TelosCAM: Identifying Burglar Through Networked Sensor-Camera Mates with Privacy Protection

ShaoJie Tang, Xiang-Yang Li, Haitao Zhang, JianKang Han, GuoJun Dai, XingFa Shen

ABSTRACT

We present **TelosCam**, a networking system that integrates wireless module nodes (such as TelosB nodes) with existing legacy surveillance cameras to provide storage-efficient and privacy-aware services of accurate, realtime tracking and identifying of the burglar who stole the property. In our system, a property owner will have a wireless module node (called *secondary module*) for each of the properties that s/he wants to protect. The secondary wireless module node (which may be equipped with a motion detector) will not store any personal information about the owner, nor any specific information about the property to be protected. Each user of the system will also have a unique wireless module node (called *primary module*) that contains some security information about the owner, thus should be privately held by the user and be kept to the user always. The primary wireless module node will periodically send the heart-beat information to the secondary wireless module node. Once a secondary wireless module cannot detect the existence of a primary wireless module within its vicinity and it detects its own movement, it will start sending out the alarm signal periodically. The alarm signal will be captured by some *reading wireless module nodes*, integrated with existing surveillance cameras. Using the trajectory information provided by the secondary wireless module node, and the images captured by the surveillance cameras, our system will then automatically pinpoint the target burglar (e.g., a person or a car) that carries the stolen property. Our extensive evaluation of the system shows that we can find the burglars with almost 100% accuracy, while significantly reduce the storage-requirement of the legacy video surveillance system. It also can help the police to catch the burglars more efficiently by providing critical images containing the burglars.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; B.0 [**Hardware**]: General, Input/Output and Data Communications; B.4 [**Hardware**]: Input/Output and Data Communications; J.4 [**Computer Applications**]: Computer in Other Systems

General Terms

Design, Experimentation, Performance, Measurement

Keywords

Object tracking, object detection, sensor networks, surveillance camera.

1. INTRODUCTION

The national rate of unrecovered vehicles is at its highest point in more that 20 years. In fact, 43 percent of vehicles stolen in 2008 (latest FBI data) were never recovered, amounting in 411,444 stolen vehicles not returned to their rightful owners. More than 56,000 bikes were stolen in 2009 - that's more than 370 million in losses. Since 2006, there have been more than 253,000 motorcycle thefts. More than 2 million laptops are reported to be stolen every year, meaning that you have about a 10% chance of becoming a victim of laptop theft. It's estimated that 10 billion to 30 billion in merchandize is stolen from cargo ships, ports, highways, railroads and freight yards each year. According to the FBI Uniform Crime Report [2], an estimated $17.2 billion in losses resulted from property crimes in the U.S. in 2008. Of this total, burglary accounted for an estimated 22.7%.

Many different systems have been proposed and used in practice to enhance the security and protection of the property. The main approaches are to use surveillance cameras, motion detectors, and/or attach a unique RFID tag to the property to be protected. For example, traditional home security systems hope to deter or detect burglar by using increased surveillance (cameras, motion detectors, and alarm systems). However, these systems cannot help track or recover the property once it is stolen. Another commonly currently used approach is to install security surveillance camera at public place, mainly at traffic crossroad or inside buildings. These surveillance cameras can only record the objects (e.g., a car, or a person) that appeared inside the view of the camera, however it cannot detect whether an object does carry a stolen property, which may not be visible from the surveillance camera. For recovering stolen vehicles, *LoJack* may be the most common system for this purpose. It installs a small device hidden inside the vehicle, which will transmit homing beacons after being activated

when the vehicle has been reported to be stolen. Once a car with LoJack device is reported to be stolen, a number of high power wireless transmitters installed by LoJack will activate the LoJack device in the car by sending activation signal. Once the device is activated, it will then transmit periodic beacons that can be received by the LoJack receivers which is used by police. This approach makes it unsuitable for protecting smaller property (such as household asserts that are especially vulnerable for burglary), nor for long-term battery-powered operations. It also does not provide a privacy of the owner since the service provider can activate the device inside the vehicle. Asset tracking products like Brickhouse [1] and Liveview [4] use GPS to obtain realtime location information of the protected property and use cellular infrastructure to communicate this data to the control center, thus recover the trajectory of the property. Their high power draw requires recharging the device approximately every five days, making them unsuitable for use in tracking properties for long-term. We point out that knowing the trajectory of a property does not mean that we can recover the property easily. On the other hand, knowing the trajectory of a legitimate owner breaches the privacy of the user. Then a debacle here is how to efficiently pinpoint the burglar who carries a stolen property and protect the privacy of the legitimate owners at the same time.

