Design and Implementation of a Dual-Camera Wireless Sensor Network for Object Retrieval

Dan Xie, Tingxin Yan, Deepak Ganesan, Allen Hanson
Department of Computer Science, University of Massachusetts Amherst
{dxie, yan, dganesan, hanson}@cs.umass.edu

Abstract

This paper presents the design and implementation of a dual-camera sensor network that can be used as a memory assistant tool for assisted living. Our system performs energy-efficient object detection and recognition of commonly misplaced objects. The novelty in our approach is the ability to tradeoff between recognition accuracy and computational efficiency by employing a combination of low complexity but less precise color histogram-based image recognition together with more complex image recognition using SIFT descriptors. In addition, our system can seamlessly integrate feedback from the user to improve the robustness of object recognition. Experimental results reveal that our system is computation-efficient and adaptive to slow changes of environmental conditions

1. Introduction

As the baby boomer generation ages, health care for the elderly poses a major challenge over the next decade. Wireless sensor networks have often been referred to as the technology that can provide an affordable solution to this problem. Consequently, many research projects have explored the use of wireless sensors for medical care at home including a combination of wearable and ambient sensors for vital sign, gait, and fall monitoring [6, 19]. In addition to monitoring for illnesses and potentially life-threatening situations, an equally important challenge in home healthcare of the elderly is providing assistance in their day-to-day life. Moderate memory impairment is common as people age, hence a major problem for the elderly is locating frequently used "common" objects such as keys, cellphones, PDAs, books and others.

There has been considerable recent interest in addressing the problem of "object finding" both in academia and industry. Many of these services seek to attach wireless tags on the object, such as RFIDs [9, 17, 35], Bluetooth chips [15], or 802.15.4 radios [23, 25], making it easier to localize the object. While this offers a feasible solution for objects such as car keys and PDAs that already have wireless tags on them, the solution is cumbersome since it requires that every possible object that may be misplaced needs to be tagged a priori. Other approaches utilize visual information [24, 34], e.g., the ASSIST project proposed the use of PTZ (pan-tilt-zoom) cameras for object finding [34]. However, such PTZ cameras are usually bulky and power hungry, making them hard to deploy around homes.

In this paper, we explore how low-power camera sensors can be distributed in a home environment to facilitate retrieval of misplaced objects. Small, battery powered cameras are portable, easy to deploy, and can be densely deployed for greater coverage. In addition, such as system can be used across many untagged objects at home without requiring that each object be tagged individually. While there have been many efforts in recent years to design low-power smart camera networks for surveillance, object tracking, and object detection [14, 31, 16, 7], our work is fundamentally different in that we focus on achieving energy-efficient recognition of commonly misplaced objects at home using such a camera network.

The design of an object recognition system using low-power cameras poses significant technical challenges. The first challenge is that state-of-art image matching techniques (e.g. SIFT [18]) require complex processing that is computationally intensive on embedded processors that are typically used on sensor nodes, and consequently energy-intensive. This makes it essential to develop techniques that are less complex and consume less energy but can still enable robust image recognition. The second challenge is that image recognition typically requires a high-end sensor device with a high-resolution camera, and substantial computation and memory resources. However, the use of higher-end sensor platforms (e.g. iMote2 [3]) comes at the cost of energy-efficiency, and consequently lowers sensor lifetime. One commonly proposed approach is to use a multi-tier network [16], where a low-power and low-resolution wireless camera node (e.g. Cyclops [5]) is used for object detection, and wakes up a higher power camera to perform object recognition only when needed. However, this does not entirely solve the problem since the high-power camera still needs to wakeup and take images periodically to update its background model.

1.1. Contributions

This paper presents the design and implementation of a low-power wireless camera network that can be used as a memory assistant tool to find a small set of pre-selected common objects (cup, cellphone, etc) around the house. The contributions of our system are summarized below.

Energy-efficient Region-of-object Estimation: Our first contribution is robust object detection and region-of-object (ROO) estimation using a dual-camera platform that combines a low-power Cyclops camera and a higher power iMote2 camera. Our dual-camera platform is configured such that the Cyclops and iMote2 camera lenses have very similar fields-of-view (FOV). A key contribution of our work is that we exploit the Cyclops camera not only for object detection but also to determine where the object is located relative to the iMote2 camera's image by using a SIFT (Scale-invariant Feature Transform) [18] based image-mapping algorithm. Such a technique eliminates the need for background detection at the high-power camera, thereby saving energy.

Energy-efficient Object Recognition: Image processing techniques for object recognition typically use the SIFT algorithm, which is computationally expensive and requires at least a few seconds of processing per image (on low power processors of the type used here). A second contribution of our work is the use of two types of features - color and SIFT features - in order to reduce the energy required for object recognition. The color and SIFT features are applied in a cascading manner which reduces energy consumption without a significant drop in accuracy. For every new object, the color features are extracted quickly with limited computation, and the object is classified using a semi-supervised clustering algorithm. If the color features provide an accurate match, no further action is taken. If not, SIFT features are extracted for the object and used for more accurate classification. Our results show that this combination of techniques almost three times more efficient than a scheme that only uses SIFT features, while being only 12% less accurate across a range of illuminations.

Incorporating User Feedback: A unique aspect of our system is the ability to incorporate user feedback to continually improve the results of object classification. The sensor proxy provides a simple GUI to enable an elderly user to query for objects that have been misplaced. The system then returns to the user the candidate object images that are most likely to match the query. The user is provided the option of confirming the results and labeling the correct categories of the images, which provides valuable real-time feedback to the system to refine the data model of semi-supervised clustering. Such user feedback enables more robust object recognition in the presence of environmental dynamics.

The remainder of this paper is structured as follows. Section 2 presents an overview of our system. Section 3 and 4 present the techniques of object detection and recognition. Section 5 presents the method for tag-based object retrieval. In Section 6 we give the implementation details of our system. The experimental results are presented in Section 7. Section 8 discusses related work. We present our conclusions and future works in Section 9.

2. System overview

In this section we present an overview of the system. The software and hardware architectures of our system are shown in Figure 1.

2.1. System framework

Our system involves a network of dual-camera nodes, each of which comprises of a low-power and high-power tier that are physically connected together as shown in Figure 2. The low-power camera sensor node (Tier-1) comprises of a MICAz mote [1] equipped with a low fidelity Cyclops camera sensor (CyclopsCam) [5, 28], and a 1GB NAND flash for storing images [21]. The high-power camera sensor node (Tier-2) comprises of a more-capable platform, the Intel Mote2 (iMote2) [3] equipped with a high fidelity Enalab camera (EnalabCam) [2], and a 1GB SD card for image storage. The two cameras on the dual-camera node are placed close enough so that they have similar field-of-view. User queries in our system can be posed from a PC or PDA that is connected to an 802.15.4 wireless radio, and can communicate with the MICAz and iMote2 nodes.

