SEVA: Sensor-Enhanced Video Annotation

XIAOTAO LIU, MARK CORNER, and PRASHANT SHENOY
University of Massachusetts, Amherst, USA

In this article, we study how a sensor-rich world can be exploited by digital recording devices such as cameras and camcorders to improve a user's ability to search through a large repository of image and video files. We design and implement a digital recording system that records identities and locations of objects (as advertised by their sensors) along with visual images (as recorded by a camera). The process, which we refer to as *Sensor-Enhanced Video Annotation (SEVA)*, combines a series of correlation, interpolation, and extrapolation techniques. It produces a tagged stream that later can be used to efficiently search for videos or frames containing particular objects or people. We present detailed experiments with a prototype of our system using both stationary and mobile objects as well as GPS and ultrasound. Our experiments show that: (i) SEVA has zero error rates for static objects, except very close to the boundary of the viewable area; (ii) for moving objects or a moving camera, SEVA only misses objects leaving or entering the viewable area by 1–2 frames; (iii) SEVA can scale to 10 fast-moving objects using current sensor technology; and (iv) SEVA runs online using relatively inexpensive hardware.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**] Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**] Information Search and Retrieval; H.5.1 [**Information Interfaces and Presentation (e.g., HCI)**]: Multimedia Information Systems—*Video*

General Terms: Algorithms, Design, Experimentation

Additional Key Words and Phrases: Video annotation, sensor-enhanced, location-based services, context-based retrieval

ACM Reference Format:

Liu, X., Corner, M., and Shenoy, P. 2009. *SEVA:* Sensor-enhanced video annotation. ACM Trans. Multimedia Comput. Commun. Appl. 5, 3, Article 24 (August 2009), 26 pages. DOI = 10.1145/1556134.1556141 http://doi.acm.org/10.1145/1556134.1556141

1. INTRODUCTION

Advances in consumer electronics technologies have led to a proliferation of cameras and camcorders that record images and video in digital form and enable easy manipulation of this data on laptops and desktop computers. This trend, coupled with the increasing capacities of PC hard drives, has encouraged users to create ever-larger personal libraries of pictures and movies. Navigating through collections containing tens of thousands of pictures and hundreds of movies requires tools to quickly search and locate content of interest.

Searching and retrieving media is greatly enhanced by textual annotations; these annotations contain the metadata of media's context, such as *when*, *where*, *who*, and *what* [Davis et al. 2004; Dourish 2004].

A preliminary version of this article appeared in *Proceedings of the 13th ACM Annual Conference on Multimedia (MM'05)*, where it won the best paper award. This work was supported in part by NSF grants CCR-0219520, EIA-0080119, CNS-0520729, CNS-0519881, CNS-0447877 and DUE-0416863.

Authors' addresses: X. Liu, M. Corner, P. Shenoy (corresponding author), Department of Computer Science, University of Massachusetts, Amherst, MA 01003; email: shenoy@cs.umass.edu.

Users enter annotations manually [Gemmell et al. 2002] or they are automatically generated by a combination of learning- and vision-based object/face recognition techniques [Fan et al. 2004; Feng et al. 2004; Jin et al. 2004; Li and Goh 2003; Nack and Putz 2004; Zhang et al. 2004]. However, manual annotation is cumbersome and faces the difficulty of imprecise human memory, while vision-based automatic annotation is error prone and has high computational requirements.

Recently, numerous sensor technologies such as RFID [Finkenzeller 2003] and low-power sensors [Hill and Culler 2002; Lymberopoulos and Savvides 2005; Mainwaring et al. 2002; Polastre et al. 2005] have emerged. A concurrent trend is the ubiquitous deployment of positioning technologies such as GPS [Bajaj et al. 2002] and ultrasound [Priyantha et al. 2000] that triangulate the exact location of a user. Taking advantage of these techniques, people can automatically record sensory data such as GPS readings, light readings, temperature readings along with images, videos, and audio [Aizawa et al. 2004; Davis et al. 2004; Ellis and Lee 2004; Gemmell et al. 2004; Naaman et al. 2003; Su et al. 2004; Toyama et al. 2003]. Such sensory data provide the metadata of media's context which can be used to improve the efficiency of media retrieval. However, these systems only record two types of contextual metadata, namely *when* and *where*, and miss the most important ones: *who* and *what*.

This article proposes a new multimedia paradigm which enhances digital recording devices with sensor technology to automatically record the most important contextual metadata—*when*, *where*, *who*, and *what*—along with visual images and videos. The process, which we refer to as *Sensor-Enhanced Video Annotation (SEVA)*, produces a tagged stream that later can be used to efficiently search for videos or frames containing particular objects or people. This greatly expanded knowledge of the contents of videos and images will enable a number of new applications and use cases.

—*Managing personal collections*. Content in personal photo and video collections is continuing to explode with the advent of digital cameras and camcorders that make recording pictures and video virtually free. However, managing and searching through libraries containing thousands of photos and hundreds of videos has become a major challenge. While manual organization and annotation of such content is tedious, automated tagging can vastly improve search capabilities of such libraries and simplify the task of managing one's personal multimedia content.

—*Social and collaborative sharing*. Sites such as Flickr and YouTube have become extremely popular by enabling users to share their pictures and videos with other Internet users. The success of these sites depends critically on user-supplied tags and annotation describing the content of each uploaded picture or video. Doing so enables not only search through these massive shared collections but also enables related content to be grouped together for a better browsing experiences. Automatic extraction of tags and metadata allows users to automatically find pictures from certain locations, containing certain people, or certain kinds of objects. Feeding this metadata into collaborative systems, users can create spatio-temporal summaries of events, objects, and people, using automated retrieval [Appan and Sundaram 2004; Adams et al. 2006]. Flickr has already taken a step in this direction by exploiting GPS-stamped pictures to extract location information and grouping pictures by location.

—*Assistive applications*. Researchers have argued that recording entire lifetimes of individuals in a "life log" is feasible with the advent of digital video recorders and inexpensive storage. Such lifelong records of personal events result in voluminous information that is useful only with powerful search capabilities. One possible application is to help the elderly and people with memory-related disabilities to recall past events using such digital records. For example, users can then perform queries to locate the last time they handled a misplaced objects ("when did I last see my keys?"). However, automated tagging of video content is key to providing accurate search functionality through such content.

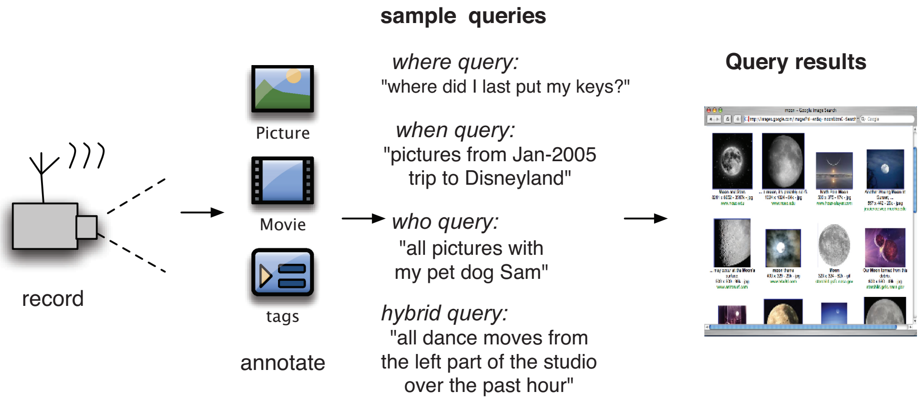

sample queries
where query:
"where did I last put my keys?"
when query:
"pictures from Jan-2005 trip to Disneyland"
who query:
"all pictures with my pet dog Sam"
hybrid query:
"all dance moves from the left part of the studio over the past hour"
Query results
record
Picture
Movie
tags
annotate

Fig. 1. Examples of use-cases and queries enabled by applications built on SEVA.

—*Sports analysis*. Referees in sports such as cricket have begun employing instant video replays to review a particular event to make decisions. Since fast sports moves are difficult to analyze even with video replays from multiple cameras, researchers have begun employing vision-based video analysis to help referees track the trajectory of a ball or the precise position of a player in the field. Use of sensor tags on players and balls can help augment such vision-based techniques by precisely identifying the positions of these objects in a blurry image or a fast-moving video clip.

—*Studio-based video production*. In video production the simple addition of metadata has the potential to greatly simplify finding particular clips containing objects or people. In a situation such as a studio space, objects of interest can be tagged with sensors, and each camera can collect its own metadata information for the video. Later, by automatically retrieving video containing particular elements, an editing system could present a few video clips for inclusion in a composition. This has the potential to make video production a much more efficient process.

—*Extracting higher-layer semantics*. Using expanded metadata information, systems that go beyond simple retrieval become possible, by inferring higher-level meaning from video in postproduction, or even during the capture process. One such system, the Mindful Camera, uses manual annotations to aid videographers, and can use very high-level semantic information to construct story lines [Barry 2005].

All of the previous scenarios are enabled by automated tagging of images and video content by tags describing "when", "where", "what", and "who" (see Figure 1). Our goal is to design a low-level substrate that would provide an automated way to generate these tags using sensor information related to each image; higher-level applications such as those described by the preceding use-cases can then be built on top of such a substrate. To enable this vision, this article assumes a future sensor-rich world in which many objects and people will be equipped with one or more small sensors or passive tags. These sensors report their identities as well as their locations when queried. For stationary objects such as a building or a street sign, the precise location is hardcoded at sensor configuration time. People may carry smart IDs (e.g., next-generation cell phones) that reveal their identities and locations to trusted entities. To handle mobile objects as well as those that do not hard-code their locations, we assume the presence of a positioning system.

Given such an environment, we design and implement a digital recording system that records identities and locations of objects (as advertised by their attached sensors) along with visual images (as recorded by a camera). The recording device includes four key elements: (i) a video camera, (ii) a digital compass, (iii) a position system, and (iv) a wireless radio. The camera is simply a digital recording device that captures video frames and the associated audio. The digital compass is used to determine the direction where the camera is pointing at any instant; we use a 3D digital compass that precisely provides the orientation (heading, pitch, and roll) of the camera. The camera is equipped with a location system so that it can determine its position in relation to objects and people. Finally, the wireless radio is used to query objects for their identities and locations.

1.1 Research Challenges

Numerous practical challenges arise in the design and implementation of SEVA.

