or higher. Under these conditions, our algorithm might not work where the target may not lie within the predicted yaw angle. To address this problem, the concept of timer may be introduced, i.e., the activated IN at the predicted target location sets a timer. The IN waits for the target until the timer is expired. When the timer is expired, this IN activates all other INs in its one-hop neighborhood and reports this ‘target lost’ error back to the previous node (IN or EN), from which it received the mobile object detected message. On the reception of this target lost error, the previous node also activates all other INs in its one-hop neighborhood. Hence, all the INs are activated near the location, where the target was last seen and got lost. Once the lost target is found, then the target is tracked by the reporting IN and all other activated INs switch to the sleep state. Using this mechanism, VISTA recovers from the anomalous behavior of the target and resumes its normal operation. Similarly, another cause of false alarm can be due to packet loss and corruption which may be compensated through reliable communication using acknowledged service. Regarding the sub-optimal deployment of cameras, auto-calibration of frustums can be achieved by allowing cameras to collaborate through sharing their ‘experiences’ of detecting an object with a certain level of surety at an angle.

		Alarm	
		+ve	-ve
Event	+ve	+ve +ve	+ve -ve
	-ve	-ve +ve	-ve -ve

Figure 18 False-alarm generation by VISTA. Description +ve +ve: Target present and VISTA activated. +ve -ve: Target present but VISTA not activated (false alarm), -ve +ve: Target not present but VISTA activated (false alarm). -ve -ve: Target not present and VISTA not activated.

Table 20 Anomalies leading to operational degradation and false alarms in VISTA

Number	Anomaly	Affected VISTA layer/function	False alarm	Compensation
1	Small bending curvature sudden stop acceleration/ deceleration of target	Image capture sub-module	False alarm (-ve +ve)	Timer-based deactivation and using reverse link denotification, and using Carrefour cameras to re-initiate MOD message
2	Packet loss/corruption (sleep schedule, time sync)	Sonar-sleeping module, sonar-sensing module, time synch module, and image capture sub-module	False alarm (+ve -ve)	Reliable communication using ACK, ReTx, and FEC
3	Misalignment/dealignment of cameras (art gallery problem)	Image capture sub-module, image change detection sub-module, and image subtraction sub-module	Frustum inconsistencies such as over laps and holes lead both to (-ve +ve) and (+ve -ve)	Automated camera calibration by tallying recent tuple (octet, angle, surety) similar to [40] to [41]

An insight is presented in Table 21 regarding specific operations of VISTA that yield energy efficiency but introduce additional complexities to achieve the performance. For example, the energy efficiency achieved through ‘just-in-time sensing otherwise sleeping’ behavior of sonars introduces latency and possible missouts. Application-specific and target type-based adjustment to pulse repetition rate at sonars can reduce unacceptable latencies and alleviate unexpected missouts. A flexible silhouette re-deployment mechanism can prove to be effective in situations where target speed and ambulatory behavior deviate from the envisaged mobility model.

6 Conclusions

In this research, a novel architecture is proposed in which energy-efficient capturing and processing of MO image is done. The architecture achieves its objective by silhouette recognition of mobile targets with images stored in the database deployed at each SN of the sensor network. The proposed architecture redefines the video capturing capability of VSNs. The SNs are placed at optimal positions in order to make communication effective. The proposed architecture assists VSN in achieving a cumulative vision. Experimental evaluation of image processing algorithms of ‘VISTA’ against baseline algorithms with respect to

Table 21 Trade-offs of energy latency-leading to operational degradation and false alarms in VISTA