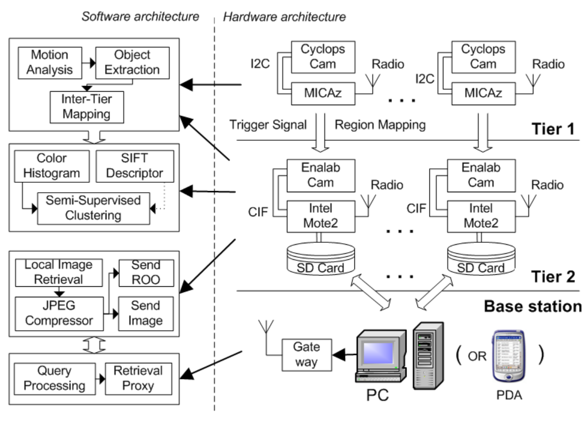


Figure 1. System architecture

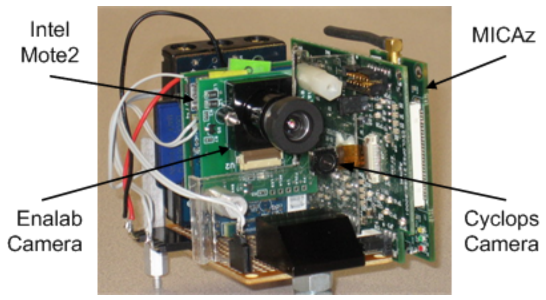


Figure 2. The dual-camera node

2.2. System operation

The operation of our system can be divided into three main components: object detection, object recognition and image retrieval. The Tier-1 low-power camera takes an image every few seconds, and performs still object detection, i.e., it determines if an object that is detected is likely to be a misplaced object as opposed to a moving object. If a still object is detected, the Tier-1 camera stores the location and size of this object in its local flash memory store. Once a batch of still objects have been detected, the Tier-1 camera wakes up the Tier-2 node and transfers the stored images together with information about the region in the image where the object was detected.

The Tier-2 node uses an inter-camera region mapping function to map the Tier-1 ROOs to its own camera coordinates. This enables it to determine which regions in its own view correspond to the new objects. Next, the Tier-2 node takes an image using its high-resolution camera, extracts the ROOs corresponding to new objects, and obtains the color histogram corresponding to each ROO. The object recognition procedure first tries to recognize the object in each ROO by using the color histogram together with a semi-supervised k-means clustering. For objects that cannot be classified correctly using color features, the SIFT recognition algorithm is used as additional evidence. The tagged classification results from the Tier-2 node together with a detection timestamp are transmitted to a sensor proxy, and the raw image data is locally stored on the flash memory. The Tier-2 camera node then goes back to sleep.

The user can query the system by specifying an object tag that needs to be located. An approximate time frame of interest can also be provided by the user to further refine the search. The proxy locates the most recent event corresponding to the requested object, and queries Tier-2 sensors that have reported the object. Since the Tier-2 sensor is asleep, this query is first received by the Tier-1 node which wakes up the Tier-2 sensor and forwards the query over the serial connection. The matching image ROOs are retrieved, and displayed on a GUI to the user, who can mark images to be ”Valid” or ”Invalid”. If the required object is not found, the proxy retrieves the previous event matching the query, and repeats the same procedure. This continues either until the object is located, or until no more objects were detected in the timeframe of interest. Periodically, the proxy also returns user feedback to the appropriate sensors, which use this information to refine their clusters, thereby enabling more efficient and more robust recognition.

3. Object detection

The object detection procedure in our system involves two steps. The first step is detecting the presence of a still object at each Tier-1 CyclopsCam. This procedure aims to filter out transient motion in the field of view of the camera such that only objects that stay relatively still are detected. The second step involves triggering the Tier-2 EnalabCam, and mapping from the ROO in the CyclopsCam to the EnalabCam. This procedure aims to address the fact that the EnalabCam is woken up infrequently and cannot maintain a reliable background model locally. Hence the EnalabCam needs to be told approximately where the detected object is located in its image.

3.1. Object detection in low power tier

For object detection, the Cyclops node maintains the background using an average update model, which is computational-efficient. The background image B_m is updated by integrating the new frame I_c into the current background with a first order recursive filter: $B_m = (1 - \alpha)B_m + I_c$. By computing the differences between corresponding pixels in I_c and B_m, a motion mask M_{mot} is obtained to identify the foreground part in the current frame. A connected components analysis is performed on M_{mot}, and a set of blobs exceeding a minimum size is extracted. These blobs represent potential objects added to the scene.

To detect that a candidate blob represents a new object placement event as opposed to a transient motion event, we need to check if this blob has been detected before and has been still for a sufficiently long time. To achieve this, we compare all the blobs in the current frame with those in the previous frame. If the size and center of a blob is similar enough to those of a blob in the previous frame, these two blobs are considered to be the same object. When the detection duration of an object blob becomes longer than a pre-defined threshold, it is classified as a still object. The object blob is then extracted and saved on the local flash memory of the MICAz mote.

3.2. Inter-tier triggering and ROO mapping

After the still objects are detected, the Tier-1 node needs to wake-up the Tier-2 camera from Deep-sleep mode. Since the wake-up of the iMote2 node from this state has long latency and consumes significant energy, the Tier-1 node triggers the Tier-2 node after a batch of still objects are detected. In this manner, the energy consumed to wakeup the Tier-2 camera is amortized across multiple detections. Note that the wakeup delay is not a problem for our application since we are trying to detect still objects that remain in the scene for a significant duration. Our system will not be as effective in a tracking scenario since the latency of wakeup needs to be low in order to track motion. Although the FOVs of two cameras are similar, there may be slight translation, rotation and considerable scaling between them due to installation bias and the nature of the mechanical mounts. To achieve robust region mapping from the pixels $\mathbf{P}_j = [u_x, u_y, 1]^T$ in CyclopsCam image to the pixels $\mathbf{P}^{'}_j = [v_x, v_y, 1]^T$ in EnalabCam image, a motion model is defined based on the affine transformation.

$$\mathbf{P}^{'}_j = \mathbf{D}_{x_0,y_0}\mathbf{S}_{s_x,s_y}\mathbf{R}_\theta\mathbf{P}_j$$

where $\mathbf{D}_{x_0,y_0}$ is the translation matrix, $\mathbf{S}_{s_x,s_y}$ is the scale matrix and $\mathbf{R}_\theta$, the rotation matrix. In order to solve the motion model, the dual-camera node executes a calibration procedure at system deployment time. Both cameras take an image simultaneously, and the Tier-1 node transfers its image over the serial port to the Tier-2 node. The Tier-2 node then extracts SIFT descriptors [18] from the two images. Since the SIFT descriptors are invariant to the image rotation, scale, and translation expected in this application, they provide a consistent set of local descriptors to match between the two images. This calibration procedure is typically more computationally intensive than object recognition since it needs to be performed on the entire image as opposed to just the ROO. However, this is a one-time computation, hence its overhead is a very small fraction of overall energy resources. After the inter-tier calibration is done, any ROO in CyclopsCam image can be mapped to the EnalabCam image efficiently using the motion model. Figure 3.b shows a mapping result, in which the bounding box around a book that is detected on the desk in the upper CyclopsCam image is mapped to the appropriate rectangle in the lower EnalabCam image.

