The Case for Multi–tier Camera Sensor Networks*

Purushottam Kulkarni, Deepak Ganesan and Prashant Shenoy

Department of Computer Science,
University of Massachusetts
Amherst, MA 01003

ABSTRACT

In this position paper, we examine recent technology trends that have resulted in a broad spectrum of camera sensors, wireless radio technologies, and embedded sensor platforms with varying capabilities. We argue that future sensor applications will be hierarchical with multiple tiers, where each tier employs sensors with different characteristics. We argue that multi-tier networks are not only scalable, they offer a number of advantages over simpler, single-tier unimodal networks: lower cost, better coverage, higher functionality, and better reliability. However, the design of such mixed networks raises a number of new challenges that are not adequately addressed by current research. We discuss several of these challenges and illustrate how they can be addressed in the context of SensEye, a multi-tier video surveillance application that we are designing in our research group.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture Design; C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems; I.4.9 [**Image Processing and Computer Vision**]: Communication Applications

General Terms

Algorithms, Design

Keywords

Video sensors, Multi–tier sensor networks

1. INTRODUCTION

The relentless pace of technological growth has led to the emergence of a variety of sensors and networked sensor platforms. Today, networked sensors span the spectrum of cost, form-factor, resolution, and functionality. As an example, consider camera sensors, where available products range from expensive pan-tilt-zoom cameras to high-resolution digital cameras, and from inexpensive webcams and "cell-phone-class" cameras to even cheaper, tiny cameras such as Cyclops [12]. A similar set of options are becoming available for sensor platforms, with choices ranging from embedded PCs to PDA-class Stargates [15], and from low-power Motes [11] to even lower power systems-on-a-chip [14]. Camera sensor networks—a wireless network of sensor nodes equipped with cameras—are useful in a variety of applications such as surveillance, intruder detection, live virtual environments, and online virtual tours.

Early work on sensor networks in general, and camera sensor networks in particular, assumed a single tier network of homogeneous sensors. However, with the spectrum of sensors available to achieve a particular task, we believe that the opportunity is ripe for designing *multi-tier, multi-modal networks*. By a multi-tier network, we mean that the sensors are organized hierarchically into multiple tiers. For instance, a two-tier surveillance application may consist of low power cameras at the bottom tier that trigger higher resolution cameras at the upper tier in an on-demand fashion. By a multi-modal network, we mean one where different sensing modalities are employed to achieve a particular task. Thus, rather than employing only camera sensors, a multi-modal surveillance application can be designed using motion, vibration, thermal imaging and camera sensors, all of which cooperate with one another to achieve a common goal.

The advantages of multi-tier multi-modal networks over single-tier unimodal networks are many: low cost, high coverage, high functionality, and high reliability. Depending on how they are designed, single tier systems often meet only a subset of these requirements. For instance, low cost can be achieved by using a single tier of inexpensive sensors but at the expense of functionality. High coverage can be achieved using a dense deployment of untethered sensors that can be placed anywhere but power considerations can sacrifice reliability. High functionality can be achieved by employing high fidelity sensors but at the expense of sacrificing coverage due to the high cost. Thus, a single choice along the axes of power, cost, or reliability will result in a sensor network that sacrifices one or more of the key requirements.

In contrast, multi-tier multi-modal networks (*henceforth,* M^2 *networks*) provide an interesting balance of cost, coverage, functionality, and reliability. For instance, the lower tier of such a system can employ cheap, untethered elements that can provide dense coverage with low reliability. However, reliability concerns can be mitigated by seeding such a network with a few expensive, more reliable sensors at a higher tier to compensate for the variability in the lower tier. Similarly, a mix of low-fidelity, low-cost sensors and high-fidelity high-cost sensor can be used to achieve a balance

*This work was supported in part by National Science Foundation grants EEC-0313747, CNS-0219520 and EIA-0098060.

NOSSDAV'05, June 13–14, 2005, Stevenson, Washington, USA.