—*Mismatch in coverage and range*. The SEVA recorder includes a video camera and a wireless radio to record images and sensor data, respectively. Typically, the camera is a directional image sensor that captures a limited view of the scene depending on where the lens is pointing. In contrast, the wireless radio antenna is an omnidirectional device and is able to listen to sensors that are outside the viewable area of the camera. This can result in false positives, since the radio may record objects that do not actually appear in the captured image. Even with a directional antenna, it is difficult to precisely match the coverage of the radio and the lens; focus and zoom capabilities of the lens further complicate the issue. Similarly, the lens can capture images of objects that are infinitely far from the camera (e.g., a distant building), while the wireless radio has a limited range and is unable to record identities of object that are outside its range. This results in false negatives where objects that are in the view of the camera are unable to report their identities to the wireless radio.

—*Mobility*. Mobile objects and moving cameras cause objects to move in and out of the field of view. SEVA must correctly identify which frames contain a particular object with a high degree of accuracy.

—*Limitations of power-constrained, bandwidth-poor sensors*. Sensors attached to objects are either battery-powered or passive. Due to power constraints, battery-powered sensors aggressively duty-cycle themselves and use sleep modes to enhance their lifetimes. Passive sensors such as RFID tags do not have a power source and instead are powered by the electromagnetic signals from the wireless radio. Further, both battery-powered and passive sensors use low-bandwidth wireless channels for communication. While a video camera can record at a rate of 30 frames/second, due to the resource constraints on sensors it is not feasible for the wireless radio to query all objects every 33ms. Thus, sensors will respond less frequently than the intraframe duration, necessitating extrapolation techniques to annotate every frame.

—*Limitations of positioning systems*. Positioning systems often update at a much slower rate compared to the video frame rate. For instance, current GPS receivers provide position readings at a rate of 1–10 readings per second, and current ultrasound systems only provide an update rate of a few readings per second. As a result, SEVA requires extrapolation techniques that can annotate every frame with location information. Further, SEVA requires a high degree of positioning accuracy in order to properly identify viewable objects. Unfortunately, the current generation of positioning systems provide limited accuracy. For instance, current GPS technology provides accuracy of 3–100 meters [Bajaj et al. 2002], while handling moving objects in ultrasound has inherent problems [Smith et al. 2004]. SEVA must deal with the error that is introduced as a result of these limitations.

1.2 Research Contributions

The primary contribution of our work is to demonstrate the feasibility and benefits of using sensors and locationing systems to automatically annotate video frames with the identities of objects. Our work has resulted in a number of novel techniques that are specifically designed to address the aforesaid practical hurdles.

The mismatch in range and coverage of sensors is handled using a combination of extrapolation and filtering. In particular, false positives are eliminated using elementary optics and filtering techniques, while false negatives caused by a visible object that moves out of radio range are handled using path extrapolation. To address the issue of mobile objects as well as a moving camera, we draw upon regression techniques and Kalman filters to determine the path of a mobile object and its location. To address the issues of resource-constrained sensors and limited update rate of positioning systems, we employ interpolation techniques to determine if an object is within range even if it did not respond to a query or if the positioning system did not provide position reading when the frame was captured. Finally, buffering and filtering are used to handle some, but not all, of the inaccuracies of positioning systems.

These techniques have led to a fully working prototype of SEVA. We conduct detailed experiments using both stationary and mobile objects as well as GPS and ultrasound. Our experiments show that: (i) SEVA has zero error rates for static objects, except very close to the boundary of the viewable area; (ii) for moving objects or a moving camera SEVA only misses objects leaving or entering the viewable area by 1–2 frames; (iii) when both are moving SEVA only misses objects leaving or entering the viewable area by 3 frames; (iv) the SEVA prototype can scale to 10 fast-moving objects using current sensor technology; and (v) SEVA runs online using relatively inexpensive hardware.

In many ways SEVA is predicated on a technological world very different from the current one, in which precise locationing and object identification are pervasive. This article assumes such a future where these technologies have evolved beyond their current state. However, given rapid advances in embedded computing and RFID tags, we believe that technology will deliver the necessary hardware and deployment for applications like SEVA in the near future.

The rest of this work is structured as follows. Section 2 presents some background. In Section 3 we present the design of SEVA. We present implementation details in Section 4 and our experimental results in Section 5. Sections 6 and 7 present related work and our conclusions.

2. SYSTEM MODEL

In this section, we present the key assumptions made in our work. SEVA assumes a world rich in sensors; we believe that, in the future, sensors will be pervasive, and most objects will be equipped with one or more sensors. Not all objects fall into this category; natural objects such as trees and mountains may not be sensor-enhanced and annotation requires techniques that are beyond the scope of this article. In general, sensors on objects will be heterogeneous and will be based of a mix of technologies such as RFID, Bluetooth, Zigbee, and 802.11. Consequently, the recording device will need a radio to interact with each type of sensor. For reasons of simplicity, our current work assumes a homogeneous sensor environment and assumes a recorder with a single wireless radio; it is straightforward to extend our prototype to handle heterogeneity.

We assume that all sensors report their identities as well as their locations when queried. For stationary objects such as a building or a street sign, the precise location can be hard-coded at sensor configuration time. To handle mobile objects as well as those that do not hard-code their locations, we assume the presence of a positioning system. In this work, we consider two types of positioning systems: GPS and an ultrasound system named Cricket [Smith et al. 2004]. GPS is an outdoor positioning system that relies on satellites, and Cricket is an indoor system based on ultrasound beacons. For passive sensors such as RFID we assume that they store their current coordinates and are reprogrammed using emerging RFID triangulation techniques [Hightower et al. 2000; Ni et al. 2003].

We also assume that the recording device incorporates four key elements: (i) a video camera, (ii) a digital compass, (iii) a locationing system, and (iv) a wireless radio. The camera is simply a digital recording device that captures video frames and the associated audio. We assume that the parameters of the lens used in the camera are precisely known. This is a reasonable assumption since these parameters are published or advertised for most models of digital cameras and camcorders. The digital compass is used to determine the direction where the camera is pointing at any instant; we use a 3D digital compass that precisely provides both the orientation and the tilt of the camera. The camera is also assumed to equipped with GPS and Cricket so that it can determine its coordinates both indoors and outdoors. Together, the positioning device and the 3D compass, in conjunction with the lens parameters, are used to determine which part of the scene can be seen by the camera. This automatic computation of the visual range of the camera is used to determine which objects are in view and which ones are false positives. Finally, the wireless radio is used to query objects for their identities and locations.

In addition to recording video, the SEVA recorder is assumed to log: (i) the orientation and rotation of the camera when the compass provides a new reading, (ii) the GPS and/or Cricket coordinates of the camera when the positioning systems provide a new reading, (iii) a timestamp for each frame, and (iv) the identities and the locations of each queried object and the time when the response was received.

Assuming such an environment, we present the architecture, design, and implementation of our *Sensor-Enhanced Video Annotation (SEVA)* application in the following sections.

3. SYSTEM ARCHITECTURE AND DESIGN

SEVA captures two streams (one sensor data and one video) and fuses them together in a series of stages. Each step requires careful filtering and melding of object location, object identification, camera positioning, and lens parameters. SEVA is capable of feeding this annotated stream of video into a database for offline querying or to a streaming query system. This process is broken into six key stages: video recording, pervasive location/identification, correlation, extrapolation and prediction, filtering and elimination, and finally database querying. Next, we describe these stages in detail.

3.1 Video Recording

SEVA provides a video recording module that receives video input and camera parameters from any video source. The source must provide frames at a constant and known frame rate, or it must timestamp each frame. This allows later stages to synchronize location information with individual frames. The camera must also supply a set of lens parameters to the recording module: the sensor size and the lens focal length. For lenses with fixed focal lengths (so-called prime lenses), the focal length will not change from frame to frame. However, SEVA is also capable of handling zoom lenses with variable focal lengths.

3.2 Pervasive Locationing/Identification

SEVA collects information about the location and identity of proximate objects. This depends on a pervasive infrastructure that responds to broadcast messages from SEVA through a wireless network. Any objects within wireless range respond with information about their identity, including relevant properties of the object. The exact response give by objects could take many forms, including a simple identifier, or more complete information following some standard schema. This issues of naming and identification are orthogonal to the SEVA system, which can use a variety of naming systems. AutoID's Physical Markup Language may provide a future standard to base such systems on [Mealling 2003]. Such identity and pervasive computing infrastructures have been proposed for a broad array of systems [Grimm 2002; Johanson et al. 2002; Kindberg et al. 2002; Roman et al. 2002] and future systems may use a variety of technologies and standards. SEVA is designed to be independent from the exact technological implementation, so here we only describe an abstract set of properties that SEVA depends on.

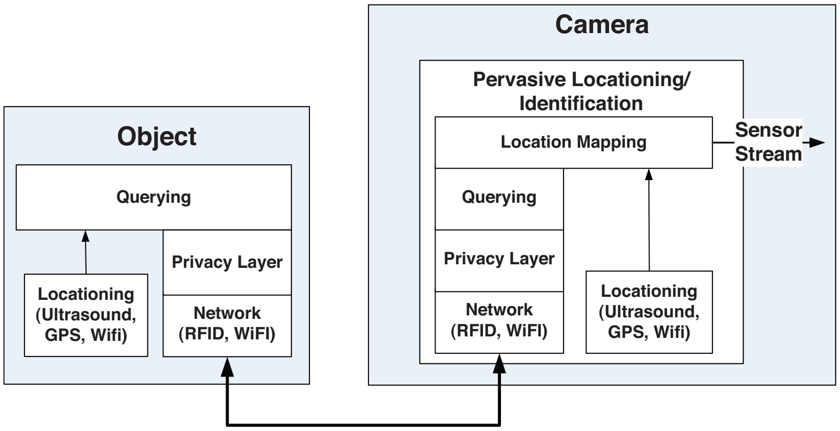


Fig. 2. Pervasive locationing/identification system.

The pervasive locationing and identification shown in Figure 2 produces the sensor stream used by later stages of SEVA. The system is organized as a set of modular layers: locationing, network, privacy, querying, and location mapping.

The locationing layer provides location information to the objects as well as the camera. The locationing system can be active, passive, or static. Active systems, such as active ultrasound, beacon to the infrastructure, which responds with a location. Passive systems, such as GPS, can compute locations with no transmission and only passive observations of radio signals. Static systems use a programmed location. Active and passive systems are best for objects that move, such as people and automobiles, whereas static systems are only appropriate for immobile objects such as buildings and landmarks. As we show in the evaluation section, the accuracy of these systems is critical to SEVA's efficacy.