4. Semi-supervised object recognition

Given the Region-of-Object extracted by the object detection module, the next task of the Tier-2 node is to accurately and efficiently recognize the object. Our approach includes three procedures: feature extraction, object recognition and constrained cluster update.

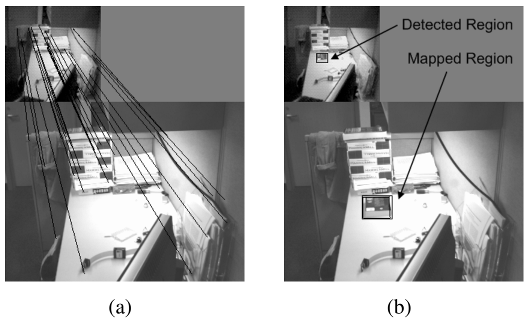


Figure 3. An example of Field-of-View and ROO mapping: (a) Control points found by SIFT. (b) The ROO mapping result.

4.1. Feature extraction

We use two kinds of features to classify objects. The first type of feature is a color histogram. Color histogram matching is one of the most widely used algorithms for detecting and recognizing objects in images [11]. Our system calculates a 32-bin hue histogram for each ROO image. The second type of feature that we use is the SIFT descriptor, which represents an image as a collection of local feature vectors that are invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine or 3D projection [18]. Given two images, a matching algorithm is performed to calculate the number of matching points between them, which represent the similarity.

Both color and SIFT features have their advantages and limitations. Color features can be calculated in a computationally inexpensive manner, and are invariant to severe scale, rotation and 3D projection; however, they are not invariant to illumination changes. SIFT tolerates illumination changes and is widely considered to be one of the best feature representation methods, but it is computationally intensive and does not perform well for deformable objects or object that has no consistent texture in appearance.

Our system uses a combination of the two features in a cascading multi-feature detection method. The idea is to use color features to filter out irrelevant images and to classify images that have distinctive color hues. The SIFT features are then used to recognize the remaining images that are hard to classify using only color. This combination enables us to tradeoff between efficiency and robustness, since the SIFT matching is performed only on a small set of un-classified but relevant images. We note that the idea of combining two kinds of image features for more robust detection has been considered in image processing literature [29, 32, 27], however, we exploit this technique for energy-efficiency as opposed to detection accuracy.

4.2. Object recognition procedure.

The object recognition procedure starts by gathering a small set of object images as training samples and making the initial clustering division with standard k-means algorithm. During this phase, clusters may be tagged by a user - for instance, one cluster might correspond to PDAs whereas another might correspond to cellphones. After this training phase, the system can be used to monitor the scene continually, and recognize the object detected using this initial cluster model. The object recognition procedure works in a batch manner, i.e., it is invoked after n objects are detected. We now describe the recognition approach on an object o_i. The pseudocode of the algorithm is shown in Algorithm 1.

4.2.1. Recognize by Color histogram. The recognition procedure first tries to classify a new object using color histogram clustering since this can be done efficiently. Since color histogram is a less precise feature, we use it in two ways: (a) to filter irrelevant images that are unlikely to be an object of interest and hence can be immediately discarded, and (b) to determine if an object can be recognized solely using the color histogram, in which case the SIFT descriptor based matching need not be performed. The procedure is shown in the first four steps of Algorithm 1.

The algorithm first finds the distance between the color histogram of the new object o_i and each cluster centroid, d_{ij}. In step 2, this distance is used to determine a normalized cluster membership metric, R_{ij}, which represents the likelihood that object o_i belongs to cluster j. If the minimum distance of o_i to all clusters is two times larger than the maximum of the distances between all samples to their cluster centroids, o_i will be considered to be an irrelevant object and will be discarded. Otherwise, in Step 4, the algorithm checks to see if the best matching cluster (i.e. the cluster with maximum membership metric) exceeds a pre-determined threshold, T_R. If so, o_i is considered to uniquely belong to the cluster. In this case, the algorithm matches the object to the tag associated with the cluster (e.g. cup, or cellphone), and terminates.

4.2.2. Combine the Color and SIFT features. While the color feature-based classification filters irrelevant images and identifies images that have clear membership in one cluster, there may be a number of images that are close to multiple clusters and cannot be classified accurately. We use the SIFT features to identify such cases. Classification using SIFT features involves three steps. First, a previously observed image that is closest to the centroid of each "nearby cluster" (the cluster j such that R_{ij} is larger than a threshold T_R^L) is chosen as the "representative" image for the cluster. Second, the SIFT descriptors for the new image are compared to the representative images for each cluster, and a SIFT cluster membership score is assigned based on the similarity. Finally, a combined score is assigned to the new object based on a weighted combination of the color-based and the SIFT-based membership metric. The object is considered to belong to all clusters, and assigned all tags for which the weighted score is greater than a pre-defined threshold.

```
**Algorithm 1: Pseudocode of object recognition procedure.**
Input: object $o_i$
Current model: Sample set $\mathcal{X}$ and the $k$ clusters $\{\mathcal{X}_h\}_{h=1}^k$
Parameters: $T_R$, $T_R^L$, $\alpha$, $TH_{tag}$
Method:
1. Calc histogram $x_i$ for $o_i$.
2. Calc membership of $o_i$ to all cluster centroids $\mu_j$:
   $R_{ij} = 1/(d_{ij}^2 \sum_{j=1}^k (1/d_{ij}^2))$, where $d_{ij} = \|x_i - \mu_j\|$.
3. If $\min(d_{ij}) > 2\max(d_{lj})_{l=1,j=1}^{l=n,j=k}$, $o_i \ni \{\mathcal{X}_h\}_{h=1}^k$, exit.
4. If $\min(R_{ij}) > T_R$, $similarity_{ij} = R_{ij}$, goto step 6.
5. (1) For each $\mu_r$ such that $R_{ir} > T_R^L$, do:
           Calc/save SIFT descriptors for $o_i$ and the closest sample $s_l$ to $\mu_r$; Calc number of SIFT matching points $M_{ir}$ between $o_i$ and $s_l$.
   (2) For the rest $\mu_r$ such that $R_{ir} < T_R^L$, let $M_{ir} = 0$.
   (3) Calc $R_{SIFT}^{ir} = M_{ir}/\sum_{j=1}^k M_{ij}$ for all $\mu_r$.
   (4) $l = \max_r(R_{SIFT}^{ir} = M_{ir})$. $o_i, s_l \to \mathbb{S}$.
   (5) $similarity_{ij} = R_{ij} + \alpha R_{SIFT}^{ij}$.
6. Let $k = \max_j(similarity_{ij})$, $o_i \in \mathcal{X}_k$.
7. For all $j$ that $similarity_{ij} > TH_{tag}$,
   $stringof(\mu_i) \to Tags$.
```