Node	*Power*	*Cost*	*Capability*
Cyclops	46mW	Unpriced	Fixed angle lens, 352x288 at 10fps
CMUCam	200mW	$50 (only camera)	Fixed angle lens, 352x288, up to 60 fps
Web-Cam	200mW	$50	Auto-focus lens, 640x480 at 30 fps
High-end PTZ Camera	1W	$1000	Retargetable pan-tilt-zoom lens, 1024x768 up to 30 fps

Table 1: Technology Trends in Cameras

Figure 1: Cyclops low-power camera sensor.

between cost and functionality. Application performance can also be improved by exploiting alternate sensing modalities that may reduce energy requirements without sacrificing system reliability.

In this position paper, we argue that the design of M^2 sensor networks raise a number of research challenges that are not fully addressed by the existing literature on single-tier sensor networks. We begin by presenting recent trends in camera, processor, radio, and storage technologies and then discuss several research challenges the arise in the design of M^2 networks. Finally, we illustrate how these challenges are being addressed in the context of *SensEye*, a multi-tier video surveillance application that is being designed in our research group.

The rest of this paper is structured as follows. Section 2 presents recent technology trends for various components of a sensor network. Section 3 presents the system model for a typical M^2 network. Section 4 discusses research challenges in designing such networks, and Section 5 presents an overview of our SensEye camera sensor application.

2. TECHNOLOGY TRENDS

A rapidly growing market for wireless embedded devices has spawned a revolution in low-power processors, sensors, radios, and flash memory storage. In this section, we review some salient characteristics of embedded hardware that are available today, and review how different technology trends impact research on camera sensor networks.

Camera sensors: Table 1 reviews four classes of cameras that are available today either as prototypes or as commercial products. At the lowest end of the spectrum are tiny Cyclops [12] (shown in Figure 1) that consume a mere 46mW and can capture low resolution video. CMU-cams [13] are cell-phone class cameras with on-board processing for motion detection, histogram computation, etc. At the high-end, web-cams can capture high-resolution video at full frame rate while consuming 200mW, whereas pan-tilt-zoom cameras are retargetable sensors that produce high quality video while consuming 1W.

Sensor Platforms: A variety of sensor nodes have emerged in the last few years, from the resource–constrained Mica Motes [11] to intermediate platforms such as the Yale XYZ [8], to larger PDA-class platforms such as the Intel Stargate [15]. Table 2 compares the power consumption and the available processing, memory and storage resources on these platforms.

The Mica Motes are highly resource–constrained and very low–power, and hence are only suitable for simple sensing and detection tasks. The Yale XYZ platform is more capable and has an order of magnitude more memory and processing resources than the Mote. However, it consumes roughly 3 times the power of the Mica mote at the highest frequency setting. These nodes can be used for simple object identification and target localization. At the higher end of the spectrum are PDA-class devices such as the Stargate, which are an order of magnitude more powerful than the intermediate nodes but also consume an order of magnitude more power. These nodes can perform complex tasks such as object identification or resource–intensive tasks such as video streaming.

Different nodes use different embedded processors to suit their requirements. The choice of processor is typically a function of the per-instruction energy efficiency as well as sleep and wakeup power consumption. A wide range of embedded processors are available today that use dynamic voltage and frequency scaling techniques for low power consumption. Processing costs (joules per instruction) are roughly two to three orders of magnitude lower than communication costs (joules per bit) on available embedded platforms such as Mote and Yale XYZ.

While the power consumption of radio communication has decreased in recent years, achievable gains have been limited by the physics of radio propagation and the overhead of signal processing circuitry. As shown in Table 2, at the lowest end of the power spectrum are low bit-rate radio technologies such as 802.15.4 (Zigbee), which consume roughly 50mW and can transmit at 250Kbps, whereas higher end 802.11 radios consume more than 1W but can transmit at 54Mbps.