In this paper, we present *TelosCam*, a system to track and identify personal property theft, improve historically dismal stolen property recovery rates, and disrupt stolen property distribution networks. TelosCam consists of small embeddable wireless motes and surveillance cameras. Each property owner will have a unique mote node (called *primary wireless module*) that stores the security information to authenticate himself/herself. The owner attaches a wireless mote (called *secondary wireless module*) to each of the properties to be protected, *e.g.*, a laptop or an electric bicycle or a vehicle. Here we assume that the installation of the secondary wireless module is tamper-proof. The primary wireless module node will periodically send the heart-beat information to each secondary primary wireless module node. The details of the heart-beat packets will be described later. Each secondary wireless module node will have three operation modes: *sleep mode*, *monitoring mode*, and *alarm mode*. When the secondary wireless module node receives the heart-beat information from the primary wireless module node, it will update its operation modes accordingly. If the secondary wireless module node is in the monitoring mode and cannot detect the existence of any primary wireless module node within its vicinity, it will start collecting its moving trajectory using motion detection component (if installed), and then send an alarm signal periodically, which will be received by some *reading wireless module nodes* associated with some surveillance cameras on the moving trajectory of the secondary wireless module node. The secondary wireless module node will remain in the *alarm mode* once it detects the movement of the property (same as the movement of the target burglar), and cannot detect the existence of the primary wireless module node. Once the target burglar passes by a surveillance point which is equipped with the mate of a camera and a wireless mote, a sequence of images of the burglar carrying the property are captured with high possibility and transmitted to a central server for further processing. The travel trajectory of the stolen property can be further tracked easily using the position sequence of the surveillance points the target passed by, in addition to the motion data collected by the secondary wireless module (if a motion detector is installed in the secondary wireless module). The possibility of identifying the burglar (e.g., a person or a vehicle), which is the common part of the image sequence, increases with the number of the surveillance points the target burglar has passed by. Eventually, the trajectory, and the critical and unique characteristics about the target are obtained, which can help identifying the burglar significantly, using the well-designed image matching techniques and probabilistic inferring based on multi-modal target data including spatio-temporal information about the target position and the topology of the surveillance infrastructures.

In this paper, we designed a framework where we can extract the object-wise semantics from a multi-camera system. This framework has following main components: discrete sampling of trajectory, video extraction, camera calibration, inter-camera data fusion. We developed a trajectory-based video extraction method to extract those videos which we believe will contain the target burglar with high probability. We also propose a filtering technique using the motion activity characteristics of the target to reduce the number of objects in an image (or a sequence of images) to be processed. It is well known that the same object may have different characteristics when captured by different cameras. To solve this calibration problem, we developed a correlation graph and matching based method. We use color histograms to determine inter-camera radiometric mismatch. Then, a maximum weighted matching is found to establish a mapping function between two cameras. In addition, we use a novel distance metric to determine the object correspondences.

Compared with the legacy video surveillance system, our designed TelosCam system has the following advantages:

1. **Camera Storage Efficiency**: it decreases the image storage requirement significantly due to the fact that only sequence of critical images stamped by wireless module messages, but not the completed raw data of all video streams, are recorded and stored in TelosCam.
2. **Identifying Burglars Efficiently**: By confining our research of potential burglars in all images to only the related images, using automatic object identification in images, and smart automatic objects mapping among images from different surveillance cameras, our system is able to quickly identify potential burglars. Our experiments also show that our automatic burglar pinpoint achieves almost 100% accuracy.
3. **Privacy Protection**: TelosCam can protect the user's

privacy compared to other camera surveillance or RFID systems according to the following facts. Firstly, there is no tight-coupling between the tag and the property to be protected in TelosCam. Secondly, the target tracking and identifying process can be triggered only by the owner of the property, adopting the authentication of both Tag ID and user' PIN number or password. Recall that the secondary wireless module node will periodically send out the alarm beacon only if it is in *alarm mode*.

Recently, Guha *et al.* [16] presented *AutoWitness* system to deter, detect, and track personal property theft. Their novel system uses accelerometer and the RF signal from cellular tower to compute the moving trajectory of the target. For reconstructing the trajectory path, they are able to achieve an accuracy of over 90% even if only crude localization (from cell towers) is available for the destination. Compared with our system, AutoWitness mainly focused on recovering the moving trajectory of the target, while ignoring the *target identification*. Our TelosCam system not only can estimate the moving trajectory of the target, but also can pinpoint the specific target that contains the stolen property. Our TelosCam system and AutoWitness system complement each other: the novel techniques presented in AutoWitness can be used in the TelosCam system to add extra trajectory information to improve the target identification accuracy.

Our Results: We designed, developed, and extensively tested our TelosCam system to study its performance. The real-world deployments demonstrate technical feasibility and effectiveness of the TelosCam design. A 4-surveillance-point network are deployed under the indoor scenario, where GPS solutions can't work, to catch the burglars on foot of important assets in an office building. We plan to test our system in the outdoor scenario, to catch the stolen assets carried by people on foot, by bicycle, or by car. We conducted extensive experiments of our TelosCam system on a variety cases (the number of secondary wireless modules may vary, the number of available surveillance cameras may vary, the total number of people that appeared in the system may vary, and the number of people with similar characteristics may vary, and so on). We found that in all cases, our TelosCam system can pinpoint the correct burglars with almost 100% accuracy using our auto-matching techniques.

The rest of the paper is organized as follows. In Section 2, we present the overview of TelosCam system and our design rational. We then show an intelligent triggering scheme that can protect the user's privacy in Section 3. In Section 4 we show how to efficiently retrieve needed video frames based on the collected trajectory. Several novel methods were presented in Section 5 to identify possible targets from these video frames. In Section 6, we report the extensive experimental results of our TelosCam system. Our experiments show that our system can uniquely identify the potential targets with almost 100% accuracy. We review the related work in Section 7 and conclude the paper in Section 8.

2. SYSTEM OVERVIEW AND DESIGN RATIONAL

2.1 Requirements and Challenges

The TelosCam system requires the detection of the property theft, tracking of the burglar as it passes by the surveillance points, and identification of the burglar (e.g., a car, or a person) from a sequence of images captured by surveillance cameras, while protecting the privacy of the owner. The system should be able to track and identify the target (or called burglar) in both realtime and offline. TelosCam system is an attractive solution for theft threat identification and reduction. Here the privacy protection requires that the data collected by the system cannot be used to enhance the trajectory tracking of the owner. Observe that we assume that surveillance cameras are already in use. The images captured by the surveillance cameras clearly will reveal some information about the trajectory of the target even the moving target is a legitimate owner. However tracking and identifying a moving target using multiple surveillance cameras is a notorious difficult problem. Traditional approaches of pinpointing the potential burglar target are often labor-intensive.