This procedure is shown in Steps 5-7 of Algorithm 1. In step 5, we calculate the number of SIFT matching points between o_i and the representative image s_j corresponding to each cluster. For cluster j, s_j is the closest sample to the centroid μ_j. We can then calculate the number of SIFT matching points M_{ij} between o_i and s_j in each cluster. R_{sift}^{ij} represents the SIFT similarity between o_i and the sample s_j in cluster j. R_{sift}^{ij} represents the evaluation score of the membership of o_i to cluster j given by SIFT features.

The overall evaluation score of the identity of o_i is calculated by combining color and SIFT features: $similarity_{ij} = R_{ij} + \alpha R_{sift}^{ij}$, where α is a weight that reflects the importance of SIFT features. If $similarity_{ij} > TH_{tag}$, then the object is associated with a specific tag, where TH_{tag} is a predefined threshold that balances the false positive and false negative of object recognition. After an object is recognized, the ROO and the entire scene image is stored in the flash memory of Tier-2 node, and metadata about the recognized object (node id, timestamp, tags) is transmitted to the proxy.

4.3. Constrained cluster updating

Until now, we have discussed how the color-based clusters can be used for classification. We will now describe how the clusters themselves can be evolved in a dynamic manner, by taking into account both information gleaned from the SIFT feature-based matching as well as from user feedback. SIFT feature-based matching provides links between two ROOs. For instance, if the matching score R^{ij}_{sift} of two samples i, j is high, it is likely that sample i and j are the same object. In addition, since our object retrieval application interacts with users, we can even get more constraints from user feedback. For example, a user can directly label the class of a sample, or denote if two samples are the same object. We assume in this work that user feedback is always accurate.

4.3.1 Constraint definition. We define two broad classes of constraints - hard-label constraints and pair-link-constraints. The former captures constraints provided by user feedback, where the user labels an image ROO, for instance, as a PDA. The latter captures constraints between pairs of images - for instance, based on SIFT image matching. We now formally define these constraint classes.

Hard-labeled-constraint (HLC) indicates a definite match between a sample and a certain cluster. $\mathbb{H}$ denotes the set containing HLC.

Pair-link-constraint (PLC) represents the constraint between pairs of examples. There are three subclasses:

(i) A Must-link constraint indicates that two samples should belong to the same cluster. *(ii) A Cannot-link* constraint indicates that two samples must belong in different clusters. *(iii) A Soft-link* constraint indicates two samples are probably in one cluster. The evidence of each soft-link constraint is computed by SIFT descriptor matching, and its weight is assigned based on the SIFT matching score R^{ij}_{sift}.

For Pair-link-constraint, we define $\mathbb{S}$, $\mathbb{M}$, $\mathbb{C}$ as the sets containing the soft-link, must-link and cannot-link constraints.

4.3.2. Constrained k-means. Although clustering algorithms like k-means have the ability to handle hard-label constraints, it is difficult for them to handle pair-link constraints. A few techniques have been suggested in semi-supervised k-means algorithms to address this problem [33, 8, 12], upon which our approach is based.

Since standard k-means cannot handle pairwise constraints explicitly, the goal of clustering is formulated as minimizing a combined objective function which is the sum of the total distance between the samples and their cluster centroids and the cost of violating the pair-link constraints. The clustering problem can be formulated as minimizing the following objective function, where x_i is assigned to the partition $\mathcal{X}_i$ with centroid μ_{l_i}.

$$\Phi = \beta \sum_{x_i \in \mathcal{X}} \|x_i - \mu_{l_i}\|^2 + \sum_{(x_i,x_j)\in\mathbb{S}} w_S^{ij} \perp[l_i \neq l_j] + \sum_{(x_i,x_j)\in\mathbb{M}} w_M \perp[l_i \neq l_j] + \sum_{(x_i,x_j)\in\mathbb{C}} w_C \perp[l_i = l_j]$$

in which $\perp$ is the indicator function, with $\perp[true] = 1$ and $\perp[false] = 0$. β is a parameter to trade off the importance of the data set itself with that of the constraints. The cost of violating a pair link constraint is given by the weight of this link. w_S^{ij} denotes the weight of the soft-link constraints based on SIFT matching, and w_M and w_C denote the weights on must-link and cannot-link constraints. Since explicit user feedback is more precise than the result that SIFT-based matching returns, we use higher value for w_M and w_C than that for w_s^{ij}.

Algorithm 2: Constrained cluster updating algorithm.

```
Input: A set of old samples $\mathcal{X} = \{x_i\}_{i=1}^n$
       The old clusters: disjoint $k$ partitioning $\{\mathcal{X}_h\}_{h=1}^k$
       A set of new samples: $\mathcal{X}_{new} = \{x_i\}_{i=1}^m$
       Constraint sets: $\mathbb{S}$, $\mathbb{M}$, $\mathbb{C}$, $\mathbb{H}$
Parameters: $\beta$, $w_M$, $w_C$, $w_s$
Method:
1. Load the cluster configuration $\{\mathcal{X}_h\}_{h=1}^k$
2. Repeat until convergence:
   (1) Assign all sample with HLC: For the sample $(x_i \to j) \in \mathbb{H}$, directly assign $x_i$ to cluster $h_j$. For the sample $(x_i \nrightarrow j) \in \mathbb{H}$, assign it to the closest cluster $h$ such that $h \neq j$.
   (2) Assign each other sample $x_i$ to the cluster $h_L$, for
       $h_L = \arg\min_h (\beta \|x_i - \mu_h^{(t)}\|^2 + \sum_{(x_i,x_j)\in\mathbb{S}} R^{ij}_{SIFT} \perp[h \neq l_j] + \sum_{(x_i,x_j)\in\mathbb{M}} w_M \perp[h \neq l_j] + \sum_{(x_i,x_j)\in\mathbb{C}} w_C \perp[h = l_j])$
   (3) Estimate and update means:
       $\{\mu_h^{(t+1)}\}_{h=1}^k \leftarrow \{\frac{1}{|\mathcal{X}_h^{(t+1)}|} \sum_{x\in\mathcal{X}_h^{(t+1)}} w_s x\}_{h=1}^k$
   (4) $t \leftarrow (t+1)$
3. Delete a set of the oldest samples from the clustered data.
```

Algorithm 2 shows the cluster update algorithm. The algorithm alternates between the cluster assignment and centroid estimation steps. In the cluster assignment step, every sample x_i is assigned to a cluster such that it minimizes the sum of the distance of x_i to the cluster centroid and the cost of constraint violations incurred by that cluster assignment. The centroid re-estimation step is the same as standard k-means.