Finally, the use of storage on sensor nodes is an important but less studied aspect of sensor networks. The costs of flash memory has plummeted and it is possible today to purchase a 1 GB flash card for less than $100. In addition, newer flash memory chips are very efficient energy-wise for writes and erase operations and are roughly two orders of magnitude less expensive than communication over the radio. This makes them ideal for archival and caching of video data at sensor nodes.

These technological trends make a strong case for designing camera sensor networks that comprise a mix of tethered and untethered, low–power and high–power, resource–constrained and resource–rich devices.

3. SYSTEM MODEL

Given the above spectrum of hardware choices, today's system designers are no longer constrained to a homogeneous network of sensor nodes. We envision that future sensor networks will be organized hierarchically into multiple *tiers* (see Figure 2). The sensing devices, radios, processors, and the nodes within each tier are assumed to be homogeneous with respect to their sensing, processing, storage and memory capabilities. Different tiers are assumed to be heterogeneous with respect to their capabilities and sensing modalities. To illustrate, a set of Motes attached to vibration sensors could constitute one tier, while Stargates equipped with web-cams could be form another tier. Even when different tiers employ the *same* sensing modality, they are assumed to employ sensors with significantly different tradeoffs along the cost, power and reliability axes. For instance, three different tiers could employ three different types of camera sensors: Cyclops cameras, web-cams, and high-end pan-

Sensor Platform	*Power*	*Processor*		*Radio*		*Resources*
		Type	*Active Power*	*Type*	*Transmit Power*	
Mica Mote	84mW	Atmega128 (6MHz)	24mW	802.15.4	50mW	4KB RAM, 512KB Flash.
Yale XYZ	240mW (at 57MHz) and 50mW (at 2MHz)	OKI Arm Thumb processor 2MHz to 57MHz dynamic frequency scaling	7mW (@2MHz)- 160mW (@57MHz)	802.15.4	50mW	32KB on-chip RAM, 2MB external RAM
Stargate	700mW	XScale PXA255 processor 100MHz-400MHz	170mW (@200MHz) to 400mW (@400MHz)	802.11	$>$1W	32MB RAM, Flash CF card slot

Table 2: Technology Trends in Sensor Platforms

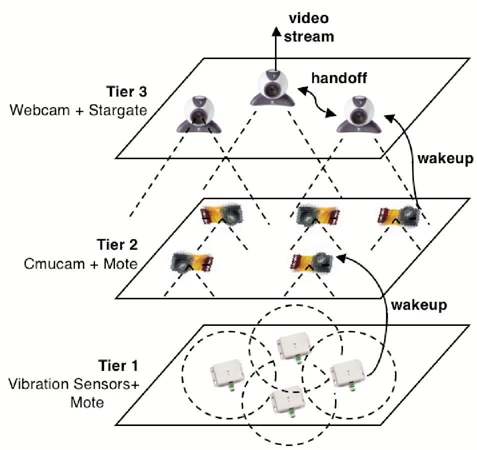

Figure 2: Example of a three-tier camera sensor network.

tilt-zoom (PTZ) cameras.

The various tiers are assumed to interact and coordinate in different ways to achieve application goals. Interactions between tiers can be *spatial* or *temporal* in nature. Further, there can be *parent-child* interactions between tiers as well as *peer* interactions within a tier. For instance, a node equipped with a low-power camera that wakes up a node with a high-power camera constitutes a temporal parent-child relationship. In contrast, multiple low-power cameras that coordinate with one another to exploit redundancy and wake-up alternately to save power represent a spatial peer relationship.

Nodes within and across tiers are assumed to communicate using their wireless radios. General-purpose message-oriented communication within and across tiers can employ existing multi-hop routing protocols such as Directed Diffusion [7]. In the context of a camera-based sensor network, we assume additional network protocols that support streaming and handoffs in hierarchical, ad-hoc collection of sensors.

With this background, we now present research challenges in designing M^2 networks.