Number	Memory/energy efficiency in VISTA	Implementation layer	Trade-off	Compensation
1	Triplet formation of SONARs for reducing shutter ‘On’ time of IN cameras and breach avoidance amidst failing nodes	Sonar-sensing module	SONAR 3-pulse detection introduces latent response of INs causing false alarm (+ve -ve)	Pulse repetition rate (PRR) in SONAR tailored to match target speed and range [42]
2	Sonar sleep scheduling to conserve energy	Sonar-sleeping module	False alarm (+ve -ve) amidst holes in full-width half maximum (FWHM) coverage	Automated SONAR calibration by triplet training similar to [42]
3	Database with relative silhouettes only (mobility model compliant) to reduce (a) memory occupancy, (b) comparison computations, and (c) frequency of IP module usage	Image change detection sub-module and image subtraction sub-module	Tight coupling with assumed mobility model can lead to both false alarms	Silhouette distribution and confiscation dynamically associated with per path cumulative surety can reduce the effect of mobility
4	Image change detection triggers subsequent IP modules (need-based computations only)	Image change detection sub-module	Slow moving objects become immersed in scene causing false alarm (+ve -ve)	Using GMM [11] alleviates foreground becoming background
5	Surety-based transcoding	Image compression and storage sub-module	Higher transcoding neutralizes the job of subsequent IP modules	Benchmark change detection and subtraction against acceptable transcoding levels

execution time and memory shows significant reduction in image data and memory occupancy. The deployment of sonar and SNs is analyzed by NS2 simulations that are performed in order to assess the performance of the proposed architecture when the number of cameras, database size and distribution, object's trajectory, stored perspectives, and network depth are varied. The simulations in NS2 show that surety level of object increases with larger database. Also, when MO goes deeper into RoI, the probability of finding an exact match to the outlines stored in distributed database increases. Similarly, different paths can have different factors that directly affect the probability of object identification, such as the number of SNs deployed in the RoI and number of silhouettes stored in database; therefore, when two MOs traverse the same path, the surety levels of these MOs can be different, depending upon different underlying factors for each MO. When a large number of SNs are deployed in a region, larger area is effectively covered by cameras and more perspectives per region are available. This results in increased surety level of MOs.

Abbreviations

BIC, Bayesian information criterion; CDV, cooperative distributed vision; EN, edge node; FOV, field of view; FRILL, feature-dependent silhouette segmentation for low-energy comparison; FV, fixed view; GMM, Gaussian mixture model; IN, inner node; IP, image processing; MO, mobile object; MOD, mobile object detected; Ool, object of interest; PTZ, pan-tilt-zoom; RoI, region of interest; RSSI, received signal strength indicator; SICS, surety-based image compression and storage; SIFT, scale invariant feature transform; SMAC, sensor medium access control; SN, sensor node; VISTA, achieving cumulative vision through energy efficient silhouette recognition of mobile targets through collaboration of visual sensor nodes; VSNs, visual sensor networks; WSNs, wireless sensor networks; WYSIWYD, what you see is what you do.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

SJ has defined the main architecture and developed the manuscript. AHA conceived the idea and supervised the research. SZ helped refine the manuscript and fine-tuned the details in it. MMQ contributed in camera deployment, calibration, and alignment. MH contributed in database design and packet format. All authors read and approved the final manuscript.

Acknowledgements

The authors would like to thank Ms. Noureen Jabbar and Ms. Maryyam Muhammad Din for their participation in obtaining practical results for this research. The authors also would like to acknowledge the Al-Khawarizmi Institute of Computer Science and University of Engineering and Technology (UET) Lahore for supporting by sparing SJ to carry out the research.

Author details

¹Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology (UET), Lahore 54890, Pakistan. ²Department of Electrical Engineering, University of Engineering and Technology (UET), Lahore 54890, Pakistan. ³Department of Computer Science and Engineering, University of Engineering and Technology (UET), Lahore 54890, Pakistan.