The proof of the convergence property of our algorithm is similar to the proof in [8]. In our algorithm, the pairwise constraints are given only by SIFT features and user feedback, which are not explicit functions of the centroid, so in re-estimating the cluster centroid μ_h, only the component $\sum_{h=1}^{k} \sum_{x_i \in \mathcal{X}_h} \|x_i - \mu_h\|^2$ is minimized. Hence the objective function decreases after every cluster assignment and centroid re-estimation step till convergence, implying that the algorithm will converge to a local minimum of Φ. We give samples with hard constraints more weight in the centroid re-estimation step.

The computational complexity of k-means is O_{ndk}, where n, d, k represent the number of data points, dimensions and clusters respectively. It is linear in the size of the input, which makes the algorithm efficient.

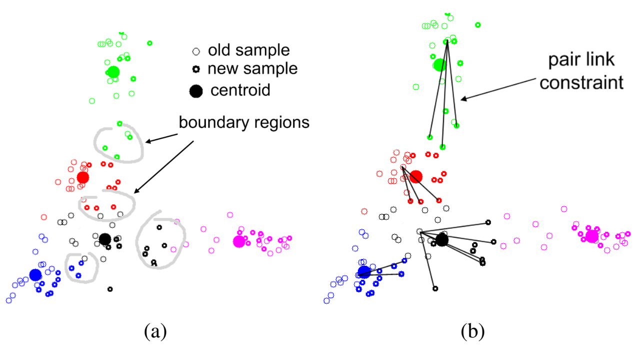


Figure 4. An example of clustering under constraints. Color is used to identify different clusters. To represent clusters, Two dominating dimensions are calculated by PCA and used as x, y coordinates.

An illustration of constrained cluster updating is shown in Figure 4, in which each color represents an actual cluster. Due to the slight illumination change, there is a little shift between new samples and old samples in each cluster. As seen from Figure 4 (a), the new samples in the "boundary regions" between clusters may be incorrectly assigned due to the shift. Figure 4 (b) shows that by using pair-link constraints to update clusters, the centroids shift towards the new samples so that the cluster model represents the new samples better. In this way, our system is more reactive to changes in illumination. The cluster update algorithm is performed infrequently at the Tier-2 node (iMote2) when a sufficiently large number of constraints have been accumulated, hence the computational overhead of the approach is not significant.

5. Object retrieval in proxy-tier

Our system employs a proxy-tier to organize the information from the dual-camera sensors, as well as process user queries.

5.1. Event Database

The proxy-tier maintains a database of event messages sent by sensor nodes. Each time a sensor node detects and recognize an object, it will send an event notification to the proxy. In the event that multiple overlapping cameras are placed to cover an area of interest, it is possible that multiple nodes can detect and recognize the same object, thereby suppressing false negatives. To merge the recognition results from multiple nodes, the proxy combines event messages with similar timestamps, and stores it in a local database for future retrieval. Note that consistent timestamps can be obtained by using a network-wide time synchronization protocol such as FTSP [20]. Table 1 shows an example of the stored item in database in the proxy-tier, where "Global ID" represent the global sequence number of the event, "Node-Addr (Local-ID)" indicates the address of the nodes detect this event attached with the local event index in the node. "Tags" is an intersection of the recognition results of these nodes that see the same object. "Timestamp" is the average time of the same event detected by multiple sensors.

Table 1. Database at the proxy-tier

Global ID	Node-Addr (Local-ID)	Tags	Timestamp
n	2 (14)	Key; PDA	2007-10-3-22:25:50
n+1	2 (15)	Book	2007-10-3-22:31:27
n+2	2 (16); 3 (15)	Book; Cup; PDA	2007-10-3-22:38:54

5.2. Tag-based object retrieval

Our system provides a tag-based object retrieval capability. The user can provide the name of a tag or class as the command to retrieve the latest location of this object. The retrieval process is performed in an interactive manner. The proxy first searches for the query string (object name) in the *Tags* field of local database. The proxy locates the latest item whose *Tags* field contains the query string, and sends an "ROO Request" message to the appropriate sensors in NodeID field to retrieve the ROOs of interest. Each node that receives this request compresses the candidate ROO image in JPEG format and transmits it over the wireless radio to the proxy.

The candidate ROO sent back by the sensor node will be shown to the user using an easy-to-use GUI. If the user marks this ROO as ”Valid”, i.e., confirms that it is, in fact, the queried object, the proxy sends an ”Image Request” command to the sensor node and a full image containing the ROO will be transmitted back to the proxy and shown to the user. Otherwise if the user marks this ROO as ”Invalid”, it means the ROO is not the queried object due to a false positive. The proxy will continue to search through its database to locate an older item that matches the user query. This process is repeated until an ROO is accepted by the user. Such an interactive retrieval approach ensures that we don't transfer an entire scene image unless we are sure that it contains the queried object, thereby saving energy.

The user feedback also provides constraints that can be exploited for better clustering, as described in Section 4.3.2. In addition to the ”Valid/Invalid” marking, users also have the option of correctly labeling a candidate ROO, or indicating if two candidate ROOs are the same object or not. This information is periodically fed back from the proxy to the appropriate sensor nodes, which use them to update the cluster model.

6. System implementation

This section describes the implementation details of our system based on the design discussed in previous sections.

6.1. Hardware implementation

Tier-1: Tier-1 comprises of a Cyclops camera [28] connected to a MICAz [1] mote. The Cyclops comprises of an Agilent ADCM-1700 CMOS camera module, a Xilinx FPGA and an ATMega128 microcontroller. The Cyclops communicates with MICAz via I2C bus and uses a 2.4GHz CC2420 radio chip as the wireless component.

Tier-2: Tier-2 node comprises of an Enalab camera [5] and an iMote2 [3]. Enalab camera module comprises an OV7649 Omnivision CMOS camera chip, which provides color VGA (640x480) resolution. The iMote2 comprises an 18-400MHz Xscale PXA271 processor and a CC2420 radio chip. The Enalab camera is connected to the Quick Capture Interface (CIF) on iMote2. To support large image data storage, a 1GB external flash memory is attached.