4. CHALLENGES IN DESIGNING M^2 NETWORKS

This section articulates four research challenges in designing M^2 networks: (i) exploiting multiple tiers to achieve various design tradeoffs, (ii) exploiting multiple sensing modalities, (iii) protocol support and dynamic resource management for multi-tier interactions, and (iv) programming abstractions.

4.1 Design Tradeoffs in Multiple Tier Networks

M^2 sensor networks have the potential to provide a mix of low-cost, long lifetime, high coverage, high functionality and high reliability. A core challenge in designing such networks is to build a highly tunable system that can be used to achieve any particular point solution along the above-mentioned dimensions. Further, many applications have multiple objectives, making it harder to find the best design point for an application. For instance, long-lifetime and high reliability is often required in many ad-hoc surveillance applications. Such dual-constraint optimization is difficult when the optimization criteria conflict with one another. In this section, we describe instances of design tradeoffs, and how the system can be optimized for them.

Maximizing Lifetime and Reliability: Maximizing lifetime across a multi-tier network involves exploiting sensing resources at the *least* energy-constrained tier to compensate for the lack of energy resources at more constrained tiers. While duty-cycling techniques can be employed to conserve energy, this can also result in lower reliability, since fewer nodes are actively sensing at any given time. Achieving dual goals of lifetime and reliability involve two key challenges: (a) placement of tiers to achieve maximally redundant coverage, and (b) duty-cycling tiers to minimize energy costs. While sensor placement strategies (eg: [4, 16]) as well as duty-cycling techniques (eg: [18]) have been studied extensively in recent years, they have considered single tier sensor networks with circular sensor coverage models. M^2 networks pose new challenges since sensor modalities can differ widely in their range and directionality properties and tiers can have different coverage properties.

Maximizing Lifetime and Minimizing Latency: Latency is an important criteria in sensor-based monitoring applications since it impacts the timeliness of response to events. Optimizing latency requirements in a single tier application is determined by two factors– the latency of detecting an event and latency of routing the event notification to a proxy or base-station. The multi-tier case is more complex and involves several possibilities at each tier. Choices need to be made about where a particular block of data processing should be performed. For instance, object recognition in video surveillance is often resource–intensive since it involves searching through a library of images to find the best match. Consider the case when a resource–constrained sensor has detected an object and needs to decide whether to perform the object recognition locally or transmit the data and let a more resource–rich node perform the task. If performed locally, processing can incur significant latency but will consume low power, since computation is significantly cheaper than communication. Alternately, if data is transmitted to and processed at a higher-tier node, then latency can be lower but energy costs will be higher due to greater communication overhead. Thus, optimizing both lifetime and latency involves multiple, conflicting challenges. In general, processing tasks should be split between different tiers such that total event notification latency satisfies the timeliness requirements of the application. Further, the most energy-efficient option that does not violate latency requirements should be selected to maximize lifetime. Finally, sleep schedules of nodes should optimize the amount of time for which radios are switched off, again without violating application latency requirements.

While we have discussed only two design tradeoffs for multi-tier networks, many other tradeoffs need to be addressed. For instance, users will require a variety of functionality from camera sensors. Some applications will require streaming live video of events, whereas others may require target snapshots; some might require real-time streaming or notification, while others may be satisfied with archival and post-facto data retrieval. Thus, different sets of tradeoffs arise in different applications, and these will need to be addressed by making appropriate design choices.

4.2 Exploiting Multiple Sensing Modalities

While multi-tier networks can provide numerous benefits, they are fundamentally restricted by the limitations of the sensor modality at each tier. The use of different sensor modalities can often provide gains along orthogonal axes. Consider a surveillance application comprising video sensors. These sensors can detect a target only when it is within visual range. However, a vibration sensor can be used to provide early warning since it can detect vibrations even if the target is beyond visual range. Different sensor modalities can provide benefits from an energy perspective as well. The sampling costs of different sensors vary significantly. A video sensor is considerably more expensive energy-wise than an acoustic sensor, which is more expensive than a vibration sensor. The different sensing costs can be exploited so that detection can be performed with cheaper sensors, which then trigger more expensive ones. A drawback of employing multiple sensing modalities, however, is that the application design becomes more complex. Thus, techniques for choosing specific sensing modalities for a particular application task as well as algorithms that exploit the presence of multiple modalities are two key challenges that need to be addressed.