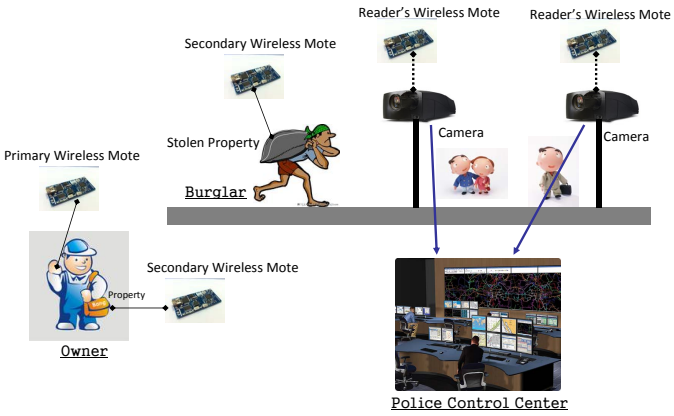


Figure 1: System flow of TelosCAM.

Figure 1 shows the architecture of TelosCAM system. Each owner will have a unique wireless module, that stores some private ID and key information from the owner. Each property will be attached a unique wireless module. Notice that here an owner may want to protect multiple asserts and a property may be protected by multiple users. The pairing between a primary wireless module and a secondary wireless module is configured when the user bought them. Each surveillance point, called TelosCAM mate, is composed of one or multiple cameras, and one or multiple wireless module nodes (we will discuss how we address the case when we only have surveillance cameras at some surveillance points). Wireless module node adopts some microcontroller for very simple computation and a radio chip for communication The overall system is built on an underlying network that connects multiple TelosCAM mates. In specific, TelosCAM

mate are deployed in a distributed fashion; the images from the cameras are filtered in some application-specific manner, and are then fused together in a form that makes it easy for an end user (human or some program) to monitor the area. The control center (that is computation intensive) will analyze multiple camera feeds from different regions to extract higher-level information such as presence or absence of a suspicious human or vehicle, and then identify the unique burglar carrying a reported stolen property (i.e., carrying a given secondary wireless module).

In our TelosCAM system, the integration of the wireless module nodes and surveillance camera will cause the privacy violation and may not reduce the labor-intensive identification of the target if the system is not carefully designed. In the next, we discuss in detail our design and design rational of our TelosCAM system that achieves the tracking and identification, and privacy protection simultaneously.

2.2 Overall Design and Design Rational

Intelligent Triggering Scheme and User Privacy Protection: Using a tag attached to the property to-be-protected and letting the tag send some information periodically to some control center has been used in many designs. If the tag contains the information that is unique to the owner, the privacy is not protected since the control center (which may be run by some commercial service provider) now has the needed data to compute the trajectory of the owner efficiently. Another challenge is when a tag should send information to the control center. For example, in the AutoWitness system, the tag will send the estimated trajectory information whenever the assert is moving and it is within the RF range of cellular tower. In other words, they use vehicular driving as an indicator of theft and as the trigger for sending out the alarm. Thus, the location information could be sent to the control center even if the owner drives the vehicle.

We present a privacy aware triggering scheme for determining if a tracking process needs to be triggered. The basic idea is to measure the distance between the protected property and its owner, if the property is out of a specified range of its owner, then the tracking processing will be immediately triggered. Otherwise, the tag on the property will stop sending beacon message to preserve the privacy of the owner. In our TelosCAM system, to extend the protection coverage, we will use an additional wireless module node (called primary wireless module node) for each user. The tag attached to an assert will not send any information whenever it detects a pairing primary wireless module node in its vicinity, or it was informed by the primary wireless module node to remain silent.

Video Retrieval: Once the tracking process is triggered, the secondary wireless module attached to the property periodically broadcasts alarm messages. This alarm message will be received by some nearby wireless module nodes associated with some surveillance cameras. According to the received alarm message, TelosCAM system starts retrieving interested image frames from all the related surveillance cameras (based on projected moving trajectory of the potential burglar). TelosCAM extracts information from a video stream by invoking the corresponding wireless module node. Once a wireless module node detects a suspicious event, *e.g.*, a protected property has entered this wireless module's communication range, the corresponding camera starts capturing frames from current video stream. If the bandwidth required to disseminate all streams exceed the available bandwidth at the control center, network will end up dropping packets. This leads to the need for priority-aware communication in the data network. Based on these needs, the prioritization strategies employed by our system can be grouped into the following categories: priority-aware computation and priority-aware communication.

Video Processing: For simplicity, assume that we know an assert, with the secondary wireless module node i attached to it, is reported stolen. After the control center received video frames from surveillance cameras that more likely will contain the suspicious target (carrying an assert with tag i), the control center will then apply techniques from computer vision and graphics to detect objects (called *blob*) in each video frame and match the objects among different frames to find one suspicious object in these video frames that more likely will carry the stolen assert. The system will pinpoint a suspicious person or car from among all the relevant videos captured, using some novel matching techniques to be discussed in detail later.