The network layer provides communication between the camera and objects. As long as the interface supports broadcasting, sending, and receiving, the particular technology used (WiFi, Bluetooth, Zigbee, RFID) is immaterial. The range of the communication should be sufficient to capture most objects within camera range; however, too great of a range will affect the scalability of the system. The limited range does mean that large, distant objects such as mountains will not be captured by the identification system; future SEVA mechanisms might support such operations through Geographic Information Systems, or other databases of larger, static objects.

A privacy layer ensures that objects can control their own visibility. While a complete implementation of such a system is beyond the scope of this article, the privacy layer should permit people to provide varying levels of information. For instance, a person will provide her name to her friend's camera, whereas she will only provide metainformation such as "a person" to an untrusted camera. Recently, researchers have begun to tackle the complexity of providing privacy in large multimedia collections [Ahern et al. 2007]; however the focus has been on providing privacy at the granularity of the photo, rather than the objects that are contained in the photos themselves.

One alternative to making privacy decisions at the time of capture is for all objects to provide an encrypted version of their metadata information, and privacy decisions can be made at a later time using a structured access control policy. For instance, after obtaining information about the objects in frames, the person taking the photos could send queries to the owners of objects. One such system, Confab [Hong and Landay 2004], provides support for building a privacy-aware application, which could

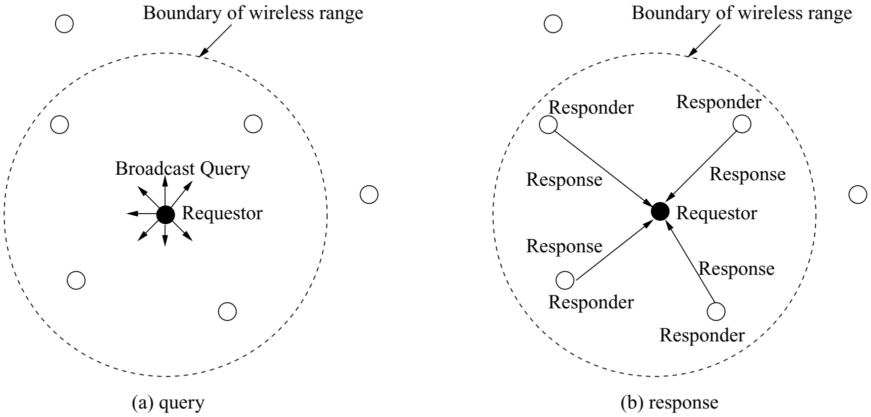


Fig. 3. Query and response model.

be used to build privacy into SEVA at the granularity of objects. Given the complexity of building such a system, we leave this as future work.

The querying layer manages interactions between the camera and the objects. The camera broadcasts query messages to objects, which respond with identifying and location information, as shown in Figure 3.

The location mapping layer corresponds different object locations to a frame of reference relative to the camera. Therefore, SEVA can still compute visibility even when different objects use different locationing systems, and enable interoperability across locationing systems.

3.3 Stream Correlation

The sensor stream needs to be time synchronized with the video stream in order to correlate the location information in the former with specific frames in the latter. Unfortunately, transmission, contention, and processing delays cause location information to be desynchronized with the video.

Depending on whether sensors are active or passive, correlation can be done in two ways. A straightforward implementation assumes a synchronized clock present at each object; SEVA uses GPS receivers, cellular phone references, or NTP-based time sources. If the sensor does not have a clock (e.g., RFID) or lacks resources to run a synchronization protocol, then instead of a timestamp, it provides an estimate of the time from query to response. This includes MAC-layer delays (only meaningful for response from nearby objects, not available for the camera location) and internal processing. The recorder subtracts this delay from the receipt time of the response and assigns the corrected timestamp to the sensory information (propagation delays are assumed to be negligible). By performing this correlation, SEVA associates each query response and each camera location to the appropriate frame. The choice of which method to use is dependent on the type of tag and time synchronization features should be advertised as part of the tag response.

3.4 Extrapolation and Prediction

Some per-object, per-frame location information will be missing from the correlated sensor stream. This is due to three factors. First, sensors duty-cycle themselves to maximize their battery lifetime and will respond to queries only when awake. Broadcast requests will be sent out every frame duration (e.g., every 33ms for 30 frames/s video) while sensors may sleep for tens or hundreds of milliseconds between two wakeups. Second, it is unlikely that the network layer can scale its MAC protocol to the number of awake objects (due to the possibility of MAC-layer collisions). In that case, the individual objects must randomly ignore broadcast requests. Finally, the update rate of the positioning systems is slower (e.g., at most 10 positions/s) than the typical video rate (e.g., 25–30 frames/s). Therefore, only a subset of frames contains the camera location.

SEVA explicitly deals with all of these scenarios by assuming: (i) each query will obtain responses from only a *subset* of the objects within radio range; and (ii) only a *subset* of the frames contain the camera location. SEVA employs postprocessing techniques to account for missing responses and missing camera locations. Since we use the same postprocessing techniques for both objects and the camera, in the following we only present our techniques for objects. Depending on whether the objects are stationary or mobile, such interpolation is done as follows.

Rather than considering locations that are relative to the camera, SEVA considers the absolute locations (relative to the world coordinate) of both the camera and the objects. This can simplify the interpolation procedure because the locations logged in different frames use the same reference coordinate frame.

Static objects. If the objects are static, extracting missing information is straightforward: We simply copy the reported location of the object to intermediate frames. In particular, if the object responds to queries at time t_1 and t_2 and reports the same location for both queries, this location is tagged for all frames captured between times $[t_1, t_2]$.

Mobile object. Next we consider a mobile object; determining missing location information in this case requires a motion model. SEVA is capable of using two different tracking algorithms: regression techniques [Devore 1999] and a Kalman filter [Simon 2006] to determine the object at any time instant. In the following, we present details of the Kalman filter technique, and refer interested readers to the preliminary version of this article [Liu et al. 2005] for details of the regression technique.

Kalman filter. SEVA exploits an Extended Kalman Filter (EKF) to track the object's movement, a commonly used technique in object tracking. Our use of the EKF incorporates a state vector with 6 components, 3 position components (x, y, z), and 3 velocity components (v_x, v_y, v_z), and is inspired by the work of Smith et al. [2004]. In particular, after getting a position sample, the EKF knows its state and the corresponding confidence which is represented by a covariance matrix. By assuming an object moves at a constant velocity between position samples, the EKF can project ahead its internal state for any time instant before the arrival of the next position sample. In this prediction step, the predicted velocity components are the same as those of the internal state at last position sample, and the predicted position components are given by the internal state at the last position sample plus the movement during the time difference. When the next position sample arrives, the EKF then corrects its prediction according to the difference between the predicted position and the actual reported position. Finally, the EKF uses the corrected state vector as the location and velocity estimate at the time instant of the new sample. In summary, the EKF estimates the object's state (position and velocity) by using a form of feedback control: The filter estimates the object's state at some time and then obtains feedback in the form of noisy position samples. An overview of the Kalman filter formulation is given in APPENDIX A.

Unlike the regression technique we used in the preliminary version of this article [Liu et al. 2005], when doing interpolation, the Kalman filter technique takes the variance of the locationing system into account. Our older regression techniques did not consider such variance and *essentially assumed that the locationing system was accurate.* Since all locationing systems have some error, by incorporating the variance of position measurements, our Kalman filter can make better predictions than the regression method, especially when the variance in measurements is large or not constant.

In reality, we don't have position sample for every time instant, and consequently, the EKF cannot run a correction step for every time instant. To solve this, we use the predicted state vector as the

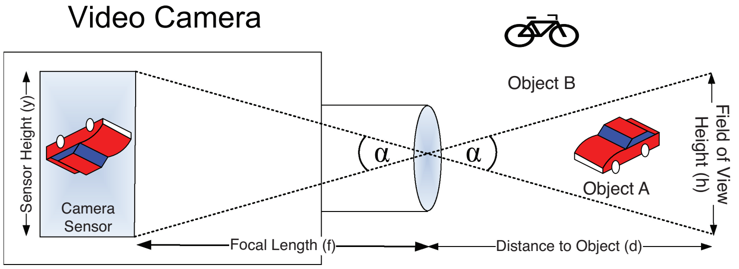


Fig. 4. The basic optics model.

object's state estimate for the time instants without position samples; we use the corrected state vector as the object's state estimate for the time instants with position samples.

Extrapolation. The Kalman filter enables us to interpolate the location of an object given its path for an interval $[t_1, t_n]$. However, this does not yield any location information for frames captured before time t_1 and those captured after time t_n. This is useful when an object goes out of the range of the wireless radio but remains in view of the camera (e.g., an object that is steadily backing away from the camera). Once the object leaves the wireless radio range its presence is no longer detected, yielding false negatives.

The state vectors computed by the Kalman filter for t_1 and t_n can be used to extrapolate this information and annotate a small number of frames before t_1 and after t_n. This extrapolation can be done only for a few frames (e.g., for a few seconds) in order to reduce errors caused by a change in trajectory after the object leaves the wireless range. As discussed previously in this section, our confidence (covariance matrix) in the state vector will degrade over time. Currently, our prototype uses a configurable parameter to control the number of frames for which location information is extrapolated beyond the $[t_1, t_n]$ interval.

3.5 Filtering and Eliminating

After the extrapolation and prediction stage, every video frame has been annotated with object location information and SEVA must now determine which objects are within the camera's field of view.

For each frame, SEVA constructs a field of view based on an optics model, the camera's focal length, and parameters of the camera's sensor. As shown in Figure 4, let f denote the focal length of the lens and let y denote the height of the CMOS sensor of digital camcorder. This implies that the camcorder has a viewable angle $\alpha = 2tan^{-1}\frac{y}{2f}$. At a distance d from the lens, the camera can see a view that is $h = \frac{f}{d} \cdot y$. So if the object is within $\frac{h}{2}$ of the camera's axis, it is considered in view, otherwise it is out of view. In Figure 4, the object A is in the view and object B is out of view. The figure shows a two-dimensional model, and it easily extends to three dimensions. Using this model, combined with the location information, SEVA determines which objects are in the view of the camera.

This model does not take obstructions into account and SEVA will believe that some objects that are hidden by walls are actually visible. One possible solution is to use the calculated distance with radio power control and a free-space communications model to estimate whether the object is obstructed. Similarly, the object may be out of focus and therefore not visible. Some cameras have variable apertures and optics that provide the depth-of-field of the image. This allows us to compute whether objects are in or out of focus and tag them appropriately. Further, it may be advantageous to record the presence of *all* objects, and denote which are visible; this would be useful in constructing higher-level understanding of the scene.