4.3 Protocols for Multi-tier Interaction and Resource Management

The various tiers in the network will need to interact and coordinate with one another to achieve application goals. Further, resources at these tiers will need to be allocated dynamically in order to meet application needs at run-time. Consequently, the design of a M^2 sensor application requires a suite of protocols to enable interactions and coordination as well as dynamic resource management techniques.

Protocol Suite: Much of the research on protocols for sensor networks has focused on low-level issues such as multi-hop routing, unicast, and local broadcast. In addition to these low-level protocols, M^2 application design will require support for high-level interactions between tiers. For instance, data fusion is commonly used in sensor networks to to increase sensing fidelity by exploiting observations from multiple overlapping sensors or multiple sensing modalities [17]. Any data fusion algorithm requires support for gathering data from multiple sensors, local processing, and propagation of results to other sensors. By designing a protocol that supports data gathering, local refinement, and propagation of results, one can simplify the implementation of any data fusion algorithm. Observe that such a protocol enables higher-level interactions using low-level routing, unicast and multicast protocols for sensor networks. Other high-level interactions that are common in sensor networks include triggering upon event detection, and handoff between nodes. In order to simplify application design, protocols that support canonical high-level interactions between nodes within and across tiers will need to be developed.

Dynamic Resource Management: Nodes in an M^2 network are heterogeneous with respect to their processing, storage, sensing and radio capabilities. Further, the sensing and processing workload seen by a sensor network can exhibit significant temporal and spatial variability, where a quiet period is followed by a burst of localized events. The heterogeneity in sensor nodes and the dynamics of the workload motivate the need for dynamic resource management in M^2 sensor networks. While dynamic resource management is not a new problem in the context of distributed systems, the goal of resource management in sensor networks is to optimize for power, unlike traditional distributed systems that are optimized for availability or performance. Issues such as (i) balancing the processing load by actively distributing it among nodes within and across tiers, (ii) handling node failures by redistributing sensing tasks to other overlapping nodes, and (iii) multi-tier power management need to be addressed in the context of M^2 networks. The primary challenge is to design resource management algorithms that are fully decentralized and yet sufficiently simple to run on nodes as constrained as Motes or Stargates.

4.4 Programming Abstractions

Although M^2 networks have a number of advantages, an important drawback is that they make application design more complex. In a single-tier network, the same code runs on all sensor nodes, and all nodes have identical roles. In contrast, the heterogeneous nature of M^2 network implies that the application tasks are partitioned across tiers and different components of the application execute on different tiers, thereby complicating application design. The situation is exacerbated by the current generation of programming tools which are designed primarily for homogeneous single-tier networks and have limited support for multi-platform application development. Further, while the scale of a sensor network introduces numerous challenges even in the single tier instance, the complexity grows significantly for multi-tier networks.

In an M^2 network, different tiers comprise different platforms, each of which has different hardware characteristics and runs different operating systems. Thus, the application designer will require expertise for programming multiple embedded hardware platforms and may also need to program multiple implementation of the same algorithm, one for each platform. For instance, multiple implementations of motion detection may be required in a surveillance application—one for highly resource–constrained Motes that run TinyOS, and another for less-constrained Stargates that run a embedded Linux distribution.

Consequently, a major challenge for M^2 networks is to provide high-level programming tools and libraries that can significantly ease the complexity of developing applications. For instance, the application designer might program an algorithm once, and have programming tools to tailor it to different platforms. Similarly, a library of modules that implement common services, protocols, and algorithms for a multitude of embedded platforms can also simplify application design.

