SEVA: Sensor-Enhanced Video Annotation

Xiaotao Liu, **Mark Corner** and **Prashant Shenoy**

Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
{xiaotaol,shenoy,mcorner}@cs.umass.edu

Abstract

Advances in consumer electronics technologies have led to a proliferation of digital cameras and camcorders that record images and video in digital form and has encouraged users to create ever-larger personal libraries of pictures and movies. A concurrent trend is the emergence of numerous sensor technologies such as RFID and low-power sensors positioning technologies such as GPS and ultrasound.

This paper proposes a new multimedia application that is enabled by the confluence of these trends. In particular, we study how a sensor-rich world can be exploited by digital recording devices such as cameras and camcorders to improve an 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.

1 Introduction

Advances in consumer electronics technologies have led to a proliferation of digital 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. A concurrent trend is the emergence of numerous sensor technologies such as RFID [10] and low-power sensors [24]. In the future it is likely that many objects will be equipped with sensors that encode their identities. For instance barcodes on objects such as books and food will be replaced with RFID sensors that serve as electronic tags. Street signs, buildings, and popular locations might be equipped with active sensor beacons that electronically broadcast their addresses. Another trend is the ubiquitous deployment of positioning technologies such as GPS [2] and ultrasound [30] that triangulate the exact location of a user.

This paper proposes a new multimedia application that is enabled by the confluence of these trends. In particular, we study how a sensor-rich world can be exploited by digital recording devices such as cameras and camcorders to improve an 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)*, produces a tagged stream that later can be used to efficiently search for videos or frames containing particular objects or people.

SEVA is different from the numerous multimedia annotation systems that have been developed in the literature. Since manual annotation of each frame or image is cumbersome, numerous automated learning and vision-based techniques for annotation of visual content have been developed [7, 8, 19, 22, 26, 35]. In contrast, SEVA exploits pervasive sensors to record locations and identities of objects and uses this information to annotate and index video. Thus, SEVA is an alternate method for annotation and indexing and can complement existing annotation and retrieval techniques by providing a new dimension of information.

The notion of stamping each picture with the GPS coordinates of the camera has been proposed in the literature [1, 3, 12, 25, 34]—doing so enables the picture to be automatically annotated with the place where the picture was taken. SEVA substantially builds on this notion—it not only envisions recording the location of the camera but also the identities and locations of all objects in its vicinity.

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 records objects that do not actually appear 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 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 a moving camera causes objects to move in and out of the field of view. SEVA must correctly identifying which frames contain a particular object with a high degree of accuracy.
- **Inherent limitations of power-constrained, bandwidth-poor sensors:** Sensors attached to objects are either battery-powered of passive. Due to power-constraints, battery-powered sensors aggressively duty-cycle 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, and hence, are inherently resource constrained. 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 intra-frame duration, necessitating extrapolation techniques to annotate every frame.
- **Limitations of positioning systems:** SEVA requires a high degree of positioning accuracy in order to properly identify viewable objects. Unfortunately, the current current generation of positioning systems provide limited accuracy. For instance, current GPS technology provides accuracy of 3-100 meters [2], while handling moving objects in ultrasound has inherent problems [32]. SEVA must deal with the error that is introduced as a result of these limitations.

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 above 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 the regression techniques to determine the path of a mobile object and its location. To address the address the issue of resource-constrained sensors, we employ interpolation techniques to determine if an object is within range even if it did not respond to a query when the frame was captured. 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. Finally, buffering and filtering are used to handle some, but not all, of the inaccuracies of positioning systems.

These techniques enable a fully working prototype of SEVA. We conducted 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) SEVA prototype can scale well to 10 fast moving objects using current sensor technology; and (iv) SEVA runs online using relatively inexpensive hardware.