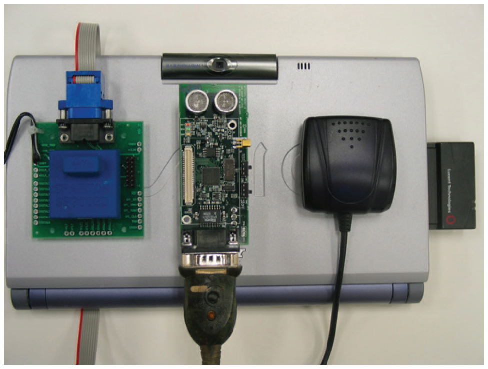


Fig. 5. SEVA recorder laptop equipped with a camera, a 3D digital compass, a Mote with wireless radio and Cricket receiver, a GPS receiver, and 802.11b wireless.

3.6 Query and Retrieval

This module consists of a storage system for annotated video and tools for query and retrieval. The storage system stores videos and corresponding annotations separately; the annotations and videos are synchronized and linked by the video's frame index; the location information in the annotations is translated into user-readable format (e.g., CS Building, Room 101). A tool allows users to query and retrieve videos of interest. Queries can specify *when* a video was captured, *where* it was captured, and *who* and *what* is in the video. The search engine then searches video annotations produced by SEVA and returns the videos' frame indexes satisfying the query. Finally, the returned frame indexes can be used to retrieve video clips from storage. Using the query interface, it will be possible to infer higher-layer semantics from images and video, either automatically or with assistance.

4. IMPLEMENTATION

To provide a test platform, we have constructed a prototype system based on a Sony Motion Eye Web camera connected to a Vaio laptop as shown in Figure 5. The location and identity querying, correlation, extrapolation and prediction, filtering and elimination, and database storage software runs on the laptop. SEVA currently uses two 3D locationing systems for the camera and objects: GPS and the Cricket Ultrasound locationing system. To obtain the orientation of the camera we augmented the laptop with a Sparton SP3003D Digital Compass that provides the orientation (heading, pitch, and roll) of the camera's lens.

Video Recording. The CMOS-based camera provides uncompressed 320 × 240 video at 12 frames per second. The camera has been set to a fixed focal length of 2.75mm, and uses a sensor size of 2.4mm by 1.8mm. The video recording module uses an MPEG encoder(ffmpeg0.4.8) to record video.

Pervasive Location/Identification. Outdoors, SEVA uses Deluo GPS receivers equipped with WAAS correction, connected to the laptop to locate the camera and the object. The GPS unit provides latitude, longitude, and altitude, and it provides an accuracy of 5–15 meters.

Indoors, SEVA employs an ultrasound locationing system called Cricket [Priyantha et al. 2000]. Using a network of ultrasound sensors built onto sensor boards, Cricket can provide 3D locations with an accuracy of a few centimeters. Cricket can be used in two modes: active and passive. In the current implementation, SEVA uses the active mode as it is more accurate. In the future SEVA will use the passive mode as it scales to a larger number of objects.

The pervasive locationing and identification system uses two different network layers to communicate with the objects. Outdoor objects are laptops equipped with WiFi and indoor objects are Mica2 [Hill and Culler 2002] low-power sensor boards equipped with 900 MHz short-range radios. The laptop communicates with the objects using a sensor board of the same type. A simple broadcast-based query protocol is implemented between the Linux-based recorder and the Mica2 nodes.

Correlation. As GPS provides a globally synchronized clock among GPS receivers, we use this clock to correlate the location information with specific frames. Since the Cricket system doesn't provide such a globally synchronized clock, SEVA simply correlates the location information with specific frames by subtracting the mean processing and MAC-layer delay from the receiving time of sensor data and assigning the corrected timestamp to the sensory information.

Extrapolation and Prediction. As discussed in Section 3.4, we use the Kalman filter technique to estimate the object's location. Because the camera's 3D orientation will affect the result of filtering and elimination, we also apply the Kalman filter technique to the camera's 3D orientation.

Filtering and Elimination. In this stage, objects' coordinates are transformed into a coordinate system with the camera as the origin. This transformation is straightforward for the Cricket system since we can easily subtract the camera's coordinate from the objects' coordinates. The transformation for a GPS system requires computing the distance between camera and object, and we use the GPS Drive library for this purpose [gpsdrive] (the library supports transformation, manipulation, and computations using GPS coordinates).

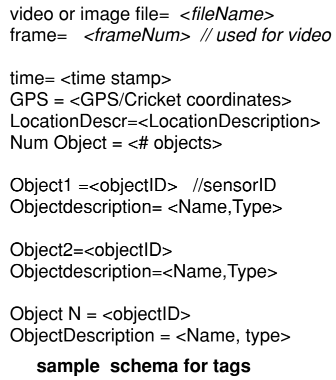
Indexing and Querying. The results of filtering and elimination are inserted into a MySQL database, while the videos are stored in the laptop's file system. Before SEVA adds annotations to the database, the outdoor GPS position (e.g., latitude, longitude, and altitude) is translated into user-readable format (e.g., parking lot 45, UMass) and the indoor Cricket location is translated into user-readable format (e.g., CS Building, Room 101). Although our current implementation uses a file that translates coordinates to human-readable location strings, this step can be automated via a service like Geocoder [geocoder] that provides the street address or location information for any GPS coordinate. Similarly, for each recorded sensor ID, we assume that the name of the corresponding object is known; such information can be stored in a personal database (e.g., tags of my friends) or can be queried from public databases.[1] For each video there is an entry in the database recording its start time and end time; for every frame in a video there is an entry in the database recording its shooting time and location; and for every annotation there is an entry in the database containing the object identity and index of the corresponding video frame. While Figure 6 shows the schema used to annotate and index videos, we note explicitly that SEVA is not constrained by this schema and is designed to work with any tag or metadata schema that describes a multimedia object such as an image or a video.

We have also implemented a simple GUI retrieval tool for content-aware queries on this database. This tool supports queries on *where* the video was captured (e.g., CS Building, Room 101), *when* it was captured (e.g., morning of May 23, 2005), and *who* or *what* is present in the video (e.g., car, book, building) and retrieves the indexes of all annotated frames that match this query. Finally, these frame indexes are used to retrieve frame sequences from videos.

5. EXPERIMENTAL EVALUATION

In evaluating SEVA, we set out to answer the following questions:

[1] There are public databases that store the object names for each bar-coded object; similar databases are assumed for RFID-tagged or sensor-tagged objects.



```
video or image file=  <fileName>
frame=   <frameNum>  // used for video

time= <time stamp>
GPS = <GPS/Cricket coordinates>
LocationDescr=<LocationDescription>
Num Object = <# objects>

Object1 =<objectID>   //sensorID
Objectdescription= <Name,Type>

Object2=<objectID>
Objectdescription=<Name,Type>

Object N = <objectID>
ObjectDescription = <Name, type>
    sample  schema for tags
```

Fig. 6. Schema of the tags/metadata used in SEVA.

—How accurate is SEVA in tagging frames with a moving camera, moving objects, and with different locationing systems?

—How well does SEVA scale to larger numbers of objects?

—What is the overhead in using SEVA?

To answer these questions we used three different locationing systems: the Cricket ultrasound system, GPS, and static locationing. We setup the Cricket locationing system in a 4m × 10m × 3m room with five Cricket receivers mounted on the ceiling that serve as the reference points for object and camera locationing. The origin of the coordinate system is one of the corners of the room and the range of x, y, and z is [0cm, 400cm], [0cm, 1000cm], and [0cm, 300cm], respectively. Our GPS experiments were conducted in a large parking lot with a clear view of the southern horizon. As the altitude did not vary significantly for object and camera positions, we did not use it in any of our experiments. The camera records all video at 12 frames/s.

To determine SEVA's accuracy in tagging frames, we subject the system to five experiments: (a) the object and camera are both static, (b) the object is moving in a straight line and the camera is static, (c) the camera is moving in different patterns and the objects are static, (d) the object is moving with semirandom trajectories and the camera is static, and (e) both the object and the camera are moving with semirandom trajectories. In these experiments, we place the object in different positions (some inside the view of camera and some outside the view of camera) and evaluate the error rate of our system when determining the viewability of objects. We selected the error rate or number of frames in error as the evaluation criteria. An error occurs when SEVA tags a frame as containing an object when it doesn't (false positives), or it tags a frame as not containing an object when it does (false negatives).

It is important to note that the objects that we are using to evaluate the system are only a few square centimeters in size. In a sense this represents a worst case. Larger objects, such as people, may have inaccuracies in the positioning information that is made up for by straddling the line between viewable and nonviewable. We leave the issue of partially viewable objects as future work.

5.1 Static Object, Static Camera

5.1.1 *Cricket Locationing System.* To evaluate SEVA's performance with static objects and a static camera, we place an object at a large number of positions along three different trajectories. The setup for this experiment is shown in Figure 7. The camera is set up at (223, 350, 57) with its lens pointing

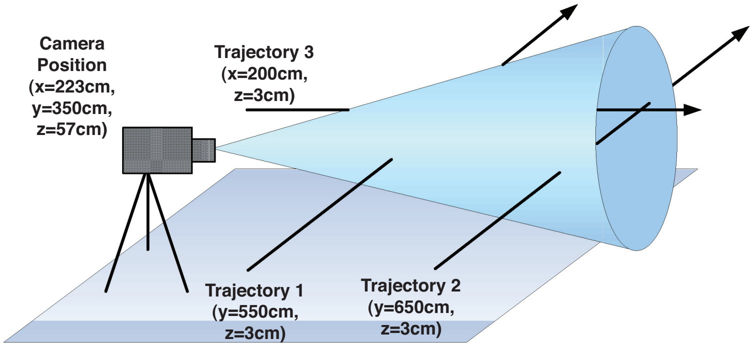
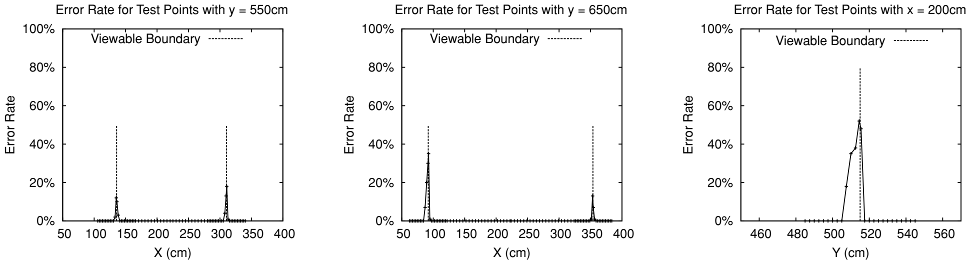


Fig. 7. The layout of static experiments using Cricket.