The Tier-1 node and Tier-2 node are connected with a trigger circuit for wakeup, and communicate through the serial port. The Cyclops camera and the Enalab camera are mounted close to each other in order to increase the accuracy in inter-tier ROO mapping. The sensor nodes are powered by batteries.

Base-station: In the prototype system a PC is used as the base station. An iMote2 node is connected to the PC and acts as the gateway.

6.2. Software environment

The software environments in our system are different on the different tiers. In Tier-1, both MICAz and Cyclops run TinyOS 1.1.14. We enhanced the object detection software available for the Cyclops to perform still object detection. The Tier-2 iMote2 runs Arm-Linux. OpenCV library [4] is used on the iMote2 to facilitate the basic image computations, such as image conversion, transformation and color histogram computation. Our SIFT algorithm is based on the SIFT++ lib, which is a lightweight C++ implementation of SIFT descriptor extraction. The Intel Integrated Performance Primitives library (IPP) is used to accelerate data processing. A JPEG compressor was also developed using IPP lib to compress images. The IEEE 802.15.4 radio protocol is used to communicate among all nodes in the system.

7. Experimental evaluation

We evaluate the performance of our object recognition system through an extensive set of experiments. We first evaluate the benefits of using a dual-camera sensor node instead of a single camera node. We then evaluate the power consumption and performance of the object detection, and object recognition algorithms individually, and finally provide a full system evaluation using multiple cameras in a realistic environment.

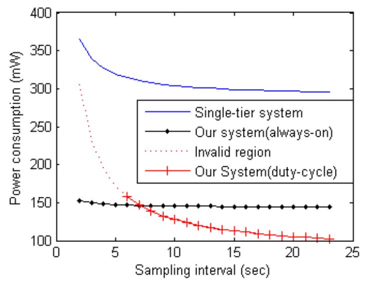
There are a number of key parameters in our system: T_R, T_R^L, TH_{tag} for Algorithm 1, and w_M, w_C, w_s for Algorithm 2. In all our experiments, T_R, T_R^L were fixed and set to 0.7 and 0.2 respectively. Our intuition for picking these values was that they screen out the samples in the ”boundary regions” (as seen in Figure 4.a). TH_{tag} is set to 0.3 in all experiments except in Section 7.5 where we evaluate the impact of tuning this parameter. For algorithm 2, the experimental results are not very sensitive to the parameters w_M and w_C, as long as they are assigned a value larger than 10. In our experiment we set $w_M = w_C = 10$. w_s is also not a sensitive parameter and is set to 2.

7.1. Energy cost of object detection

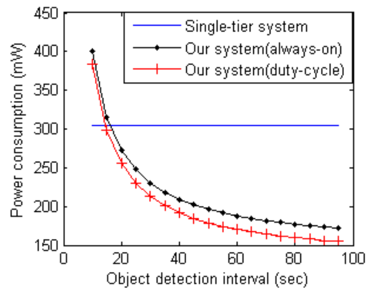
In order to provide a better intuition for the energy gains offered by our system, we compare our system with a single-tier design that keeps iMote2+EnalabCam node always on to perform the detection. In our system, the MICAz+CyclopsCam will wake up the iMote2+EnalabCam every 4 still objects are detected. We also present the power consumption for two operational modes of the CyclopsCam - a ”duty-cycle” mode and an ”always on” mode.

Figure 5 (a) shows the power consumption for continuous monitoring as a function of the sampling interval. In this experiment, no object is detected, i.e., iMote2 does not need to be woken up. As seen from the figure, both operation modes of our system consume less energy than the single-tier version, clearly demonstrating the benefits of using a tiered system for object detection. The experiment also reveals that the critical point for choosing between the two modes of the CyclopsCam is around 6 seconds. Thus, when the sampling interval is less than 6 seconds, the always-on mode of the CyclopsCam is more efficient since the energy consumption for transitioning the camera from sleep to wake state dominates the total power consumption.

Figure 5 (b) evaluates the effect of object detection interval (i.e. the average time between two consecutive object detection events) on the power consumption of the three schemes. In this experiment, the sampling interval is fixed to 10 seconds. As shown in the figure, if the still object is detected very frequently, the power consumption of our system may be a little larger than that of the single-tier version, because of the frequent wake-up overhead of the iMote2 node. For most reasonable inter-object intervals, the power consumption of the two versions of our system is considerably less than that of the single-tier version.



(a)



(b)

Figure 5. Power consumption analysis. (a) Effect of the sampling interval. (b) Effect of object detection interval.

7.2. Accuracy of object mapping

Obtaining an accurate mapping between the CyclopsCam's ROO and the EnalabCam's ROO is essential to the performance of our system. We compare the error in ROO estimation for two schemes: (a) an always-on single-tier system that uses the iMote2 and EnalabCam, and (b) our system using inter-tier wakeup and ROO mapping. The errors are calculated by comparing the object region produced by the algorithms to those labeled manually. The experimental results in Table 2 show that our system has only marginal higher error (less than two pixels along each axis) than a single-tier system that uses a high resolution camera. In addition, the absolute error is less than three pixels on each axis.

Table 2. Error analysis on object detection. Errors are measured in pixel.

	Center Error	Width error	Height Error
Test	Avg (Var)	Avg (Var)	Avg (Var)
EnalabCam	1.04 (0.88)	1.70 (4.90)	1.30 (4.46)
DualCam	2.47 (2.08)	2.84 (4.13)	2.91 (6.07)

7.3. Comparison between color and SIFT

In this section, we demonstrate the accuracy and energy benefits of using the Color and SIFT features in a cascading manner for object recognition (described in Section 4.2.1). Table 3 compares the performance of three different recognition methods, only Color features; only SIFT features, and the cascading combination of the two. We use two metrics to evaluate the schemes - rate and latency. The ”rate” metric shows the percentage of correctly recognized objects, and the ”latency” metric shows the amount of time taken by the iMote2 node for object recognition, which in turn corresponds to the energy consumption for recognition. Five common objects are used in this test: book, cup, keyring, PDA, and TV remote control. These are all common objects that are easy to lose.

We tested the three methods with a training set and a test set. The training set contains 50 samples. The test set, FIXED-ILLUM, contains 100 samples under the same lighting conditions as in the training set. The set VARY-ILLUM contains 100 samples where we introduced illumination changes by switching off one of the three ceiling lights in the room. Samples are collected by placing the objects in different locations and with different poses on a table. We first train the clusters by using the training dataset, and then classify the two test sets respectively.

Table 3 shows that the recognition rate by using a combination of Color and SIFT features has higher accuracy than using the features individually. When illumination changes are present, the improvement in accuracy over color alone is 17%. The computation time for processing is significantly greater than for a method that just uses color features, but is only a third of the time required for processing the higher accuracy SIFT descriptors. This is because our algorithm needs to run the SIFT algorithm only on roughly 20-40% of the samples. Note that cluster updates were not performed in this experiment.