3. PRIVACY PRESERVING TRIGGERING SCHEME

We first present a privacy aware triggering scheme for determining if a tracking process needs to be triggered. Our scheme is able to improve the operational privacy with respect to the methods of the prior art. This objective is achieved by using a pair of wireless module nodes, one is carried by the owner (called primary node) and the other one (called secondary node) is attached to the property to be protected, determining the distance between each other.

Consider a pair of wireless module nodes a and b. A wireless module node a is attached to the property and the other wireless module node b is carried by the owner. Node b emits RF signals, and node a periodically checks for the presence of a transmission, and performs the requested function only if fields within the message are from node b. The secondary wireless module a will have three operational modes: *sleep mode*, *monitoring mode*, and *alarm mode*. A secondary wireless module is typically in the *monitoring mode*. It switches to the *sleep mode* if the owner of the property knows that the property will be within its view for a duration of T time units, and then sends command to the secondary wireless node asking it to remain in sleep mode for the next T time units. After the sleep timer expires, it will wake up and be in the monitoring mode. The tracking process (thus the secondary module changes to the *alarm mode*) is triggered if

and only if the property is out of transmission range of the owner. A secondary node will not transmit any signal when it is in the sleep mode or the monitoring mode. In this sense, the triggering scheme is able to protect the owner's privacy by isolating the tracking process from the owner's normal daily activity.

However, the above design suffers from potential security issues if not designed carefully. For example, the attacker can listen to and record the message that is previously transmitted by the primary node b, and play back the packets at any time in the future. Play-back attacks can therefore be implemented very inexpensively. A simple method of preventing relay attack is to include a simple dynamic security code in the message that changes with each transmission. The receiver calculates the next code in the sequence, and accepts a message as valid only if the received code matches the expected code. To implement this, we often need a pseudorandom number generator such that the future produced sequence cannot be predicted based on the collected historically generated sequence. Message authentication using public key based cryptographic techniques is a better method for preventing spoofing and playback attacks. However, the available authentication algorithms based on public key systems have been too complex for implementation in very low cost systems.

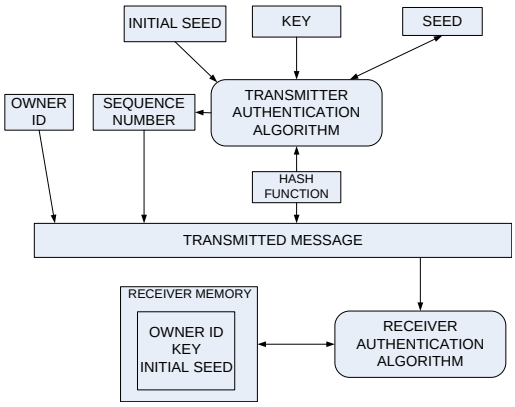


Figure 2: Work flow of authentication scheme.

In our TelosCAM system, we use the sequence number (together with the time-stamp), the owner's ID, and a hash code (using a hash algorithm $h(\cdot)$) of this information (together with the security information shared between the primary wireless module and the secondary wireless module) is included in the message structure to prevent the recording and subsequent playback of legitimate messages, and to prevent a receiver from being deceived into accepting messages from unauthorized sources. The security code sequence is to assure that it is not predictable from knowledge of past sequences. Once a transmitter is manufactured, it is programmed with a transmitter identification (ID), a default starting sequence number (N), a randomly generated initial seed S_0, and a cryptographic key k. The transmitter ID and initial seed S_0 are unique to each transmitter. The cryptographic key k may be common for a subset of primary transmitters, or unique to each transmitter. The randomly generated initial seed is used as the starting point from which the authentication algorithm advances with each transmission. The sequence number N and the timestamp t also advances with each transmission to indicate to the receiver the required number of advances that it must perform to cryptographically synchronize with the transmission. The algorithm operates on a seed code s which is changed according to a set of rules for each transmission. The seed code s could be produced based on some random number generator g using s_0, k, and N as seed, i.e., $s = g(s_0, k, N)$. A sequence number N is also incremented with each transmission and is included in the message so that the receiver (the paired secondary wireless module) will know exactly the value of s by knowing S_0, k, N, and the algorithm g to produce s. Observe that the secondary wireless module often will not receive each transmitted message from the primary wireless module.

To be specific, the information that is stored at a primary wireless node is (1) its own information: ID, initial seed s_0, (2) the key k shared between itself and a pairing secondary wireless module, (3) the algorithm g and h, and dynamically changing sequence number N, and seed S. The information stored at a secondary wireless node include the primary user ID, the initial seed s_0 of a primary user, the pairing key k, and the last received sequence number P from this primary user. If multiple users can protect an assert, then the secondary wireless node will store multiple $\langle ID, s_0, k, P\rangle$. The heart-beat message sent by the primary wireless module is

$$\langle ID, N, t, h(s_0, k, s, t)\rangle$$

The primary user will increment N. When a secondary wireless module received a heart-beat message $\langle ID_i, N, t, h(s_0, k, s, t)\rangle$, it finds the corresponding information $\langle ID_i, s_0', k', P\rangle$. It changes its state to *alarm mode* if one of the following conditions is true

1. it cannot find the user with ID_i, or
2. $P \geq N$, or
3. $h(s_0', k', g(s_0', k', N), t) \neq h(s_0, k, s, t)$.