(a) Trajectory 1 (b) Trajectory 2 (c) Trajectory 3

Fig. 8. The error rate of static experiments using Cricket.

horizontally along the positive y axis and having 0° pitch and roll. We place a single object (simply a Cricket node) at different positions along the three trajectories ($y = 550$cm, $y = 650$cm, and $x = 200$cm). As most of the errors are made very close to the viewability boundary, we took readings every 2.5cm near the boundary, and every 5cm when the object was a least 30cm from the boundary.

For each object position we capture 100 frames and for each we record the 3D orientation of the camera and the coordinates of the camera and object. These coordinates are then fed into the SEVA system and we manually reviewed SEVA's results to evaluate the error rate (false positive for nonviewable objects and false negative for viewable objects). The results of this experiment are shown in Figure 8.

As shown in Figure 8(a) and 8(b), the error rate is less than 20% when the object is along the boundary, and the error rate quickly drops to single digits when the object is only 2.5cm away from the boundary and to zero when it is only 7.5cm away. As we are making a binary determination (visible/not visible), any small errors in location will not matter away from the boundaries. One exception to the small error rates occurs on Trajectory 2, and we get close to a 40% error rate when the object is along the viewable boundary. We believe that this is caused by interference with the ultrasound system from a nearby structural pillar.

Figure 8(c) shows that the error rate along the viewable boundary for Trajectory 3 is around 50%, and it drops to zero percent when the object is only 10cm away from the boundary. The reason for this larger error rate along the viewable boundary is that the ultrasound measured location of the camera is 5cm to 7cm lower than its real position, and the ultrasound measured location of the object is 2cm to

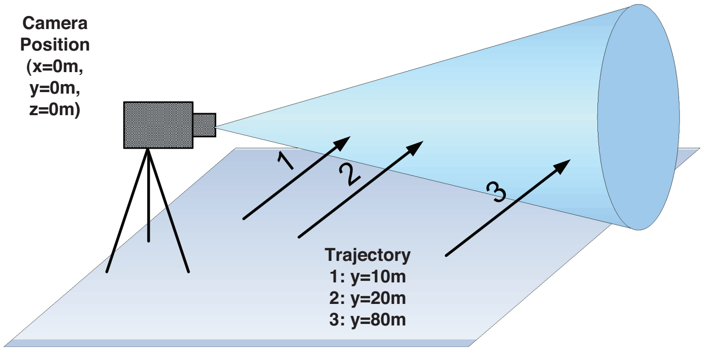
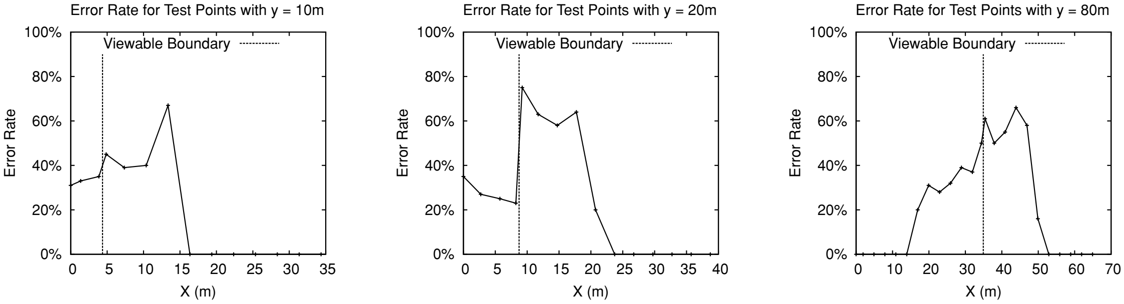


Fig. 9. The layout of experiments using GPS.



(a) Y = 10m (b) Y = 20m (c) Y = 80m

Fig. 10. The error rate of static experiments using GPS.

3cm higher than its real position in most cases. This type of error may come from the arrangement of Cricket reference points' position and could possibly be corrected by a different arrangement.

5.1.2 *GPS Locationing System.* We conducted a similar experiment with a GPS locationing system. GPS provides latitudes and longitudes relative to the equator and prime meridian; however, for readability we translate this coordinate system into (x, y) coordinates with the camera at the origin and the camera pointing along the y axis.

As shown in Figure 9, we used different positions along three trajectories: $y = 10$m, $y = 20$m, and $y = 80$m. The positions are separated by a 3m step size starting 30m from the viewable boundary and ending at the center of the field of view. For each position, we take 100 pictures, and for each picture we record the 3D orientation of the camera and the (x, y) coordinates of the camera and object. We then manually verify that SEVA produces the correct results and record the error rate. The results are shown in Figure 10.

Our results show that SEVA has more than a 20% error rate when the object is within 15 meters from the boundary, and when the distance to boundary is more than 18 meters the error rate drops to zero. The low performance is due to the low accuracy of GPS (5–15m). In light of this limitation, we have chosen to focus the remaining experiments on objects using the Cricket locationing system.

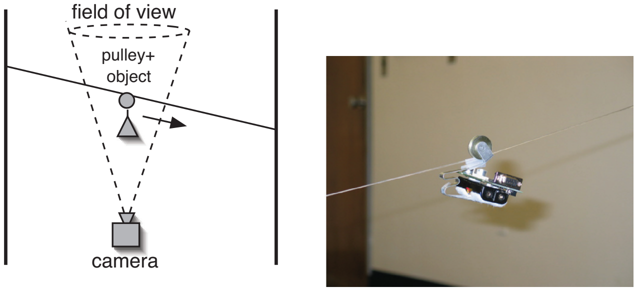


Fig. 11. Mobile object on a pulley.

	Slope 1	Slope 2	Slope 3
Gradient	7.6°	10.93°	19.47°
Length	$303cm$	$350cm$	$360cm$
AVG. Speed	$86.57cm/s$	$145.83cm/s$	$205.71cm/s$
Time	$3.5s$	$2.4s$	$1.75s$
Lengthin Viewable Area	$150cm$	$228cm$	$240cm$
AVG. Speedin Viewable Area	$112.06cm/s$	$181.90cm/s$	$271.77cm/s$
Timein Viewable Area	$1.34s$	$1.25s$	$0.88s$

Fig. 12. Characteristics of different slopes.

5.2 Dynamic Experiments

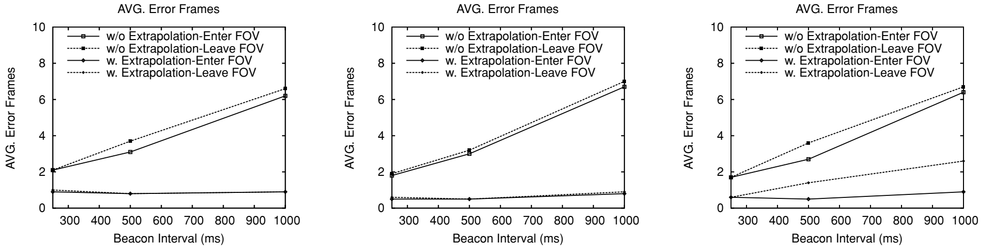
To evaluate SEVA's extrapolation and prediction mechanisms using the Kalman filter technique, we performed three sets of experiments: (i) mobile object with a stationary camera; (ii) stationary object with a mobile camera; and (iii) mobile object with mobile camera. The video clips were reviewed manually as before to determine which frames had erroneous annotations.

5.2.1 *Static Camera, Dynamic Objects.* When the object is moving and the camera is static the critical factor affecting SEVA's accuracy is the speed of the object relative to how often SEVA updates the object location. If the object speed is very high in relation to the object location, it will misextrapolate the object position and make mistakes in tagging objects as in or out of the field of view.

To explore this point we constructed two experiments: a repeatable experiment using a straight-line trajectory, and a nonrepeatable experiment using a semirandom path.

Repeatable Experiment. To construct a repeatable experiment we use an object moving at different speeds and updating its position at different intervals. In order to make the experiments as repeatable as possible we designed a test apparatus. We hung a fishing line across the camera's field of view at an angle and attached the object to a pulley (see Figure 11). When we release the object it accelerates down the line and then stops at the bottom. We can change the acceleration of the object by changing the gradient of fishing line. We accelerated the object across the camera's field of view using three different slopes: 7.6°, 10.93°, and 19.47° and the characteristics of these different slopes are shown in Figure 12. The object updates its position using the Cricket ultrasound system and it can reliably update its position at most once every 250ms. In this experiment we used three different beacon intervals: 250ms, 500ms, and 1000ms.

For each slope and each beacon interval, we encoded ten videos and used SEVA to determine the object's viewability in each frame. We manually compared SEVA's results with the original video on a frame-by-frame basis and evaluated which frame tags were in error. As before, incorrect decisions



(a) Slope 1 (b) Slope 2 (c) Slope 3

Fig. 13. Mean frames in error for a mobile object and static camera using curve fitting.

are made only when the object is close to the viewable boundary. In these experiments that situation occurs when the object either enters or exits the viewable area. The large number of frames in these experiments would make the error rate appear very small, so instead of presenting an error rate, we present the absolute number of frames that are in error. When later querying the video for sequences including a particular object, this metric determines how many extra or missing frames will be included or excluded from the sequence. The result is taken over the average of all ten experiments. We compare two systems: a full version of SEVA and a version of SEVA that does not perform any extrapolation. The results are shown in Figure 13.

The results demonstrate that without extrapolation the average number of frames in error increases from 1.8 to 7.0 as the beacon interval increases from 250ms to 1000ms. The slower beacon interval forces SEVA to use old measurements of the object's position and cannot correct for them using extrapolation. With extrapolation using the Kalman filter technique the average number of frames in error of both methods are less than 1 and are fairly constant across beacon intervals.

The worst case occurs when the object is exiting the viewable area under the highest acceleration and the beacon interval is the slowest. In this scenario the object leaves the viewable area at 375cm/s, reaches the end of the wire, and suddenly stops. This rapid deceleration causes the extrapolation method to fail and SEVA misplaces the object at intervening frame intervals. Given a faster beacon interval it is more likely that a beacon will occur after the object leaves the viewable area, but before the object stops. This means that two beacons straddle the exit from the viewable area and SEVA extrapolates the position correctly.