5. SENSEYE: A M^2 CAMERA SENSOR APPLICATION

To better understand the research challenges articulated in the previous section, we are designing and implementing *SensEye*, a camera–based M^2 sensor network application for intruder detection and surveillance. The objective of our effort is to provide a flexible prototyping platform to implement and evaluate various mechanisms, protocols, and algorithms for constructing M^2 networks.

The initial prototype of SensEye consists of four tiers: (i) a first tier comprising vibration sensors connected to Motes, (ii) a second tier comprising a dense network of low-power low-fidelity cam-

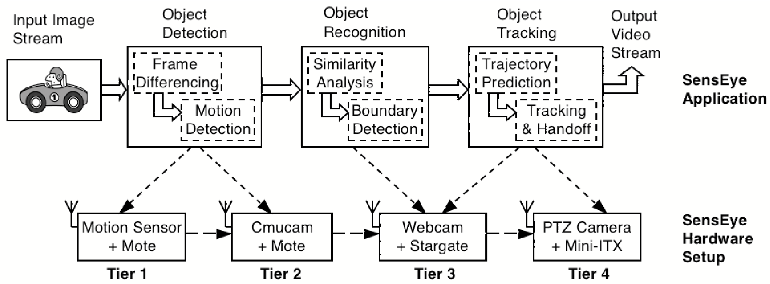

Figure 3: SensEye application components and their mapping to a four tier network.

eras (e.g., Cyclops or CMU-cams) connected to Motes, (iii) a third tier of high fidelity web-cams connected to Intel Stargates, and (iv) a fourth tier comprising a sparse network of pan-tilt-zoom cameras. The first three tiers are assumed to be untethered and battery-powered, while the fourth tier is assumed to be tethered. All nodes are assumed to be interconnected using a short-range multi-hop wireless network (e.g., 802.15.4). Observe that this setup provides two sensing modalities, namely vibration and video, to achieve application goals.

We are implementing a surveillance application using this four tier network. The goal of our surveillance application is three-fold: (i) *object detection*, which detects a new, moving object in the environment with a low latency and high probability, (ii) *object recognition*, which matches a new object to a pre-configured list of known objects in order to determine its type, and (iii) *object tracking*, which involves continual tracking of the object as it moves through the environment and streaming of the object images to a monitoring station.

Figure 3 depicts the various tasks involved in our surveillance application. Given the multiple tiers and their varying capabilities, our prototype maps these tasks to the four tiers as shown in Figure 3. The two lowest tiers are assigned the responsibility of object detection. Thus, two sensing modalities are employed for low-latency detection of new objects. Vibration sensors detect motion by sensing vibrations and can detect an approaching object even before it is in visual range. Low-fidelity cameras (Cyclops or CMU-cams) detect new objects by performing motion detection. For instance, simple frame differencing can be employed to detect motion or motion vectors can be computed for this purpose. Further, CMU-cams support motion detection in hardware, enabling processing cycles on the Motes to be utilized for other purposes.

Once a new object has been detected, lower-tier nodes wakeup one or more third-tier nodes in their vicinity. Nodes in this tier acquire a high-resolution image of the object and use the more capable processor on Stargates to perform object recognition. We assume that the nodes are preconfigured with a database of images, and they perform simple object matching to determine the best match from this list. The object is then tagged to be of the corresponding type (e.g., a car versus a truck). If multiple web-cams overlap in the coverage, then object recognition can be performed at each node and a consensus protocol can be used by these nodes to agree on the object type. Our object recognition algorithm involves boundary detection, which isolates the object from the rest of the image, and object matching, which uses various statistics, such as shape and color, to compare it to the known set of objects.

Once the object has been detected and recognized, the final task is that of tracking. A combination of web-cams and pan-tilt-zoom cameras can be employed to track the object as it moves through the environment. Motion vectors can be computed from the video captured by the web-cams to determine the trajectory of the object in the environment. As the object moves out of range of one camera into the range of another, handoff protocols are used to transfer responsibility of object tracking from one camera to another. Finally, a sequence of images can also be streamed to an external monitoring station as the object is tracked. Such streaming involves ad-hoc streaming techniques as well as handoffs from one sensor node to another.