The rest of this paper is structured as follows. We present background and assumptions in Section 2. Section 3 presents the design of SEVA. We present implementation details in Section 4 and our experimental results in Section 5. Section 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 paper. 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 [32]. GPS is an outdoor positioning system that relies on satellites, and Cricket is an indoor system based on ultra-sound beacons. For passive sensors such as RFID we assume that they store their current coordinates and are reprogrammed using emerging RFID triangulation techniques [17, 27].

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 records 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 tilt of the camera for each frame, (ii) the GPS and/or Cricket coordinates of the camera for each frame, (iii) a GPS time stamp 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 a stream of sensor data and a video stream and fuses them together in a series of stages. Each step requires careful filtering and melding locationing information, object identification, and 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 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 time stamp 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 properties of the object.

Such infrastructures have been proposed for a broad array of systems [20, 21, 15, 31] 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.

The pervasive locationing and identification shown in Figure 1 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:

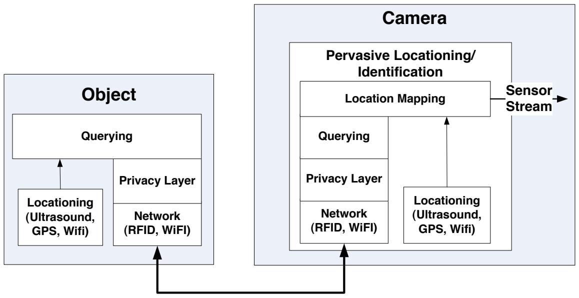


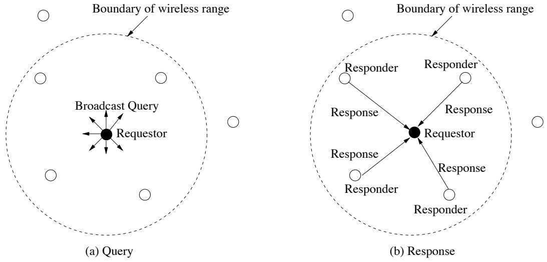
Figure 1: Pervasive Locationing/Identification System.

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 the evaluation section, the accuracy of these systems are extremely important 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 will support this feature through GIS information.

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 paper, 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 meta-information such as "a person" to an untrusted camera.

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 2.



(a) Query

(b) Response

Figure 2: Query and Response Model.

The locationing layer maps different object locations and camera locations to the same frame of reference. Different objects may use different locationing systems making it difficult to compute relative positions. For instance, GPS measures absolute positions using latitude, longitude and altitude, while Cricket measures relative positions. Since any coordinate system is relative to some frame of reference, as long as one coordinate system can be mapped into the other, SEVA can compute visibility. SEVA handles these differences by employing an internal coordinate system and a frame of reference relative to the camera, and maps all coordinates to internal coordinates, enabling 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 time stamp, it provides an estimate of the time from query to response. This includes MAC layer delays and internal processing. The recorder subtracts this delay from the receipt time of the response and assigns the corrected time stamp to the sensory information (propagation delays are assumed to be negligible). By performing this correlation, SEVA associates each query response to the appropriate frame.

3.4 Extrapolation and Prediction

Some per-object, per-frame location information will be missing from the correlated sensor stream. This is due to two factors. First, sensors duty-cycle 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.

SEVA explicitly deals with both of these scenarios by assuming that each query will obtain responses from only a

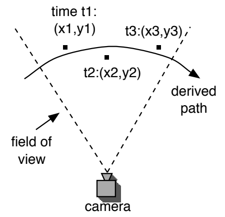


Figure 3: Deriving an object's path using curve fitting.

subset of the objects within radio range and employs post-processing techniques to account for missing responses. Depending on whether the objects and the camera are stationary or mobile, such interpolation is done as follows:

Static objects: If the objects and the camera 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 and a stationary camera—determining missing location information in this case requires a motion model. In particular, the module needs to extract determine the path (trajectory) of the object as a function of time. The location of the object at any instant can be then easily be determined.