A secondary user will remain silent only if the previous three conditions are true and the received timestamp t is within a reasonable drift of its own clock t_0. A secondary user will perform one round of clock synchronization if it found that t is significantly different from t_0. It will send a message containing its own clock t_0, and $h(s_0, k, s, t_0)$ to the primary user and asks the primary user to send a new heart-beat message using clock t_0. The primary user will check the validness of this request (using hash value $h(s_0, k, s, t_0)$) and then reply a heart-beat message and synchronize its clock to t_0. If the received new heart-beat message does not pass the previous three conditions, the secondary wireless module will switch its status to *alarm mode* and start sending out the alarm messages periodically. This will prevent a possible

playback attack and also synchronize the clocks between the pairing wireless modules. If the security checks passed, the secondary wireless module will update P as N accordingly.

4. TRAJECTORY BASED VIDEO RETRIEVAL

Requirements and Challenges: When a target burglar passes through a surveillance point, the reading wireless module(s) associated with this surveillance point will receive some alarm messages sent by the secondary wireless module attached to the stolen property. Video retrieval requires extracting those video which have high probability to contain the target. The retrieved video will be further processed later in order to identify the target.

Remembering that a wireless module node is able to tell whether or not there is a target appearing in its sensing range. Consider one specific surveillance point and one specific secondary wireless module tag_i. Let t_e be the first time when an alarm message from tag_i was received by a reading wireless module and t_l be the last time when an alarm message was received. This time information can also be used to estimate the moving speed of the object: $v = D/(t_l - t_e)$ where $D \simeq 2r$ and r is the transmission radius. Let t'_e be the first time when the target burglar carrying tag_i appeared in some images captured by this specific surveillance camera, and t'_l be the last time when this target burglar appeared in videos captured at this point. Although we can easily get the time t_e, and t_l, it is challenging to get the timestamps t'_e and t'_l. If we can get these two timestamps t'_e and t'_l, we can effectively have a video frame in which the target burglar will more likely appear in all images.

A naive video retrieval scheme is to let each camera start to store the video once the corresponding wireless module node detects the appearance of the object, until the object left the sensing area, *i.e.*, we set $t'_e = t_e$ and $t'_l = t_l$. Apparently, this scheme provides high reliability that the suspicious target is contained inside this video frame due to high frequent video capturing and storing. However, it suffers from poor storage efficiency, *e.g.*, significant amount of the captured video may not contain the target. This is mainly because the communication area of a wireless module node is typically different from the one of a camera, *e.g.*, wireless module node's communication region is a disk, but the sensing area of a camera is more like a sector. In other words, the sensing result from wireless module node is not able to directly tell whether a target has appeared at the sensing area of a camera. Another disadvantage of this naive approach is that it may introduce a large amount of noise for later object classification: many unrelated objects may be introduced to the system. Here an object is a human being (or a car) appeared in the subset of images to be processed later. To save storage and improve the accuracy of later algorithms, we would like to estimate t'_e and t'_l as accurate as possible.

To this end, we propose a trajectory based video retrieval scheme, aiming to extract the video with highest suspicious. The basic idea is to reconstruct the trajectory of the burglar, based on which the system can estimate the time when the object entered and left the camera sensing range. These information can be further utilized to filter out those video which are less likely to contain the target. The trajectory reconstruction problem can be modeled as a binary tracking problem, under which each TelosCAM mate has a sensing range such that it can report "yes" or "not" anytime to the question: "whether is there some target within its sensing range?". We give a formal problem statement as follows. Suppose that there is one object, moving through the field monitored by a set of surveillance wireless module nodes. Each surveillance wireless module reports its 1-bit reading, according to the presence or absence of targets at its sensing range. We further partition the road segment into n intervals with equal length, $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$. Obviously, the more intervals we partitioned the road segment into, the more accurate the computed location is. If we let I be the set of sensors whose binary output is 1, then the target must be located at some interval lying in the sensing area of I. For ease of presentation, we call the union of sensing area of I feasible target space: Based on the sensor readings, let the set of feasible target spaces be $\mathcal{F}[t] = \{F(t)\}$, where $F[t] \subseteq \mathcal{S}$ denotes the feasible target space at instant t. Please see Figure. 3 as an example, in this case, both mote 1 and mote 2 detects the target at the same time, thus the feasible target spaces are those shadowed intervals. Given the set $\mathcal{F}[t]$, we wish to generate estimates of the target trajectories, denoted by $\{x[t] : t \in \{1, 2, \cdots, T\}\}$, where the $x[t] \in \mathcal{S}$ denotes the location at the time instant t.

The use of particle filters for tracking an object has been proposed in [27] [28]. For presentation completeness, we next provide a sketch of the approach. We begin at $t = 1$, and proceed step by step to $t = T$, while maintaining a (large) set of K candidate trajectories (or particles) at each instant. At any time t, we have K particles (or candidate trajectories), with the current location for the kth particle denoted by $x_k[t]$. Each of those K particles extend to the next time instant $t + 1$ by choosing m candidates uniformly at random from $F(t + 1)$. We now have mK candidate trajectories. Pick the K particles with the best cost functions to get the set $x_k[t + 1], k = 1, \cdots, K$, where the cost function will be defined later. Repeat until the end of the time interval of interest. The final output is simply the particle (trajectory) with the best cost function.