Nonrepeatable experiment. In the repeatable experiment, the object moves in a straight line, and SEVA uses a 250ms location beacon interval. Although this stresses SEVA's extrapolation system and gives us repeatability, it is relatively easy to estimate the linear path using the Kalman filter technique. To test a more complex path we recorded a new object: a remote control toy car with a Cricket node attached to the top (see Figure 14). We randomly moved the car around the room for 5 minutes while recording the car with SEVA. The car moved in and out of the camera's field of view many times during the experiment and we evaluated the performance in the same manner as before. Our results show that with extrapolation using the Kalman filter, the mean number of frames in error is around 1.9. This is only slightly larger than for the object moving in a straight line.

5.2.2 *Dynamic Camera, Static Objects.* If the camera is moving but the objects are static, SEVA must interpolate the position as well as the orientation of the camera. To test this function with a variety of movement patterns, we placed 4–5 objects separated by equal distance, and moved the camera in three patterns as shown in Figure 16: (a) *straight line*, the camera moves in a straight line without changing the orientation of the lens; (b) *rotation*, the camera moves and the lens' orientation changes;

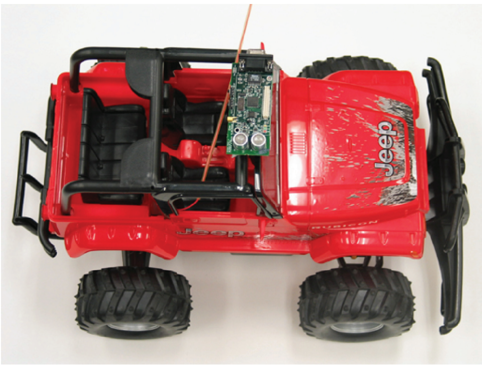


Fig. 14. Remote control toy car with a Cricket node on the top.

	Straight Line	Rotation	z Line
Slow	$50cm/sec$	$25^\circ/sec$	$50cm/sec$
Fast	$80cm/sec$	$60^\circ/sec$	$80cm/sec$

Fig. 15. Characteristics of different speeds.

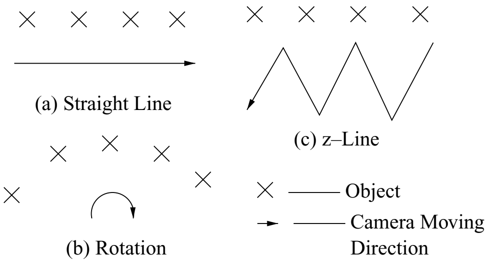


Fig. 16. Path of a mobile camera.

(c) *z-line*, the camera moves in a z-shaped line without changing its lens' orientation. We evaluated SEVA's performance using the frame error metric as before. For each movement pattern we ran experiments under two different speeds, labeled slow and fast. The characteristics of these speeds are shown in Figure 15. In all cases we used the full SEVA system with a location beacon interval of 250ms. Again, we only report the number of frames that are in error. The results are shown in Figure 17.

The results show that for the straight line the average number of error frames, which is less than 1.0, is comparable to when the object is moving and the camera is stationary. When the camera moves in a circle the average error frames is less than 2. We have traced these errors to variances in the digital compass's readings when the heading changes and the latency of the digital compass (up to 100ms). When the camera moves in a z-line the average error frames is around 1.2. Although we don't change the lens' heading, SEVA's interpolation fails when the camera makes a sharp turn, slightly increasing the average number of error frames.

5.2.3 *Dynamic Camera, Dynamic Object.* When both camera and object are dynamic, SEVA must extrapolate the position of object, and the position as well as the orientation of the camera. To test SEVA's performance under this scenario, we recorded the remote control car with a Cricket node attached to the top (see Figure 14) using a mobile camera. We randomly moved the car around the room for 10 minutes,

		Straight Line	Rotation	z - Line
Kalman Filter	Slow	0.77	1.74	1.19
	Fast	0.69	1.65	1.27

Fig. 17. Mean frames in error for a mobile camera.

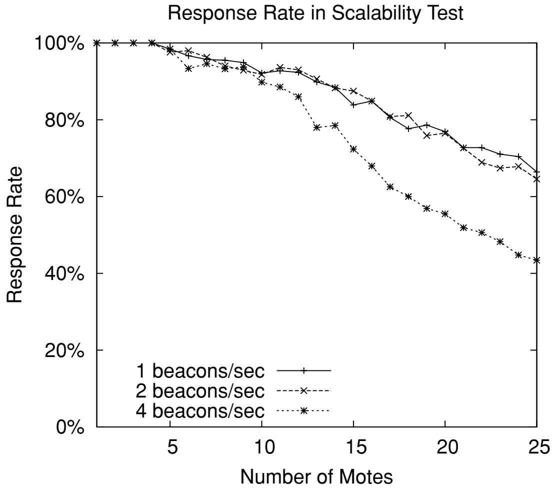


Fig. 18. Response rate of Motes.

and at the same time, we also randomly moved the camera around the room in the combinations of the moving paterns as shown in Figure 16. The car was in and out of the camera's field of view many times during the experiment and we evaluated SEVA's performance using the frame error metric as before. Our results show that with extrapolation using the Kalman filter the mean number of frames in error is around 3. This larger error is due to the movement of both camera and object. When both the camera and the object are mobile we have larger positioning errors and as a result we have a larger mean number of frames in error.

5.3 Scalability

As discussed in Section 3.2, the camera uses periodic broadcast messages to query for nearby objects. If there are a large number of objects within radio range, the radio's MAC layer may not scale to handle a large number of simultaneous responses. To test the scalability of our current prototype, we video recorded a large number of objects programmed with static locations.

To create a larger number of objects we used low-bitrate wireless sensor nodes called Motes [Hill and Culler 2002], specifically Mica2 and Mica2dots. These nodes are representative of future object tags due to their small size, low computational power, and low energy consumption. The Mica2 radio only supports a raw transmission rate of 19.2Kbps, and the effective throughput is 12.364Kbps or 42.93 packets/s.

The scalability of the system is determined by the frequency at which the camera sends queries relative to the number of objects and the rate the radio can handle. The maximum packet rate is fixed so we constructed an experiment with a variable number of objects and query frequencies. We measure the response rate, which is the ratio of responses the camera got (we only considered responses that were at most one beacon behind) compared to the number of objects. The results are shown in Figure 18.

The results show that the prototype can achieve a 100% response rate for up to 4 objects under all beacon frequencies. It achieves more than a 90% response rate for up to 10 responders under all beacon frequencies. However, the response rate for 4 beacons/s drops quickly and almost linearly with more than 10 responders, and it is 72.3% with 15 responders and 43.4% with 25 responders. The response rates for 1 beacon/s and 2 beacons/s are almost the same with up to 20 responders, while the response rate of 2 beacons/s drops more quickly than the response rate of 1 beacons/s after that.

A combination of these results with those of the dynamic object experiments indicates that the current prototype should scale well to 10 fast-moving objects. If the environment includes a mix of fast-moving objects and slow-moving objects, further scalability can be achieved if slow-moving objects respond less frequently to beacons.

5.4 Computational Requirements

We measured the computational requirements of each of SEVA's stages. The correlation and the extrapolation modules impose a small computational overhead on the laptop (less than $100\mu s$ for each object); the filtering module imposes a $200\mu s$ overhead for each object. Unlike GPS systems, the Cricket sensor gives the distances to beacons instead of 3D coordinates, thus the laptop must solve a set of linear equations to compute the 3D coordinates. This computation costs around $150\mu s$ for each object. These results show that our system incurs small overhead and will run online on relatively inexpensive hardware.

5.5 Summary and Discussions

Our experiments show that: (i) using Cricket, SEVA has zero error rates for static objects when the object is only 7.5–10cm away from the boundary; (ii) using GPS, SEVA has more than a 20% error rate when the object is within 15m from the boundary, and the error rate drops to zero when the distance to boundary is more than 18m; (iii) for moving objects or a moving camera SEVA only misses objects leaving or entering the viewable by 1–2 frames (80ms/frame); (iv) when both are moving SEVA only misses objects leaving or entering the viewable area by 3 frames; and (v) the SEVA prototype can scale well to 10 fast-moving objects using current sensor technology.

SEVA's performance and scalability are largely affected by the limits of current sensor technology: The larger error rate when using GPS is due to the low accuracy (5–15m) of current GPS technology; and the low scalability (no more than 10 fast-moving objects) is due to the low bandwidth (19.2Kbps) of Motes. However, we expect these problems will be solved in the near future as sensor technology and GPS evolve.

6. RELATED WORK

SEVA draws from several related research areas, which we survey here. Due to the overwhelming amount of related work in image retrieval, annotation, sensor systems, and locationing systems, we only highlight the most relevant work.

6.1 Content-Based Media Retrieval

Searching and retrieving media is greatly enhanced by textual annotations. There has been a great deal of work focused on content-based image retrieval in the community. Smeulders et al. surveyed this field in Smeulders et al. [2000]. They identified two obstacles that content-based image retrieval still must overcome in order to gain wide acceptance: the "sensory gap" and "semantic gap." The sensory gap is "the gap between the object in the world and the information in a (computational) description derived from a recording of that scene" [Smeulders et al. 2000]. For example, a car is recognized as two separate boxes but not as a car if there is a tree in front of it. Another example is that similar images from different objects may be recognized as images of the same objects. The semantic gap is “the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation” [Smeulders et al. 2000]. For example, a picture of a person’s birthday party should be seen by a vision system as a series of objects and people. The identification of these objects and people, the relationship between people, and the content of this event are not represented.

To bridge the sensory gap, using domain and world knowledge, researchers have proposed a combination of learning- and vision-based object/face recognition techniques [Fan et al. 2004; Feng et al. 2004; Jin et al. 2004; Li and Goh 2003; Nack and Putz 2004; Zhang et al. 2004]. A knowledge database includes several sources of general knowledge: the literal laws, perceptual laws, physical laws, geometric and topological rules, category-based rules, and man-made customs. In order to recognize an object, we must search through the database using these laws and rules. Building a comprehensive knowledge database requires a lot of human effort and experience. As a result, these techniques are error prone and have high computational requirements, and therefore are not suitable for large media libraries.

To bridge the semantic gap, researchers have proposed techniques which require users to manually enter the semantic context [Gemmell et al. 2002]. This manual procedure is normally done long after the image was captured and it requires human effort. Therefore, manual processing of each frame or image is cumbersome and faces the difficulty of imprecise human memory, and thus it is not suitable for large collections of media archives.