SEVA uses regression techniques [5] to derive a smooth curve through the reported coordinates, which is then assumed to be the path taken by the mobile object. Assume that the object has responded to n queries. Suppose that the reported locations are (x_1, y_1, z_1), (x_2, y_2, z_2), $\ldots (x_n, y_n, z_n)$ at times $t_1, t_2, \ldots t_n$. If $n = 2$ then only two locations are known, and this technique reduces to a straight line between the two reported locations. When $n > 2$, regression attempts to fit a curve through the reported points. Since the fit is not exact, the curve that yields the least error can be chosen. See Figure 3 for an example.

Our regression technique systematically tries $n - 1$ different curves for the best fit: linear, a 2nd degree polynomial, 3rd-degree and so on. The polynomial can have a degree of up to $n - 1$ for n known locations. The coefficients of each polynomial function are then determined using the least squares method [5]. Finally, a coefficient of determination is computed, which quantifies the goodness of the fit. The polynomial with the highest coefficient of determination is chosen; if all polynomials report determination coefficients less than a threshold, then the path of the object is too erratic to be approximated by a smooth curve. In this case, we simply assume that the object moves in a straight line between two successive reported locations (i.e., approximate the path as a sequence of linear segments).

Since the objects reports X-axis coordinates of $x_1, x_2, \ldots x_n$ at times $t_1, t_2, \ldots t_n$, respectively, the regression analysis yields a k-degree polynomial, $1 \leq k \leq n - 1$ that represents its location along the X-axis as a function of time:

$$X(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_k t^k \tag{1}$$

where $a_0, a_1, \ldots a_k$ denote the coefficients as determined by the least squares method. Similarly, the location along the Y and the Z-axis as a function of time is obtained:

$$Y(t) = b_0 + b_1 t + b_2 t^2 + \ldots + b_k t^k \tag{2}$$

$$Z(t) = c_0 + c_1 t + c_2 t^2 + \ldots + c_k t^k \tag{3}$$

Together, the functions $X(t)$, $Y(t)$ and $Z(t)$ enable us to determine the X, Y and Z coordinates of the object for any time instant t between $[t_1, t_n]$. Thus, the missing location information can be determined for every intermediate frame.

Mobile camera: The final scenario is one where the camera itself is mobile; objects can be stationary or mobile. One approach to handle this scenario is to consider a frame of reference relative to the camera. In this frame of reference, the camera becomes stationary and the reported location coordinates of objects are translated to this new frame of reference. Doing so reduces this scenario to the previous case of mobile objects and a stationary camera. However, this can yield errors, since a stationary object seen by a moving camera now becomes a mobile object relative to the camera. Similarly, in this frame of reference simple paths of objects (e.g., an object moving in a straight line) now become more complex trajectories.

Consequently, rather than considering locations that are relative to the camera, SEVA considers the *absolute* locations of both the camera and the objects and uses intelligent filtering techniques to account for the motion of both entities. In particular, SEVA considers the *actual* reported locations of objects and determines a trajectory of the object using regression techniques as explained above. The SEVA recorder is assumed to log the location of the camera for every single frame; since fine grain location information for the camera is already available, no interpolation is necessary.

Extrapolation: Our regression technique 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 trajectory computed by the regression analysis can be used to extrapolate this information and annotate a small number of frames before t_1 and after t_n. Extrapolation of the path beyond the intervals $[t_1, t_n]$ enables us to eliminate some of these false negatives. 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. Currently, our prototype uses a configurable parameter to determine 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. Although the figure only shows a one dimensional model, 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.

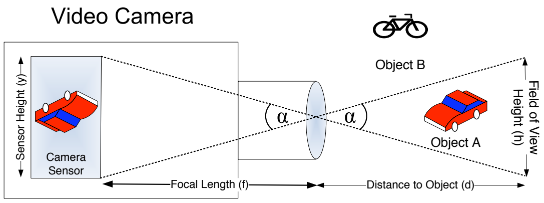


Figure 4: The Basic Optics Model

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 can then provide a measurement of the depth-of-field of the image. This allows us to compute whether objects are in or out of focus and tag them appropriately. SEVA does not include either of these mechanisms yet.