The cost function, $c_k[t]$, which used to select most possible trajectory, is defined as the norm squared of the difference between the velocity estimates at two consecutive time intervals, *e.g.*, time interval $[t, t + 1]$ and time interval $[t - 1, t]$. The intuition behind this definition is that sudden changes to velocity are unlikely to happen in smooth paths.

$$c_k[t] = (||x_k[t + 1] - x_k[t]|| - ||x_k[t] - x_k[t - 1])||)^2$$

The overall cost function associated with a trajectory par-

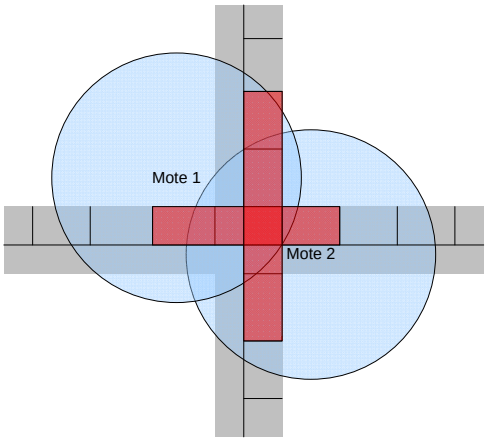


Figure 3: Illustration of feasible target space. In this example, both mote 1 and mote 2 detect the appearance of target, the corresponding feasible target space is composed of those shadowed intervals.

ticle $\{x_k[t]\}$ is the sum of all incremental costs:

$$\sum_{t=1}^{T} c_k[t]$$

From the predicted moving trajectory, we are able to estimate three types of information: 1) the time, denoted by t'_e, when the target enters this camera's view, 2) the time, denoted by t'_l, when the target leaves this camera's view and 3) the target's moving direction. By knowing the former two types of information, we are able to extract the desired video which contains the target with high probability. In specific, the video that is recorded between t'_e and t'_l is extracted and will be processed later. More importantly, by computing precise t'_e and t'_l, the *most frequently* appeared object(s) in this extracted video (to be found by using methods discussed in next section) more likely will be the target of interest. The third type of information, *e.g.*, the moving direction, will be utilized in the later stage as an important feature to find out most suspicious targets.

Priority-Aware Communication Scheme: Assume that traffic is classified into two priority classes: extracted video that will contain some suspicious targets in the view, and the rest of videos. The extracted video will be sent back to the control center with the highest priority. The video that may not have the suspicious targets will be sent back to the control center only when network capacity permits.

5. HIERARCHICAL TARGET BURGLAR IDENTIFICATION

Note that the video extraction method discussed in previous section will have a nice property: the target burglar more likely is one of the most frequently appeared objects in this extracted video. We next aim to identify the burglar from set of retrieved videos. In specific, given a set of videos from different cameras, we want to identify the object with most occurrences across those videos. To achieve this goal, we first present a target detection and classification scheme in order to identify a set of candidate objects, *e.g.*, k candidate objects, with high suspicious to be the target from single camera.

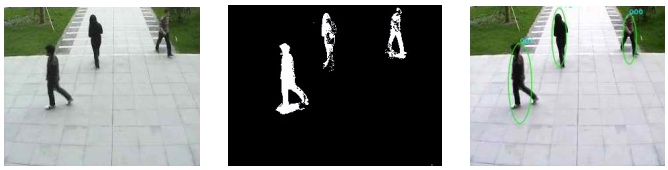
5.1 Object Classification From Single Camera

For every incoming extracted video from a camera, we want to find a set of objects (in our experiments, a set of different human beings) that appeared in this video. First a binary image is obtained by performing background subtraction, in which white pixels correspond to detected foreground objects. The background subtraction is implemented by using a robust and light-weight salient foreground detection algorithm proposed in [11]. Then, foreground pixels are grouped into blobs by connected component labeling. Each blob corresponds to a detected object. When a new foreground blob is detected, a new tracker is created. The label of this tracker, the coordinates of the bounding box and the color histogram are saved in the tracker. We use a color histogram to model the appearance of an object. Each bin in a 3-D histogram corresponds to an (R,G,B) range. In [25], a P2P multi-camera system is presented wherein each camera is attached to a different CPU and cameras have partially overlapping fields of view. For tracking on a single camera view, we use an optimized version of the tracking algorithm introduced in [25]. At every frame, trackers are matched to detected foreground blobs by using a matching criterion based on bounding box intersection and the Bhattacharyya coefficient [12]. For the candidate foreground blob centered at location $\mathbf{y}$, the Bhattacharyya coefficient is derived from the sample data

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y}), \hat{q}_u} \qquad (1)$$

where $\hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1,2,\cdots,m}$, $\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1,2,\cdots,m}$ are the probabilities estimated from the m-bin histograms of the model in the tracker and the candidate blobs, respectively. If the bounding box of a foreground blob intersects with that of the current model mask of the tracker, the Bhattacharyya coefficient between the model histogram of the tracker and the histogram of the foreground blob is calculated by using Equation (1). The tracker is assigned to the foreground blob which results in the highest Bhattacharyya coefficient, and the bounding box of the tracker is updated. The Bhattacharyya coefficient with which the tracker is matched to its object is called the similarity coefficient. If the similarity coefficient is greater than a predefined distribution update threshold, the model histogram of the tracker is updated to be the histogram of the blob to which it is matched. Based on this matching criterion, if objects merge, multiple trackers are matched to one foreground blob. The trackers that are matched to the same blob are put into a merge state, and in this state their model histograms are not updated. The de-

tails of handling merge/split cases on a single camera view can be found in [25]. Figure 4 illustrates different stages of object classification in a single camera.