All these techniques try to generate the temporal, spatial, and social context of media capture, the *when*, *where*, *who*, and *what* to bridge the sensory and semantic gaps. However, these techniques are error prone, require user intervention, and are limited by human memory. Instead, the emergence of sensor technologies that can automate the collection of metadata can provide us with highly accurate context.

6.2 Sensor Annotation of Multimedia

Several systems automatically annotate images, videos, and audios with sensor data such as GPS readings, light readings, temperature readings, etc., and use these sensor data to help media retrieval [Aizawa et al. 2004; Davis et al. 2004; Ellis and Lee 2004; Gemmell et al. 2004; Naaman et al. 2003, 2004; Su et al. 2004; Toyama et al. 2003].

Su et al. augmented film and video footage with sensor data such as light intensity, color temperature, and location, and evaluated the latency involved between an event captured on video and on their system with light readings [Su et al. 2004]. Nack and and Putz attack the problem in the context of a video studio, recording information such as lens length and camera position for later retrieval [Nack and Putz 2004].

Toyama et al. built a World Wide Media eXchange (WWMX) database to share geo-tagged images with others [Toyama et al. 2003]. They acquire time and geographic tags from manual entry and by matching timestamps between photos and the GPS receiver. Combined with a map, the system allows users to effectively view images taken in specified locations.

Naaman et al. demonstrated how to use time and GPS location information about digital photographs to organize photo collections and help recall and finding photographs in Naaman et al. [2003]. In their later work, they proposed to automatically generate related contextual metadata such as the local daylight status and weather conditions at the time and place a photo was taken, and identified the categories of contextual metadata that are most useful for retrieving photographs [Naaman et al. 2004]. They acquire the time and GPS location information in the same way as Toyama et al.

Gemmell et al. built a passive camera SenseCam, that includes an accelerometer/tilt sensor, a passive infrared sensor, a digital light sensor, a temperature sensor, and a camera module [Gemmell et al. 2004].

In this system, changes in sensory data (e.g., light change, motion) and the time interval trigger the image capture. A handheld GPS provides the location readings, while in the future an onboard GPS will be used. The photos, the sensor data, and GPS readings are uploaded into the MyLifeBits database [Gemmell et al. 2002]. They use the date/time correlation between photos and GPS readings to set the location of the photos.

Davis et al. created a prototype "Mobile Media Metadata" (MMM) on Nokia 3650 camera phones [Davis et al. 2004]. Their prototype automatically stamps each photo with the time and date of image capture, the GSM network CellID for location, and the owner of the camera. Their prototype allows users to manually annotate the photo with present objects and activity. These contextual metadata can be combined and shared to infer the media content at a later time.

Aizawa et al. developed a wearable system equipped with sensors including GPS, gyroscope, accelerometer, and a brain wave sensor [Aizawa et al. 2004]. This system continuously captures video along with the sensor readings. The GPS data (time and location) are used to extract key frames which are subjected to sophisticated processing such as image analysis. They sample the key frames by time, by distance, by the speed of a movement, and by the changes of speed and direction. They also proposed and evaluated a conversation scene detection scheme using human voice detection and human face detection.

Somewhat different from tagging images or videos with time and location information, Ellis and Lee explored the possibility of tagging continuous audio archival with GPS positions to automatically collect information on changes in location and activity [Ellis and Lee 2004].

In summary, all of these systems automatically record two parameters of media capture, namely *when* and *where*; they cannot annotate media with the present objects, namely *who* and *what*, automatically. In addition to *when* and *where*, SEVA also records *who* and *what* are in each video frame, thereby providing a richer set of annotations and enabling greater content-based retrieval.

6.3 Sensor Systems

Small sensors can help us to sense the environment and locate and identify objects. Their output can be used to determine the context of media capture. A great deal of recent work has focused on developing new sensor platforms. Several hardware systems have been developed recently for portability, extensibility, and research prototyping, such as the Mica Motes [Hill and Culler 2002], Telos [Polastre et al. 2005], and the XYZ [Lymberopoulos and Savvides 2005]. These nodes consume anywhere from 10–70mW of power in active mode, and are designed for portability, extensibility, and research prototyping. RFID, both active and passive, has significant potential to provide low-cost, short-range identification for many consumer goods and can help identify objects in SEVA [Finkenzeller 2003].

6.4 Locationing Systems

A critical component in SEVA is the locationing system. Its accuracy, deployability, and cost are crucial factors in SEVA's success. The current prototype uses GPS [Bajaj et al. 2002], and the Cricket ultrasound system [Priyantha et al. 2000], but there are many other locationing systems available. Hightower and Borriello provide an excellent overview of current systems [Hightower and Borriello 2001]. RADAR [Bahl and Padmanabhan 2000] is a building-wide locationing system based on the IEEE 802.11 WaveLan wireless networking technology and has an accuracy of 3–4.3m. Active Badge [Want et al. 1992] is an Infrared-Based (IR) indoor locationing system with room-level locationing accuracy. Active Bat [Harter et al. 1999], like Cricket, is another ultrasound-based locationing system which can locate indoor objects within 9cm in 3D space. Additional work has also been done lately on the SpotON system [Hightower et al. 2000] and LANDMARC [Ni et al. 2003] as locationing systems for active RFID tags. LANDMARC can locate an object in 2D space with the accuracy of 1–3 meters, while a complete SpotON system has not been made available as of yet.

All of the aforesaid work uses battery-powered sensors to identify and locate objects. These sensors are expensive (at least tens of dollars per sensor) and have limited lifetime (from several days to several years). These limitations have prevented them from scaling to applications dealing with hundreds and thousands of objects. For example, it is not possible to attach a Cricket sensor to every book. In contrast, passive RFID tags are inexpensive (less than a dollar per tag and falling) and do not require a battery power source. These features make passive RFID technology an ideal choice when scaling positioning services to most objects.

Researchers have proposed two positioning systems [Hähnel et al. 2004; Liu et al. 2006] incorporating passive RFID technology. Hähnel et al. proposed a 2D navigation, mapping, and localization approach using the combination of a laser-range scanner, a robot, and passive RFID technology [Hähnel et al. 2004]. Their approach provides an accuracy of several meters. Ferret [Liu et al. 2006], proposed by us, is another scalable positioning system using passive RFID technology. By combining locationing technologies with multimedia applications, Ferret can locate objects using their attached passive RFID tags and displays their locations in real time to a mobile user. Our evaluation shows that Ferret can locate an object within a 0.02m^3 region.

7. CONCLUSION

This article presents the design and implementation of an automatic, sensor-enhanced video annotation and retrieval system named SEVA. It operates by querying nearby objects for their identities and locations, extrapolating and filtering those results, and recording this information with the video stream. Through a large set of experiments we have shown SEVA's overall effectiveness in tracking static and moving objects using a moving camera and two different locationing systems.

As part of our future work, we intend to integrate the RFID locationing systems such as SpotON, LANDMARC, and Ferret into SEVA. We are also developing techniques to infer video content from the context (*when, where, who,* and *what*) produced by SEVA. While SEVA currently filters out sensor readings for objects that are not in view, this extra information can certainly be retained and exploited by applications to indicate objects in the vicinity but not in the field of view; use of such information by applications is the subject of future research. We also plan to study the possibility of storing annotations within video streams via MPEG-7 [Manjunath et al. 2002].

APPENDIX A. EXTENDED KALMAN FILTER

This section provides the details of our extended Kalman filter that we use to extrapolate the path and location of a mobile object (see Figure 19). The derivation of our EKF is inspired by the work of Smith et al. [2004]. In the prediction step, suppose the predicted state vector at the $j-1$th position sample is S^-_{j-1}, the corrected state vector is S^+_{j-1}, and the reported position is $M_{j-1} = (x_{j-1}, y_{j-1}, z_{j-1})$. Assume the object moves at a constant velocity between position samples, the predicted state vector S^-_j at time Δt after the $j-1$th sample is given by $S^-_j = f(S^+_{j-1})$ as follows. We have

$$\begin{aligned}
x^- &= x^+_{j-1} + v^+_x * \Delta t \\
y^- &= y^+_{j-1} + v^+_y * \Delta t \\
z^- &= z^+_{j-1} + v^+_z * \Delta t \\
v^-_x &= v^+_x \\
v^-_y &= v^+_y \\
v^-_z &= v^+_z
\end{aligned} \tag{1}$$

in which we omit the subscript of (v^+_x, v^+_y, v^+_z) for clarity and (x^-, y^-, z^-) is the predicted position.

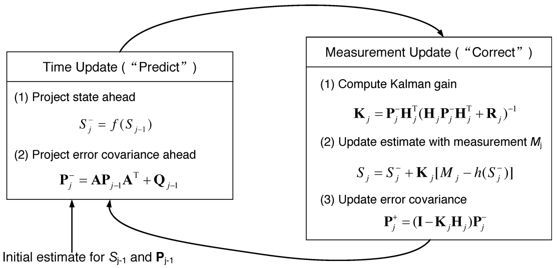

Time Update ("Predict")
(1) Project state ahead
$S_j^- = f(S_{j-1})$
(2) Project error covariance ahead
$\mathbf{P}_j^- = \mathbf{A}\mathbf{P}_{j-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}_{j-1}$
Measurement Update ("Correct")
(1) Compute Kalman gain
$\mathbf{K}_j = \mathbf{P}_j^-\mathbf{H}_j^{\mathrm{T}}(\mathbf{H}_j\mathbf{P}_j^-\mathbf{H}_j^{\mathrm{T}} + \mathbf{R}_j)^{-1}$
(2) Update estimate with measurement M_j
$S_j = S_j^- + \mathbf{K}_j[M_j - h(S_j^-)]$
(3) Update error covariance
$\mathbf{P}_j^+ = (\mathbf{I} - \mathbf{K}_j\mathbf{H}_j)\mathbf{P}_j^-$
Initial estimate for S_{j-1} and $\mathbf{P}_{j-1}$

Fig. 19. Use of Kalman filter to compute the path of a mobile object.

Assuming $\mathbf{P}$ is the 6 × 6 covariance matrix for the state vector, the covariance matrix is predicted as

$$\mathbf{P}_j^- = \mathbf{A}\mathbf{P}_{j-1}^+\mathbf{A}^T + \mathbf{Q}_{j-1}, \tag{2}$$

where $\mathbf{P}_{j-1}^+$ is the corrected covariance matrix after getting the $j-1$th position sample, $\mathbf{A}$ is the state transition matrix specific to our model and given by the Jacobians of function f, and $\mathbf{Q}_{j-1}$ reflects how the quality of the state vector degrades over time.