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 by the video's frame index. 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* is in the video. The search engine then searches video annotations produced by SEVA and returns video clips satisfying the query.

4 Implementation

To provide a test platform, we have constructed a prototype system based on a Sony Motion Eye web-camera

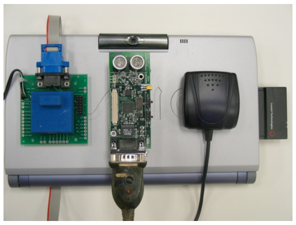


Figure 5: SEVA recorder laptop equipped with a camera, a 3D digital compass, a Mote with wireless radio and Cricket receiver, a GPS receiver, and 802.11b wireless.

connected to a Vaio laptop. The location and identity querying, correlation, extrapolation and prediction, filtering and elimination, and database storage software runs on the laptop. SEVA currently uses two 3-D 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 320x240 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 [9]) to record video.

Pervasive Location/Identification. Outdoors, SEVA uses Deluo GPS receivers equipped with WAAS correction [4], 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 [4].

Indoors, SEVA employs an ultrasound locationing system called Cricket [30]. Using a network of ultrasound sensors built onto sensor boards, Cricket can provide 3-D 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.

To provide an accurate position in active mode, at least four fixed sensors must receive the object's beacon. Each receiver computes its distance to the object and sends a radio message to the laptop. The laptop then computes the location of the object using a set of linear equations [29]. When the object is not not moving this scheme works quite well. When moving, receivers sometimes provide inaccurate distance estimates, and/or the object fails to reach four ultrasound receivers simultaneously.

To correct for these, we chose to use: (i) a simple filtering scheme to filter out the obviously incorrect distance measurements; for example, a distance measurement to a reference point that changes significantly from the previous measurement while the measurements to other reference points only change by a small amount; (ii) a simple buffering scheme, rather than a complex filter as has been previously proposed [32], to deal with the case of object failing to reach four ultrasound receivers simultaneously. This scheme has a buffer to store the most recent valid distance measurements to each reference point, and we use the most recent distance measurements in this buffer to compensate for missing data.

The pervasive locationing and identification system uses two different network layers to communicate with the objects. Outdoors objects are laptops equipped with WiFi and indoors objects are Mica2 [18] 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. These particular sensors can handle a limited number (42.93) of messages per-second necessitating a higher-layer backoff layer when using large numbers of objects. To reduce MAC contention, objects wait for a random period before sending a message and give up after some number of unsuccessful attempts. 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 Cricket system doesn't provide such a globally synchronized clock, SEVA simply correlates the location information with specific frames via subtracting the mean processing and MAC layer delay from the receiving time of sensor data and assigning the corrected time stamp to the sensory information.

Extrapolation and Prediction. As discussed in Section

3.4, we use regression analysis to find the mathematical relationship between location and time. Because the camera's 3D orientation will affect the result of filtering and elimination, we also apply regression analysis on camera's 3D orientation when their data are missed. In order to reduce the computational complexity, we use the frame index instead of the real clock time to represent the time.

Filtering and Elimination. In this stage, objects' coordinates are transformed into coordinates of space with camera as the origin and centimeter as the tick unit. This transformation is quite straightforward for Cricket system since we can easily subtract the camera's coordinate from objects' coordinates. The transformation for GPS system requires computing the distance between camera and object, and we use the GPS Drive package for this purpose [14].

Indexing and Querying . The results of filtering and elimination are put into a MySQL database. 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* is present in the video (e.g., car, book, building) and retrieves all annotated frames that match this query.

5 Experimental Evaluation

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

- 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 x 10m x $3m$ 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 videos at a rate of 12.5 frames/s.