(a) (b) (c)

Figure 4: Different stages of object classification from single camera: (a) original image, (b) background subtraction, (c) objects classification

Initial Target Filtering: In this stage, we aim to select a small number of high suspicious objects from each extracted video. The underlying motivation is to reduce the number of objects to be processed and further improve the identification accuracy. Remember that in the previous stage, we are able to obtain the entry location and moving direction of the target from trajectory prediction. We first filter out those objects whose moving direction is different from the predicted one. Then we rank all the remaining objects in non-increasing order of their appearance durations. By selecting top-k objects from each extracted video, we reduce the searching space dramatically. Here k is a constant parameter used in our system.

5.2 Burglar Identification Across Multiple Cameras

After selecting k objects from each single camera, we try to identify the object with most occurrences across different cameras. We will then label this object as the final pinpointed burglar.

5.2.1 Inter-Camera Calibration

Recall that TelosCAM consists of several non-overlapping view of cameras. Usually, multiple identical cameras that are operating under various lighting conditions, or different cameras that have dissimilar radiometric characteristics. Even identical cameras, which have the same optical properties and are working under the same lighting conditions, may not match in their color responses. Images of the same object acquired under these variants show color dissimilarities. As a result, the correspondence, recognition, and other related computer vision tasks become more challenging. In this work, we propose the following color calibration techniques in order to tackle those issues.

We compute pair-wise inter-camera color model functions that transfer the color histogram response of one camera to the other in the calibration stage. First, we record images of the identical objects for each camera. For the images of an object for the current camera pair, we find color histograms $\mathbf{h}_1$, $\mathbf{h}_2$. A histogram, $\mathbf{h}$, is a vector $\{h[0], \cdots, h[N]\}$ in which each bin $h[i]$ contains the *percentage* of pixels in this object corresponding to the color range. Using the histograms $\mathbf{h}_1$ and $\mathbf{h}_2$, we compute a weighted bipartite graph between two histograms as the positive weighted edge represent the bin-wise histogram distances where each element $w_{i,j}$ is a positive real number such that $w_{i,j} = d(h_1[i], h_2[j])$ and $d(\cdot) \geq 0$ is a distance norm. Given two histograms and their correlation graph, the question is what is the best mapping between colors from those two histograms? We reduce the mapping of two histograms to finding the maximum weighted matching in the correlation graph. Finding such a matching is studied as the assignment problem. It can be solved by using a modified shortest path search in the augmenting path algorithm. If the Bellman-Ford algorithm is used, the running time becomes $O(V^2E)$ where V is the number of nodes and E is the number of edges. We then establish a mapping function between two histograms as follows: $f(i) \rightarrow j$, *e.g.*, color i in the first histogram is mapped to color j in the second histogram. Here i is matched with j in maximum weighted matching.

5.2.2 Burglar Identification

After camera calibration, we focus on the following problem: *Given a set of extracted videos from different cameras, each of which contains k suspicious objects, we aim to quickly and accurately identify the target among those objects that has the most occurrences across all images.*

To find the corresponding objects in different cameras and in a central database of the previous appearances, we evaluate the likelihood of possible object matches by fusing the object features such as color, height, movement *etc.*.

Color: After color calibration, similarity between color histograms s_C is a main evidence for appearance similarity, it is given the highest weight among all the criteria. s_C is calculated based on Bhattacharyya coefficient by using Equation (1). The color histogram is built when the object is in a good position in the view, such as with a better resolution or when it is not occluded. For instance, if the object is in the merge state, the color histogram will not be constructed until merge is resolved.

Height: If H_1 and H_2 are the object's heights measured at the entry locations in the first camera and second camera views, respectively, the height similarity s_H is calculated by: $s_H = |\frac{H_2 - H_1}{H_1}|$.

Speed: If V_1 and V_2 are the object's estimated speeds in the first camera and second camera views, respectively, the speed similarity s_V is calculated by: $s_V = |\frac{V_2 - V_1}{V_1}|$. For an object captured by a camera, its speed is estimated as $L/(t_l - t_e)$ where L is the estimated distance traveled by this object and t_e and t_l are the entry time and leaving time of this object in this camera.

We combine multiple features by calculating a weighted sum of the similarity score of each feature

$$f_s = \omega_1 s_C + \omega_2 s_H + \omega_3 s_V$$

A proper threshold and weight assignment for the overall

similarity are learned during training stage. We then build a similarity graph $G = (O, E)$, where O is the set of objects from all s surveillance cameras. Consider two objects from different cameras, if the overall similarity score between these two objects is greater than a pre-defined threshold β, *e.g.*, $f_s \geq \beta$, then we add an edge between these two objects. Based on the resulted similarity graph, we next convert the burglar identification problem to a *Maximum Clique Problem*. When the size of similarity graph is small, we can always find the maximum clique through brute force. However, general maximum clique problem is NP hard. Several attempts in the literatures [13] [9] have been made to find a clique that, although not maximum, has size as close to the maximum as can be found in polynomial time, and those approaches can be used when input similarity graph is extremely large.

After the maximum clique is found, all objects that belong to that clique are labeled as the burglar. The intuition behind this approach is that, real burglar intends to have most occurrences across all extracted videos. Recall that the retrieved video is extracted based on the predicted trajectory of burglar, then if the predicted trajectory is reasonably accurate, most of those videos must contain the burglar. Therefore, compared with other objects, burglar should appear most frequently across those videos. Our method is illustrated by an example in Figure. 7. In this example, we have four videos from four cameras. Through the above similarity computation, we construct a similarity graph where we add an edge between any pair of objects whose similarity score is higher than certain threshold. Apparently, two cliques are naturally formed, one is of size 3 and the other is of size 4. By selecting the larger one, which is represented by red lines, our system successfully identify the burglar (the one who wears red sweater).