Our prototyping efforts draw upon a number of open–source projects to implement various functionality: (i) Open Source Computer Vision Library (OpenCV) [10], which implements several commonly used vision algorithms, (ii) Movement Video Capture (MVC) [9], a tool for motion detection, (iii) FFmpeg [2], a tool for encoding and streaming audio and video, (iv) GStreamer [5], a framework for creating streaming media applications. We also build upon other research efforts that have targeted single-tier camera sensor networks such as Panoptes [1] and CVSN [3]. We assume a TinyOS environment on the Motes and Emstar with Familiar Linux on the Stargates. Existing services from TinyOS and Emstar for tasks such as radio communication, routing, localization and time synchronization are exploited by our implementation.

We are using our prototype to study issues related to initial bootstrapping and calibration of cameras, dynamic resource management, lifetime, latency and reliability tradeoffs, protocol design, and M^2 application programming. Although studied in the context of camera sensor networks, we expect that our research to shed light on the broader implications of these issues on designing other M^2 sensor networks and applications.

6. REFERENCES

[1] W. Feng, B. Code, E. Kaiser, W. Feng, and M. L. Baillif. Panoptes: Scalable Low-Power Video Sensor Networking Technologies. In *ACM Transactions on Multimedia Computing, Communications and Applications*, January 2005.

[2] FFmpeg Multimedia System. http://ffmpeg.sourceforge.net/

[3] A. Gamal and L. J. Guibas. Collaborative Visual Sensor Networks. `http://mediax.stanford.edu/projects/cvsn.html`

[4] D. Ganesan, R. Cristescu, and B. Beferull-Lozano. Power Efficient Node Placement and Transmission Structure for Data Gathering under Distortion Constraints. In *IPSN*, 2004.

[5] GStreamer: Open Source MultiMedia Framework. `http://gstreamer.freedesktop.org/`

[6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Proceedings of ASPLOS-IX*, November 2000.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed

Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of ACM Mobicom*, Boston, MA, USA, August 2000.

[8] D. Lymberopoulos and A. Savvides. XYZ: A Motion-Enabled, Power Aware Sensor Node Platform for Distributed Sensor Network Applications. In *IPSN SPOTS*, April 2005.

[9] Movement Video Capture. http://www.turbolinux.com.cn/ merlin/mvc/

[10] Open Source Computer Vision Library. http://www.intel.com/research/mrl/research/opencv/

[11] J. Polastre, R. Szewczyk, C. Sharp, and D. Culler. The Mote Revolution: Low Power Wireless Sensor Network Devices. In *Proceedings of Hot Chips 16: A Symposium on High Performance Chips*, August 2004.

[12] M. Rahimi, D. Estrin, R. Baer, H. Uyeno, and J. Warrior. Cyclops, Image Sensing and Interpretation in Wireless Networks. In *Proceedings of ACM Sensys 2004*, 2004.

[13] A. Rowe, C. Rosenberg, and I. Nourbakhsh. A Low Cost Embedded Color Vision System. In *Proceedings of the IROS Conference*, 2002.

[14] Spec Mote. `http://www.jlhlabs.com/jhill_cs/spec/`

[15] Stargate Platform. `http://platformx.sourceforge.net/`

[16] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. In *Proceedings of ACM SenSys*, November 2003.

[17] Y. Wu, E. Y. Chang, K. C.-C. Chang, and John R. Smith. Optimal Multimodal Fusion for Multimedia Data Analysis. In *ACM International Conference on Multimedia*, October 2004.

[18] T. Yang, T. He, and J. Stankovic. Differentiated Surveillance for Sensor Networks. In *Proceedings of ACM SenSys*, November 2003.