To determine SEVA's accuracy in tagging frames, we subject the system to four 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, and d) the object is moving with semi-random trajectories and the camera is static. 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 by straddling the line between viewable and non-viewable. 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 6. The camera is set up at $(223, 350, 57)$ with its lens pointing 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: $y = 550cm$, $y = 650cm$ and $x = 200cm$. As most of the errors are made very close to

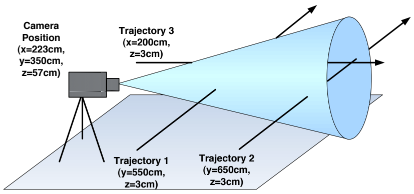

Camera Position (x=223cm, y=350cm, z=57cm)
Trajectory 3 (x=200cm, z=3cm)
Trajectory 1 (y=550cm, z=3cm)
Trajectory 2 (y=650cm, z=3cm)

Figure 6: The layout of static experiments using Cricket.

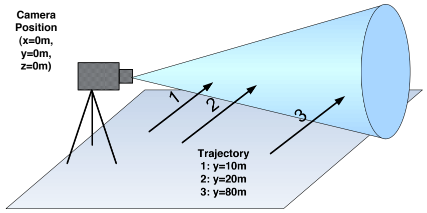


Figure 8: The layout of experiments using GPS.

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 take 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 non-viewable objects and false negative for viewable objects). The results of this experiment are shown in Figure 7.

As shown in Figure 7(a) and 7(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. One exception occurs on Trajectory 2, and we get close to 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 7(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 measured location of the camera is $5cm$ to $7cm$ lower than its real position, and the measured location of the object is $2cm$ to $3cm$ 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 of the Cricket reference points.

5.1.2 GPS Locationing System

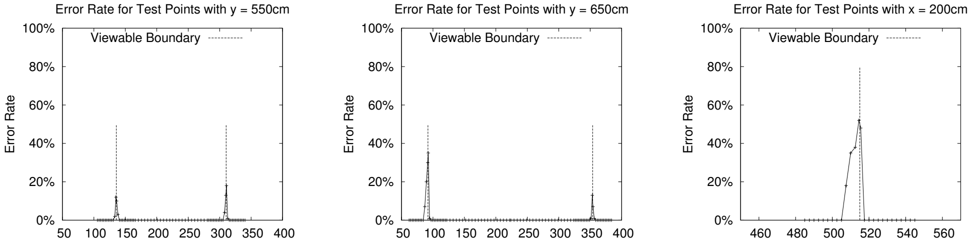
We conducted a similar experiment with a GPS locationing system. GPS provides latitudes and longitudes relative to the equator and prime meridian; however, for readability we translate this coordinate system into (x, y) coordinates with the camera at the origin and the camera pointing along the Y axis.

As shown in Figure 8, we used different positions along three trajectories: $y = 10m$, $y = 20m$, and $y = 80m$. The positions are separated by a $3m$ step size starting $30m$ from the viewable boundary and ending at the the center of the field of the 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 (false positive for non-viewable objects and false negative for viewable objects). The results are shown in Figure 9.

Our results show that SEVA has more than 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); however, we expect that SEVA's performance using GPS will increase dramatically in a few years as GPS is expected to reach $1 - 5m$ accuracy by the year 2013 with further improvements after 2016 [13].

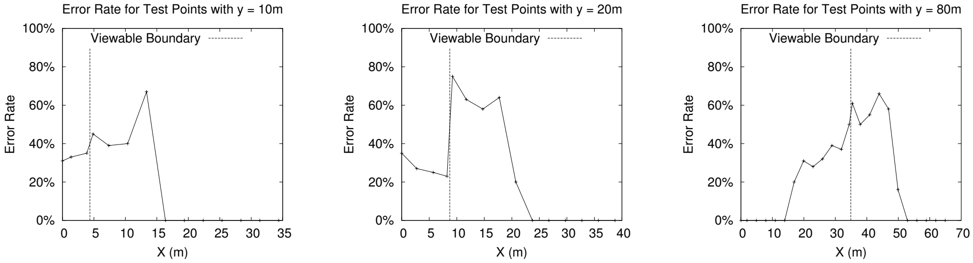
5.2 Dynamic Experiments

To evaluate SEVA's extrapolation and prediction mechanisms, we performed two sets of experiments: (i) mobile object with a stationary camera and (ii) stationary object



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

Figure 7: The error rate of static experiments using Cricket.