Table 3. Comparison on recognition methods (The power consumption during object recognition is 681mW)

Methods	FIXED-ILLUM		VARY-ILLUM	
	Rate	Latency	Rate	Latency
1. Color	84%	0.0034s	63%	0.0032s
2. SIFT	83%	6.49s	75%	6.17s
3. Color + SIFT	91%	2.07s	80%	2.96s

7.4. Benefits of using constraints

We now evaluate the benefits of using pair-link constraints to improve clustering results. In this experiment, we collect another test dataset, VARY-ILLUM-1, that contains 100 samples under the same illumination conditions as in VARY-ILLUM. We then evaluate whether the recognition rate on VARY-ILLUM-1 improves as a result of refinement of the clusters with constraints that are obtained from VARY-ILLUM in the previous experiment. Table 4 shows the results of this experiment. The ”No constraint” column shows the results using only standard k-means without constraints, and the ”Using constraint” column shows the results of our constrained k-means algorithm with refinement using constraints derived from VARY-ILLUM (described in Section 4.2.2). As seen from Table 4, the use of constraints improves the recognition results by 10% when only color features are used, and by 6% when both color and SIFT features are used. The refinement of clusters also improves the latency required to perform the object recognition by about 20%, since the clusters are more accurate and hence the SIFT recognition algorithm is invoked fewer times.

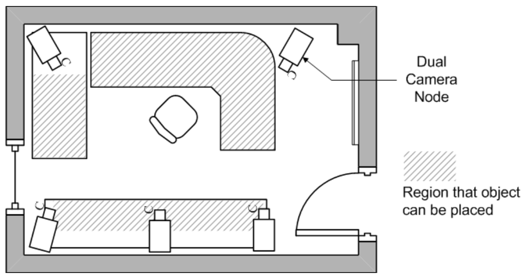
Table 4. Improvements on using constraints (The power consumption during object recognition is 681mW)

Recognition methods		VARY-ILLUM-1	
		No constraint	Using constraint
Color	Rate	68%	78%
Color + SIFT	Rate	77%	83%
	Latency	2.89s	2.24s

The recognition results above are all produced by a single camera node. We also evaluate the benefits of placing multiple sensor nodes with overlapping coverage for object recognition. In our experiment we evaluated the recognition rate using two camera views and found that the recognition rate improves from 82% (single view) to 86% (two views).

7.5. System performance on object retrieval

We now evaluate the overall performance of our object retrieval system using an experiment in a real room environment with multiple sensor nodes. In this experiment, we placed 5 dual-camera nodes in a room so that the FOVs of these nodes cover most of the area in which human activity may happen. The cameras are placed in an ad-hoc manner, so some cameras have overlaps in their field of view. Figure 6 shows the deployment of the camera network. We use the same object set as previous experiments: book, cup, key ring, PDA, and TV remote control. In this experiment, objects are randomly placed and removed from the monitored area. Queries for each object are generated after roughly every 20-25 object placements events.



(a)

Figure 6. Camera network deployment, a top view map.

Table 5 gives the results from this experiment. The first column labeled ”Correct/Total” stands for the ratio of number of correctly retrieved images to the number of queries. A correct retrieval is the case where the system returns the latest scene image containing the queried object. ”Average ROO Transmitted” denotes the average number of candidate regions that need to be transmitted to get the retrieval result. As seen from the table, in the absence of user feedback, our system achieves 90% accuracy in ROO retrieval, with less than four candidate regions retrieved per correct retrieval. If users provide additional feedback by labeling returned candidate ROOs, the accuracy increases by 5% to 95%, and the average number of ROOs need to be transmitted for each query is reduced from 3.6 to 2.5 (a reduction of 7KB in bytes transmitted).

Impact of Tagging Threshold: The correct rate and the

Table 5. Object retrieval performance (The tagging threshold TH_{tag} is fixed to 0.3)

	Correct/Total	Average ROO (data bytes) Transmitted
No user feedback	18/20	3.6 (22.5KB)
With user feedback	19/20	2.5 (15.6KB)

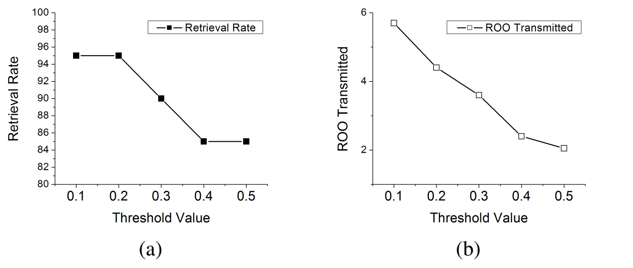


Figure 7. Effect of the tagging threshold TH_{tag}. (a) Retrieval rate. (b) ROO transmitted.

number of ROOs that are transmitted are sensitive to the value of threshold TH_{tag}. As described in Section 4.2.2, TH_{tag} influences the number of false positives and number of false negatives in object recognition, and also determines the number of category tags saved for each ROO. Figure 7 illustrates the effect of changing TH_{tag} and shows that the threshold provides a tradeoff between the number of ROOs transmitted against the accuracy of retrieval. If available energy in the system is limited, a higher value can be used for TH_{tag}, which will reduce the number of ROO images transmitted, but will also decrease the correct rate of retrieval. The experiment result shows that our system functions well in real world setting with multiple camera nodes, and can be tuned to tradeoff recognition accuracy for amount of energy expended for communication.

8. Related work

In this section, we review related work that we have not discussed earlier in this paper.

Memory assistant tool for object finding: The design of a memory assistant tool for object locating and finding has received considerable attention in recent years. The work in [26] investigates the real-world nature of what losing an object means and general strategies that can be used to find those objects. Other work utilizes visual information to track and locate objects, for instance [24, 34] use PTZ cameras to detect and recognize the indoor objects. Our work differs from all these in that we explore how to use low-power camera sensors distributed in a home environment to facilitate retrieval of misplaced objects.

Multi-tier sensor network: The multi-tier structure for wireless sensor network has also been considered in prior work. Tenet [13] argues for a multi-tier design and SensEye [16] proposes a three-tier camera sensor network for surveillance. Our work uses two tiers, but they are tightly coupled as part of a single platform. In addition, we propose novel techniques for splitting an object recognition task between a low-power and high-power camera.

Object recognition: Many different approaches to object recognition have been proposed in computer vision, including model-based and appearance-based approaches [30]. In recent years, methods using local appearance features [18, 22] have come more popular. In [32] the SIFT descriptor is combined with color histograms. The work in [27] discusses fusion methods for SIFT and LUV color moments descriptors. In our work we exploit the combination method of SIFT and color features for energy-efficiency as opposed to recognition accuracy.