Received: 18 June 2013 Accepted: 30 May 2014
Published: 25 June 2014

References

1. M Chen, S Gonzalez, VC Leung, Applications and design issues for mobile agents in wireless sensor networks. *IEEE Wireless Commun.* **14**, 20–26 (2007)
2. B Nelson, PK Khosla, Integrating sensor placement, visual tracking strategies, IEEE International Conference on Robotics and Automation. San Diego CA, 1351–1356 (1994)
3. LE Navarro-Serment, JM Dolan, PK Khosla, Optimal sensor placement for cooperative distributed vision, in *Paper presented at the IEEE international conference on robotics and automation (ICRA), vol. 1* (New Orleans, LA, USA, 26 April–May 1, 2004), pp. 939–944
4. F Capezio, F Mastrogiovanni, A Sgorbissa, R Zaccaria, Robot-assisted surveillance in large environments. *J. Comput. Inform. Technol.* **17**, 95–108 (2004)
5. KS Bhat, M Saptharishi, PK Khosla, Motion detection and segmentation using image mosaics, in *Paper presented at the IEEE international conference on multimedia and expo (ICME), vol. 3* (NY, USA, 30 July – 2 Aug, 2000), pp. 1577–1580
6. M Saptharishi, JB Hampshire, PK Khosla, Agent-based moving object correspondence using differential discriminative diagnosis, in *Paper presented at the IEEE conference on computer vision and pattern recognition, vol. 2* (SC, USA, 13–15 June, 2000), pp. 652–658
7. N Ukita, T Matsuyama, Real-time cooperative multi-target tracking by communicating active vision agents. *Comput. Vis. Image Underst.* **97**(2), 137–179 (2005)
8. T Matsuyama, *Cooperative distributed vision: dynamic integration of visual perception, action, and communication. Mustererkennung 1999*. (Springer Berlin Heidelberg, 1999), pp. 138–151
9. SC Tien, YLu TL Chia, Using cross-ratios to model curve data for aircraft recognition. *Pattern Recognit. Lett.* **24**(12), 2047–2060 (2003)
10. S Soro, W Heinzelman, A survey of visual sensor networks. *Adv. Multimedia* (2009). doi:10.1155/2009/640386
11. C Stauffer, WEL Grimson, Adaptive background mixture models for real-time tracking, in *Paper presented at the IEEE Computer Society conference on computer vision and pattern recognition, vol. 2* (CO, USA, 23–25 June 1999
12. AT Ihler, Fisher III JW, Moses RL, Willisky AS, Nonparametric belief propagation for self-localization of sensor networks. *Select. Areas Commun. IEEE J.* **23**(4), 809–819 (2005)
13. Rabbat M, *R Nowak, Distributed optimization in sensor networks*, CA, USA, (Unknown Month 26), pp. 20–27
14. Y Cho, SO Lim, HS Yang, Collaborative occupancy reasoning in visual sensor network for scalable smart video surveillance. *IEEE Trans. Consum. Electron.* **56**(3), 1997–2003 (2010)
15. TH Tsai, CY Lin, Exploring contextual redundancy in improving object-based video coding for video sensor networks surveillance. *IEEE Trans. Multimedia.* **14**(3), 669–682 (2012)
16. Ellis T, Multi-camera video surveillance, in *Paper presented at the 36th annual international Carnahan conference on security technology* (Atlantic City, NJ, USA, 20–24 Oct, 2002), pp. 228–233
17. L Paletta, S Wiesenhofer, N Brandl, O Sidla, Lypetsky Y, Visual surveillance system for monitoring of passenger flows at public transportation junctions, in *Paper presented at the IEEE intelligent transportation systems* (Austria, 13–16 Sep, 2005), pp. 862–867
18. M Kumar, *Automating visual sensor networks*, (Brown University, 2009)
19. V CB Margi, K Petkov, R Obraczka, Manduchi, Characterizing energy consumption in a visual sensor network testbed, in *Paper presented at the 2nd international conference on testbeds and research infrastructures for the development of networks and communities (TRIDENTCOM)* (Spain, March, 2006), pp. 1–3
20. R CB Margi, K Manduchi, Obraczka, Energy consumption tradeoffs in visual sensor networks, in *Paper presented at the 24th Brazilian symposium on computer networks (SBRC), vol. 1* (Curitiba, Brasil 29 May–2 June, 2006)
21. F Qureshi, D Terzopoulos, Smart camera networks in virtual reality. *Proc. IEEE.* **96**(10), 1640–1656 (2008)
22. F Qureshi, D Terzopoulos, A simulation framework for camera sensor networks research, in *Paper presented at the 11th communications and networking simulation symposium (ACM)* (Ottawa, ON, Canada, 13–16 April, 2008), pp. 41–48
23. FZ Qureshi, D Terzopoulos, Planning ahead for PTZ camera assignment and handoff, in *Paper presented at the third ACM/IEEE international*