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

Figure 9: The error rate of static experiments using GPS.

with a 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 mis-extrapolate 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 non-repeatable experiment using a semi-random 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 10). 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 Table 1.

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 of each frame. We manually compared SEVA's results with the original video on a frame-by-frame basis and evaluate which frame tags were in error.

As before, incorrect decisions are made only when the

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

Table 1: Characteristics of different slopes.

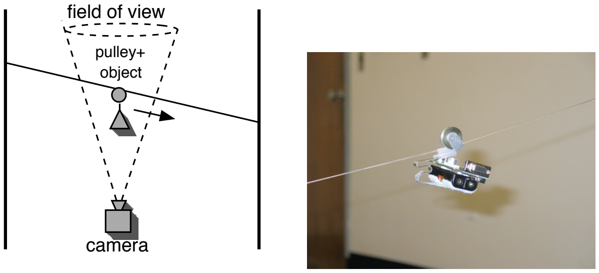


Figure 10: Mobile object on a pulley.

object is close to the viewable boundary. In these experiments that occurs either when the object 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 11.

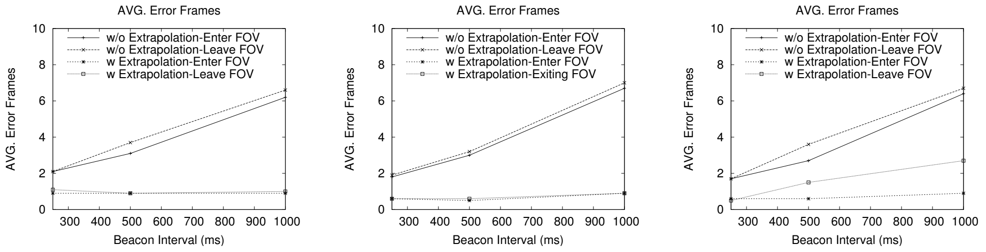
The results demonstrate that without extrapolation the average number of frames in error increases from 1.8 to 7.0 as the beacon interval increases from $250ms$ to $1000ms$. The slower beacon interval forces SEVA to use old measurements of the object's position and cannot correct for them using extrapolation. With extrapolation the average number of frames in error is less than 1 and is 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 $375cm/sec$, 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.

Non-Repeatable Experiment: In the repeatable experiment, the object moves in a straight line. Although this stresses SEVA's extrapolation system, it does not require higher-order regression analysis to determine the linear path. To test a more complex path we recorded a new object: a remote control toy car with a Cricket node attached to the top. We randomly moved the car around the room for 5 minutes while recording the car with the SEVA system. The car moved in and out of the camera's field of many times during the experiment and we evaluated the performance in the same manner as before. Our results show that the mean number of frames in error is 2. This is only slightly larger than object moving in a straight line.

5.2.2 Dynamic Camera, Static Object

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 12: (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; (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 Table 2. In all cases we used the full SEVA system with a location bea-



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

Figure 11: Mean frames in error for a mobile object and static camera.

	Straight Line	Rotation	z Line
Slow	$50cm/sec$	$25°/sec$	$50cm/sec$
Fast	$80cm/sec$	$60°/sec$	$80cm/sec$

Table 2: Characteristics of different speeds.

con interval $250ms$. Again we only report the number of frames that are in error. The results are shown in Table 3.

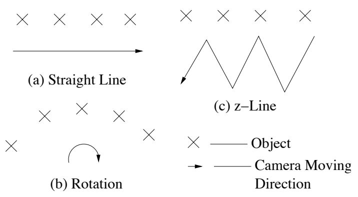


Figure 12: Path of a mobile camera.

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 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.