In the correction step, once having the jth position sample, we can correct the predicted state vector and covariance matrix by computing the weighted output between the predictions and the measurements based on their relative covariances. Suppose the jth position sample is M_j, and the variance matrix of the new position sample is $\mathbf{R}_j$ given by the object; the predicted state vector and covariance matrix is given by S_j^- and $\mathbf{P}_j^-$, respectively. We define a function $h(S)$ that computes the expected position of the object given a state vector S, and $\mathbf{H}$ is the Jacobians of h. Therefore, the corrected state vector S_j^+ and the corrected covariance matrix $\mathbf{P}_j^+$ is given by

$$\mathbf{K}_j = \mathbf{P}_j^-\mathbf{H}_j^T(\mathbf{H}_j\mathbf{P}_j^-\mathbf{H}_j^T + \mathbf{R}_j)^{-1}, \tag{3}$$

$$S_j^+ = S_j^- + \mathbf{K}_j[M_j - h(S_j^-)], \tag{4}$$

$$\mathbf{P}_j^+ = (\mathbf{I} - \mathbf{K}_j\mathbf{H}_j)\mathbf{P}_j^-, \tag{5}$$

where $\mathbf{K}_j$ is the Kalman gain. The Kalman gain represents our relative confidence in the predictions and the measurements is used to do the weighting computation. If the measurement noise is large, then $\mathbf{K}_j$ decreases, and the corrected output S_j^+ approaches the predicted state vector S_j^-. In contrast, if the measurement noise is small, then $\mathbf{K}_j$ increases, and the corrected output S_j^+ approaches the new measurement M_j.

REFERENCES

ADAMS, B., PHUNG, D., AND VENKATESH, S. 2006. Extraction of social context and application to personal multimedia exploration. In *Proceedings of the 14th Annual ACM International Conference on Multimedia (MULTIMEDIA '06)*. ACM Press, New York, 987–996.

AHERN, S., ECKLES, D., GOOD, N., KING, S., NAAMAN, M., AND NAIR, R. 2007. Over-exposed? Privacy patterns and considerations in online and mobile photo sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 357–366.

AIZAWA, K., TANCHAROEN, D., KAWASAKI, S., AND YAMASAKI, T. 2004. Efficient retrieval of life log based on context and content. In *Proceedings of the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experience (CARPE'04)*, 22–31.

APPAN, P. AND SUNDARAM, H. 2004. Networked multimedia event exploration. In *Proceedings of the 12th Annual ACM International Conference on Multimedia (MULTIMEDIA '04)*. ACM Press, New York, 40–47.

BAHL, P. AND PADMANABHAN, V. N. 2000. Radar: An in-building rf-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom'00)*, vol. 2, 775–784.

BAJAJ, R., RANAWEERA, S. L., AND AGRAWAL, D. P. 2002. Gps: Location-tracking technology. *Comput. 35,* 4, 92–94.

BARRY, B. 2005. Mindful documentary. Ph.D. thesis, Massachusetts Institute of Technology.

DAVIS, M., KING, S., GOOD, N., AND SARVAS, R. 2004. From context to content: Leveraging context to infer media metadata. In *Proceedings of the 12th Annual ACM International Conference on Multimedia (MM'04)*, 188–195.

DEVORE, J. L. 1999. *Probability and Statistics for Engineering and the Sciences*, 5th Ed. Brooks/Cole.

DOURISH, P. 2004. What we talk about when we talk about context. *Personal Ubiquitous Comput. 8,* 1, 19–30.

ELLIS, D. P. W. AND LEE, K. 2004. Minimal-impact audio-based personal archives. In *Proceedings of the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experience (CARPE'04)*, 39–47.

FAN, J., GAO, Y., AND LUO, H. 2004. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In *Proceedings of the 12th Annual ACM International Conference on Multimedia (MM'04)*, 540–547.

FENG, H., SHI, R., AND CHUA, T. 2004. A bootstrapping framework for annotating and retrieving www images. In *Proceedings of the 12th Annual ACM International Conference on Multimedia (MM'04)*, 960–967.

FINKENZELLER, K. 2003. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd Ed. John Willey & Sons.

GEMMELL, J., BELL, G., LUEDER, R., DRUCKER, S., AND WONG, C. 2002. Mylifebits: Fulfilling the memex vision. In *Proceedings of the 10th Annual ACM International Conference on Multimedia (MM'02)*, 235–238.

GEMMELL, J., WILLIAMS, L., WOOD, K., LUEDER, R., AND BELL, G. 2004. Passive capture and ensuing issues for a personal lifetime store. In *Proceedings of the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experience (CARPE'04)*, 48–55.

geocoder. Find the latitude and longitude of any us address. http://www.geocoder.us.

gpsdrive: Gpsdrive 2.09. http://www.gpsdrive.cc/.

GRIMM, R. 2002. System support for pervasive applications. Ph.D. thesis, University of Washington, *Department of Computer Science and Engineering*.

HÄHNEL, D., BURGARD, W., FOX, D., FISHKIN, K., AND PHILIPOSE, M. 2004. Mapping and localization with rfid technology. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'05)*, 1015–1020.

HARTER, A., HOPPER, A., STEGGLES, P., WARD, A., AND WEBSTER, P. 1999. The anatomy of a context-aware application. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, 59–68.

HIGHTOWER, J. AND BORRIELLO, G. 2001. Location systems for ubiquitous computing. *Comput. 34,* 8, 57–66.

HIGHTOWER, J., WANT, R., AND BORRIELLO, G. 2000. Spoton: An indoor 3D location sensing technology based on rf signal strength. Tech. rep. 00-02-02, University of Washington.

HILL, J. AND CULLER, D. 2002. Mica: A wireless platform for deeply embedded networks. *IEEE Micro 22,* 6, 1224.

HONG, J. I. AND LANDAY, J. A. 2004. An architecture for privacy-sensitive ubiquitous computing. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, 177–189.

JIN, R., CHAI, J. Y., AND SI, L. 2004. Effective automatic image annotation via a coherent language model and active learning. In *Proceedings of the 12th Annual ACM International Conference on Multimedia (MM'04)*, 892–899.

JOHANSON, B., FOX, A., AND WINOGRAD, T. 2002. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Comput. 1,* 2.

KINDBERG, T. AND ET. AL. 2002. People, places, things: Web presence for the real world. *Mobile Netw. 7,* 5.

LI, B. AND GOH, K. 2003. Confidence-based dynamic ensemble for image annotation and semantics discovery. In *Proceedings of the 11th Annual ACM International Conference on Multimedia (MM'03)*, 195–206.

LIU, X., CORNER, M., AND SHENOY, P. 2005. Seva: Sensor-enhanced video annotation. In *Proceedings of the 13th ACM Annual Conference on Multimedia (MM'05)*, 618–627.

LIU, X., CORNER, M., AND SHENOY, P. 2006. Ferret: Rfid localization for pervasive multimedia. In *Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp'06)*.

LYMBEROPOULOS, D. AND SAVVIDES, A. 2005. XYZ: A motion-enabled, power aware sensor node platform for distributed sensor network applications. In *Proceedings of Information Processing in Sensor Networks (ISPN)*.

MAINWARING, A., POLASTRE, J., SZEWCZYK, R., CULLER, D., AND ANDERSON, J. 2002. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, 88–97.

MANJUNATH, B. S., SALEMBIER, P., AND SIKORA, T. 2002. *Introduction to MPEG 7: Multimedia Content Description Language*, 4th Ed. John Wiley & Sons.

MEALLING, M. 2003. Auto-id object name service (ons) 1.0. Working Draft 12.

NAAMAN, M., HARADA, S., WANG, Q., GARCIA-MOLINA, H., AND PAEPCKE, A. 2004. Context data in geo-referenced digital photo collections. In *Proceedings of the 12th Annual ACM International Conference on Multimedia (MM'04)*, 196–203.

NAAMAN, M., PAEPCKE, A., AND GARCIA-MOLINA, H. 2003. From where to what: Metadata sharing for digital photographs with geographic coordinates. In *Proceedings of the 10th International Conference on Cooperative Information Systems (CoopIS'03)*, 196–217.

NACK, F. AND PUTZ, W. 2004. Saying what it means: Semi-automated (News) media annotation. *Multimedia Tools and Applications 22,* 3, 263–302.

NI, L. M., LIU, Y., LAU, Y. C., AND PATIL, A. P. 2003. Landmarc: Indoor location sensing using active rfid. In *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*. 407–417.

POLASTRE, J., SZEWCZYK, R., AND CULLER, D. 2005. Telos: Enabling ultra-low power wireless research. In *Proceedings of the 4th International Conference on Information Processing in Sensor Networks: Special Track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*.

PRIYANTHA, N. B., CHAKRABORTY, A., AND BALAKRISHNAN, H. 2000. The cricket location-support system. In *Proceedings of the 6th Annual ACM International Conference on Mobile Computing and Networking (MobiCom'00)*, 32–43.

ROMAN, M., HESS, C., AND CAMPBELL, R. 2002. Gaia: An oo middleware infrastructure for ubiquitous computing environments. In *ECOOP Workshop on Object-Orientation and Operating Systems*.

SIMON, D. 2006. *Optimal State Estimation*, 1st Ed. Wiley-Interscience.

SMEULDERS, A. W. M., WORRING, M., SANTINI, S., GUPTA, A., AND JAIN, R. 2000. Content-based image retrieval at the end of the early years. *IEEE Trans. Patt. Anal. Mach. Intell. 22,* 12, 1349–1380.

SMITH, A., BALAKRISHNAN, H., GORACZKO, M., AND PRIYANTHA, N. 2004. Tracking moving devices with the cricket location system. In *Proceedings of the 2nd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'04)*, 190–202.

SU, N. M., PARK, H., BOSTROM, E., BURKE, J., SRIVASTAVA, M. B., AND ESTRIN, D. 2004. Augmemting film and video footage with sensor data. In *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications (PerComm'04)*, 3–12.

TOYAMA, K., LOGAN, R., AND ROSEWAY, A. 2003. Geographic location tags on digital images. In *Proceedings of the 11th Annual ACM International Conference on Multimedia (MM'03)*, 156–166.

WANT, R., HOPPER, A., FALCAO, V., AND GIBBONS, J. 1992. The active badge location system. *ACM Trans. Inf. Syst. 10,* 1, 91–102.

ZHANG, L., HU, Y., LI, M., MA, W., AND ZHANG, H. 2004. Effective propagation for face annotation in family albums. In *Proceedings of the 12th Annual ACM International Conference on Multimedia (MM'04)*, 716–723.

Received September 2006; revised December 2007; accepted May 2008