- conference on distributed smart cameras (ICDSC), *Societ del Casino, Teatro Sociale di Como* (Como, Italy, 30 Aug–02 Sept, 2009), pp. 1–8
24. N Krahnstoever, Y Ting, L Ser-Nam, P Kedar, T Peter, Collaborative real-time control of active cameras in large scale surveillance systems, in *Paper presented at the workshop on multi-camera and multi-modal sensor fusion algorithms and applications (M2SFA2)* (Marseille, France, 18 Oct, 2008)
 25. Obraczka K, Manduchi R, Garcia-Luna-Aveces J, Managing the information flow in visual sensor networks, in *Paper presented at the 5th international symposium on wireless personal multimedia communications*, vol. 3 (Sheraton Waikiki, Honolulu, HI, USA, 27–30 Oct, 2002), pp. 1177–1181
 26. IF Akyildiz, T Melodia, KR Chowdhury, A survey on wireless multimedia sensor networks. *Comput. Netw.* **51**(4), 921–960 (2007)
 27. Akyildiz IF, Su W, Y Sankarasubramaniam, E Cayirci, Wireless sensor networks: a survey. *Comput. Netw.* **38**(4), 393–422 (2002)
 28. G Pei, M Gerla, X Hong, LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility, in *Paper presented at the 1st ACM international symposium on mobile ad hoc networking and computing* (Boston, MA, USA, 11 Aug, 2000), pp. 11–18
 29. Tong F, Tang W, Peng LM, Xie R, Yang WH, Kim YC, A node-grade based AODV routing protocol for wireless sensor network, in *Paper presented at the second international conference on networks security wireless communications and trusted computing (NSWCTC)*, vol. 2 (Wuhan, Hubei, China, 24–25 April, 2010), pp. 180–183
 30. A Marcus, O Marques, An eye on visual sensor networks. *IEEE Potentials.* **31**(2), 38–43 (2012)
 31. F Bai, A Helmy, A survey of mobility models. *Wireless Adhoc Networks*. University of Southern California, USA 206 (2004)
 32. E Onur, C Ersoy, Deliç H, How many sensors for an acceptable breach detection probability *Comput. Commun.* **29**(2), 173–182 (2006)
 33. W Ye, J Heidemann, D Estrin, An energy-efficient MAC protocol for wireless sensor networks, in *Paper presented at the IEEE twenty-first annual joint conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3 (Hilton, NY, USA, 25–27 June, 2002), pp. 1567–1576
 34. C Ford, I Stange, A framework for generalizing public safety video applications to determine quality requirements, in *Paper presented at the IEEE conference on multimedia communications, services, and security* (Krakow Poland, May, 2010)
 35. CC Loy, T Xiang, S Gong, Incremental activity modeling in multiple disjoint cameras. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(9), 1799–1813 (2012)
 36. Bashir AK, Akbar AH, Chaudhary SA, Hussain CS, Kim KH, Collaborative detection and agreement protocol for routing malfunctioning in wireless sensor networks, in *Paper presented at the 8th international conference on advanced communication technology (ICACT)*, vol. 1 (Phoenix Park, Korea, 20–22 Feb, 2006), pp. 327–332
 37. Chandra S, Ellis CS, Vahdat A, Managing the storage and battery resources in an image capture device (digital camera) using dynamic transcoding, in *Paper presented at the 3rd ACM international workshop on wireless mobile multimedia (ACM)* (Boston, MA, USA, 11 Aug, 2000), pp. 73–82
 38. L Xu, S Zhang, Z He, Y Guo, *The comparative study of three methods of remote sensing image change detection*, (Fairfax, VA, USA, 12–14 Aug, 2009), pp. 1–4
 39. K Fall, K Varadhan, “The ns Manual (formerly ns Notes and Documentation). The VINT Project, A collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC.” (2002). September 16
 40. KCT Aslan, R Bernardin, *Stiefelhofen, Automatic calibration of camera networks based on local motion features*, (France, 18 Oct, 2008)
 41. SN Sinha, M Pollefeys, Camera network calibration and synchronization from silhouettes in archived video. *Int. J. Comput. Vis.* **87**(3), 266–283 (2010)
 42. Z Zhang, *A Micro-Doppler Sonar for Acoustic Surveillance in Sensor Networks*. Ph.D. dissertation, (2009). The Johns Hopkins University

doi:10.1186/1687-5281-2014-32

Cite this article as: Jabbar et al.: 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* 2014 **2014**:32.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com