	Straight Line	Rotation	z - Line
Slow	0.8	1.78	1.2
Fast	0.7	1.67	1.3

Table 3: Mean frames in error for a mobile camera.

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-bit rate wireless sensor nodes called Motes [18], 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 19.2 Kbps, and the effective throughput is 12.364 Kbps or 42.93 packets/sec. Coincidentally, the packet rate is similar to the rate at which RFID tags (another possible object tag) can be queried.

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 of messages 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 13.

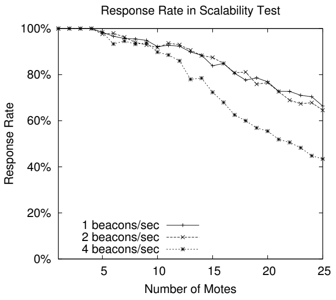


Figure 13: Response rate of Motes.

The results show that the prototype can can achieve 100% response rate for up to 4 objects under all beacon frequencies. It achieves more than 90% response rate for up to 10 responders under all beacon frequencies. However the response rate for 4 beacons/sec 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/sec and 2 beacons/sec are almost the same with up to 20 responders, while the response rate of 2 beacons/sec drops quicker than the response rate of 1 beacons/sec after that.

A combination of these results with those of the dynamic object experiments indicate 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 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 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.

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.

Content-based media retrieval: Searching and retrieving media is greatly enhanced by textual annotations. The annotations are either manually entered [11] or automatically generated by a combination of learning- and vision-based object/face recognition techniques [7, 8, 19, 22, 26, 35]. Manual annotation 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. Automatic annotation by the learning and vision-based techniques is error prone and has high computational requirements.

Sensor Annotation of Multimedia: Several systems annotate images, videos, and audio with sensor data such as GPS readings, light readings, temperature readings [1, 3, 6, 12, 25, 33, 34], and use these sensor data to help media retrieval. Many of these systems automatically tag images with time and GPS coordinates of where the image was taken, and then infer other information about the image later. All of these systems only record two parameters of video capture—*when* and *where*, unlike SEVA which also records *what* objects are in each video frame, thereby providing a richer set of annotations.

Sensor Systems: A great deal of recent work has focused on developing new sensor technologies. Several hardware platforms have been developed recently, such as the Mica Motes [18], Telos [28], and the XYZ [23]. 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 [10].

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 [2], and the Cricket ultrasound system [30], but there are many other locationing systems available. Hightower and Borriello provide an excellent overview of current systems [16]. Additional work has also been done lately on the SpotON system [17] and LANDMARC [27], as locationing systems for RFID tags, providing another locationing system for future SEVA systems.

7 Conclusions

This paper 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.

References

[1] K. Aizama, D. Tancharoen, S. Kawasaki, and T. Yamasaki. 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), New York, NY*, pages 22–31, October 2004.

[2] R. Bajaj, S. L. Ranaweera, and D. P. Agrawal. Gps: Location-tracking technology. *Computer*, 35(4):92–94, March 2002.

[3] M. Davis, S. King, N. Good, and R. Sarvas. From context to content: Leveraging context to infer media metadata. In *Proceedings of the 12th annual ACM International Conference on Multimedia (MM'04), New York, NY*, pages 188–195, October 2004.

[4] Deluo gps waas. http://www.deluoelectronics.com/.

[5] J. L. Devore. *Probability and Statistics for Engineering and the Sciences*. Brooks/Cole, fifth edition, 1999.

[6] D. P. W. Ellis and K. Lee. Minimal-impact audio-based personal archives. In *Proceedings of the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experience (CARPE'04), New York, NY*, pages 39–47, October 2004.

[7] J. Fan, Y. Gao, and H. Luo. 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), New York, NY*, pages 540–547, October 2004.

[8] H. Feng, R. Shi, and T. Chua. A bootstrapping framework for annotating and retrieving www images. In *Proceedings of the 12th annual ACM International Conference on Multimedia (MM'04), New York, NY*, pages 960–967, October 2004.