Semi-supervised clustering: Clustering is traditionally viewed as an unsupervised method for data analysis. Based on the widely used k-means algorithm, some constrained versions have been developed [33, 8, 12] to incorporate the information about the problem domain that is available in addition to the data instances themselves. As an extension of the model proposed in [8], our work incorporates hard and soft constraints together.

9. Discussion and future work

As a prototype system, our system performs well with single user in the room moving the objects around slowly. While our study demonstrates that the system is efficient and useful, there are still some limitations need to discuss.

First, our system is tolerable to slight-to-moderate light changes. With more severe light changes, our system needs to perform the expensive SIFT algorithm frequently or even can not achieve reasonable object retrieval result. In our application, the light condition can be satisfied at most time since the system is in-door use only. Second, the performance of our system is affected by the visual appearance of objects. The system may fail to recognize if two objects have both similar color and undistinguishable SIFT descriptors. The performance may also degrade with very small object, since our camera sensors can not zoom. In addition, our system can not discover the object if it is severely occluded, or goes out of the FOV of the cameras.

Some future works can be done to alleviate these limitations. We plan to explore statistical models such as EM [10] to estimate the distribution of target objects. We also plan to add one more tier that comprises PTZ cameras to enhance the capability of object recognition. Finally, we will combine other sensors such as RFID tags and acoustic sensors to improve the performance of the object finding system. Our system will be deployed in the senior center at Amherst MA as a part of ASSIST project [34].

10. Conclusion

This paper presents the design and implementation of an indoor object retrieval system using a network of dual-camera wireless nodes, each of which combine multiple cameras with complementary capabilities. Our system proposes a number of novel techniques including: (a) the use of the low-power camera both for still object detection and region-of-object estimation, (b) the use of two different visual features - color histogram and SIFT descriptors - for energy-efficient yet accurate object recognition, and (c) refinement of clusters for more accurate classification using pairwise constraints from SIFT matching and user feedback. Our experimental results demonstrate that the system is energy efficient, computationally efficient, and accurate.

References

[1] Crossbow wireless sensor platform. http://www.xbow.com/.

[2] Enalab camera. http://enaweb.eng.yale.edu/drupal/.

[3] Intel mote2 platform. http://www.xbow.com/products.

[4] Intel open source computer vision library. http://www.intel.com/technology/computing/opencv/.

[5] Cyclops platform. http://enaweb.eng.yale.edu/drupal/. 2007.

[6] H. Aghajan, J. Augusto, C. Wu, P. McCullagh, and J. Walkden. Distributed vision-based accident management for assisted living. *Int. Conf. on Smart homes and health Telematics (ICOST)*, 2007.

[7] E. Ardizzone, M. Cascia, G. Re, and M. Ortolani. An integrated architecture for surveillance and monitoring in an archaeological site. *Proc. of VSSN*, 2005.

[8] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. *Proc. of the 4th SIAM Intl. Conf. on Data Mining*, 2004.

[9] G. Borriello, W. Brunette, M. Hall, C. Hartung, and C. Tangney. Reminding about tagged objects using passive rfids. *Proc. of the 8th Ubicomp Conference*, 2006.

[10] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 1977.

[11] B. Funt and G. Finlayson. Color constant color indexing. *IEEE Transactions on PAMI*, May 1995.

[12] J. Gao, P.-N. Tan, and H. Cheng. Semi-supervised clustering with partial background information. *Proc. of the 6th SIAM International Conference on Data Mining*, 2006.

[13] O. Gnawali, B. Greenstein, K. Jang, A. Joki, J. Paek, M. Vieira, D. Estrin, R. Govindan, and E. Kohler. The tenet architecture for tiered sensor networks. *SENSYS*, 2006.

[14] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan. Mesheye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance. *IPSN-SPOTS*, 2007.

[15] J. Kientz, S. Patel, A. Tyebkhan, B. Gane, J. Wiley, and G. Abowd. Where's my stuff?: design and evaluation of a mobile system for locating lost items for the visually impaired. *ACM conf. on Computers and accessibility*, 2006.

[16] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu. Senseye: A multi-tier camera sensor network. *ACM Multimedia*, 2005.

[17] X. Liu, M. Corner, and P. Shenoy. Ferret: Rfid localization for pervasive multimedia. *Ubicomp Conf*, 2006.

[18] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 2004.

[19] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. *International Workshop on Wearable and Implantable Body Sensor Networks*, 2004.

[20] M. Marti, B. Kusy, G. Simon, and A. Ldeczi. The flooding time synchronization protocol. *ACM SENSYS*, 2004.

[21] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy. Ultra-low power data storage for sensor networks. *IPSN-SPOTS*, 2006.

[22] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on PAMI*, 2005.

[23] T. Nakada, H. Kanai, and S. Kunifuji. A support system for finding lost objects using spotlight. *MobileHCI*, 2005.

[24] R. Nelson and I. Green. Tracking objects using recognition. *International Conference on Pattern Recognition*, 2002.

[25] R. Orr, R. Raymond, J. Berman, and A. Seay. A system for finding frequently lost objects in the home. *Georgia Tech GVU Technical Report: GIT-GVU-99-24*, 1999.

[26] R. Peters, R. Pak, G. Abowd, A. Fisk, and W. Rogers. Finding lost objects: Informing the design of ubiquitous computing services for the home. *Georgia Tech GVU Technical Report: GIT-GVU-04-01*, 2004.

[27] P. Quelhas and J.-M. Odobez. Natural scene image modeling using color and texture visterims. *Proc. of CIVR*, 2006.

[28] M. Rahimi, R. Baer, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: In situ image sensing and interpretation in wireless sensor networks. *Proc. of ACM SENSYS*, 2005.

[29] P. Schgerl, R. Sorschag, W. Bailer, and G. Thallinger. Object re-detection using sift and mpeg-7 color descriptors. *MCAM*, 2007.

[30] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Transactions on PAMI*, 1997.

[31] T. Teixeira, D. Lymberopoulos, E. Culurciello, Y. Aloimonos, and A. Savvides. A lightweight camera sensor network operating on symbolic information. *Proceedings of 1st Workshop on Distributed Smart Cameras*, 2006.

[32] J. van de Weijer and C. Schmid. Coloring local feature extraction. *Proc. of ECCV*, 2006.

[33] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. *International Conference on Machine Learning*, 2001.

[34] A. Williams, D. Xie, S. Ou, R. Grupen, A. Hanson, and E. Riseman. Distributed smart cameras for aging in place. *Proc. of 1st Workshop on Distributed Smart Cameras*, 2006.

[35] P. Wilson, D. Prashanth, and H. Aghajan. Utilizing rfid signaling scheme for localization of stationary objects and speed estimation of mobile objects. *IEEE International Conference on RFID*, March 2007.