[9] Ffmpeg 0.4.8. http://ffmpeg.sourceforge.net/index.php.

[10] K. Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Willey & Sons, second edition, 2003.

[11] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong. Mylifebits: Fulfilling the memex vision. In *Proceedings of the 10th annual ACM International Conference on Multimedia (MM'02), Juan Les Pins, France*, pages 235–238, December 2002.

[12] J. Gemmell, L. Williams, K. Wood, R. Lueder, and G. Bell. 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), New York, NY*, pages 48–55, October 2004.

[13] Why modernize gps? http://www.gps.oma.be/gb/modern_gb_ok_css.htm.

[14] Gpsdrive 2.09. http://www.gpsdrive.cc/.

[15] R. Grimm. *System support for pervasive applications*. PhD thesis, University of Washington, Department of Computer Science and Engineering, December 2002.

[16] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, August 2001.

[17] J. Hightower, R. Want, and G. Borriello. Spoton: An indoor 3d location sensing technology based on rf signal strength. Technical Report 00-02-02, University of Washington, 2000.

[18] J. Hill and D. Culler. Mica: a wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):1224, November/December 2002.

[19] R. Jin, J. Y. Chai, and L. Si. 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), New York, NY*, pages 892–899, October 2004.

[20] B. Johanson, A. Fox, and T. Winograd. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1(2), 2002.

[21] T. Kindberg and et. al. People, places, things: Web presence for the real world. *Mobile Networks*, 7(5), October 2002.

[22] B. Li and K. Goh. Confidence-based dynamic ensemble for image annotation and semantics discovery. In *Proceedings of the 11th annual ACM International Conference on Multimedia (MM'03), Berkeley, CA*, pages 195–206, November 2003.

[23] D. Lymberopoulos and A. Savvides. XYZ: A motion-enabled, power aware sensor node platform for distributed sensor network applications. In *Proceedings of Information Processing in Sensor Networks (ISPN)*, Los Angeles, CA, April 2005.

[24] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02), Atlanta, GA*, pages 88–97, September 2002.

[25] M. Naaman, A. Paepcke, and H. Garcia-Molina. 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), Catania, Sicily*, pages 196–217, November 2003.

[26] F. Nack and W. Putz. Designing annotation before it's needed. In *Proceedings of the 9th annual ACM International Conference on Multimedia (MM'01), Ottawa, Canada*, pages 251–260, September 2001.

[27] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. Landmarc: Indoor location sensing using active rfid. In *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom'03), Dallas-Fort Worth, TX*, pages 407–417, March 2003.

[28] J. Polastre, R. Szewczyk, and D. Culler. 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)*, April 2005.

[29] G. Pottie and W. Kaiser. *Principles of Embedded Network Systems Design*. Cambridge University Press, first edition, 2005.

[30] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual ACM International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA*, pages 32–43, August 2000.

[31] M. Roman, C. Hess, and R. Campbell. Gaia: An oo middleware infrastructure for ubiquitous computing environments. In *ECOOP Workshop on Object-Orientation and Operating Systems*, Malaga, Spain, June 2002.

[32] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha. Tracking moving devices with the cricket location system. In *Proceedings of the 2nd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'04), Boston, MA*, pages 190–202, June 2004.

[33] N. M. Su, H. Park, E. Bostrom, J. Burke, M. B. Srivastava, and D. Estrin. Augmemting film and video footage with sensor data. In *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications (PerComm'04), Orlando, FL*, pages 3–12, March 2004.

[34] K. Toyama, R. Logan, and A. Roseway. Geographic location tags on digital images. In *Proceedings of the 11th annual ACM International Conference on Multimedia (MM'03), Berkeley, CA*, pages 156–166, November 2003.

[35] L. Zhang, Y. Hu, M. Li, W. Ma, and H. Zhang. Effective propagation for face annotation in family albums. In *Proceedings of the 12th annual ACM International Conference on Multimedia (MM'04), New York, NY*, pages 716–723, October 2004.