Assume that the most occurred object may have some unique feature f (such as color, height, and moving speed). Notice that it is possible that for some extracted video, we may have multiple objects with this feature. After finding the most occurred feature f, we will then find a camera such that there is only one object in its extracted video with this feature f. That object will be labeled as the final identified burglar. Otherwise, we will return all objects from the clique with feature f as identified burglar(s).

6. IMPLEMENTATION AND EVALUATION

6.1 Experiment Setup

6.1.1 TelosCAM Implementation

To avoid the high cost of the real-time video data collection from the distributed cameras, we conduct the experiment in an offline mode. Instead of transmitting the captured video data of all the networked cameras to a remote center, the laptops are used as the distributed storage components of the whole networked system. In fact, one laptop is connected with one sensor node and one camera, and all the three components are together taken as one sensor-camera mate unit.

In one sensor-camera mate unit, the sensor node component is the TelosB node [24], low-consumption motes equipped with a Texas Instruments MSP430 microcontroller (8 MHz, 10-kB random access memory, 48-kB Flash memory), and a radio chip Chipcon CC2420 (support up to 250 kbps data rete), which implements the communication protocol IEEE 802.15.4 [5]. The sensor nodes of the sensor-camera mates are powered via Universal Serial Bus (USB) connection. The TelosB node taken by the burglar is powered by two AA batteries.

The sensor program is developed based on TinyOS 2.1 [3]. We implement a simple one-hop transmission protocol by using nesC language [14]. By using the transmission protocol, one sensor node can transmit data packets to another sensor node within its communication region. The transmission protocol adopts the best-effort mode in which no ACK message is needed to response the sender of the previously received message. Because the experiment is conducted in the in-door environment, we change the transmission power of the TelosB nodes so that the communication radius of the sensor nodes is similar with the sensing radius of the camera (which is about $5m$). In our experiment, we set the transmission power of the TelosB nodes to level 5 through the TinyOS interface. Observe that our TelosCAM system works correctly in a general scenario when the relation between the sensing range of camera and the communication range of TelosB nodes are arbitrary.

The web cameras with five million pixels are used to capture the video information, and are directly connected to the laptops via the USB interfaces. The cameras sample the visual information of the surveillance regions at a frame rate of 15 Hz, and the resolution of the captured video sequence is 360×240 pixels. The video processing algorithm was carried out on the platform of VC++ .NET 2005 combined with OpenCV (the open source computer vision library supported by Intel Corporation). The first 150 frames of each camera's video sequence are used to model the background of the video data captured from the corresponding surveillance region, the background subtraction method is based on the computational color model presented in [17] [26].

6.1.2 Deployment of Networked Sensor-Camera Mates

Fig. 5 shows the deployment situation of the Networked sensor-camera mates. The deployment area is at the first floor of an office building. The width of the building gallery is about 4 meters. 6 sensor-camera mate units are placed at the positions shown in Fig. 5. The web cameras are fixed at the top of the tripods which is about 1.9 meters high, and the directions of the web cameras are fixed. The viewing angle of the web camera is about 45 degrees, and the zoom operation is not used.

6.1.3 Data Collection

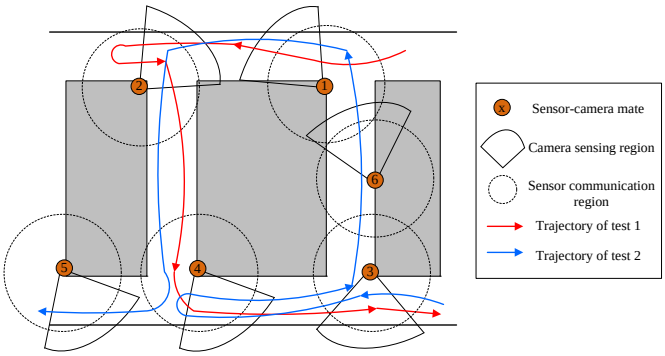


Figure 5: Network deployment and test trajectories.

The web camera continuously captures the video data from its surveillance region. Once the TelosB node detects another node which enters into its communication region, it activates the system to save the current video data in the laptop. Then, the two sensor motes keep communication to tell the system there may be a suspect burglar within the sensing region of the corresponding web camera. When the suspect burglar leaves the communication region of the TelosB node and the communication of the two nodes is interrupted, the system stops saving the captured video data in the laptop. The laptop system time is used as the reference time of all the operations performed by the sensor node and the web camera.

6.2 Experimental Results

6.2.1 Test of Single Mate Scenario

First, we test the object identification capacity of our approach based on the sensing information obtained from single sensor-camera mate. In this test, the burglar that took one TelosB node walked through the sensing region of camera 4. Assume the walking speed of people is 0.5 m/s. Fig. 6 shows some selected frames in the saved video sequence captured by camera 4. Because the video sequence starts at the time of the burglar entering the communication region of sensor 4 and ends at the time of the burglar leaving the communication region of sensor 4, the people with the maximum occurrence number in the saved video sequence is usually the most suspicious object.

As shown in Fig. 6, there are two people in the earlier video frames, and we can not identify the suspicious object based on these video frames. However, there exists about 1.5 meters distance between the two people, and this means that the time of the burglar staying in the saved video is about $1.5/0.5 = 3$s more than that of the other people. Therefore, it can be seen that there is one people in the last dozens of video frames, and the occurrence numbers of the two people are different. In the test video sequence, the people with the maximum occurrence number is the people marked with the red triangle in Fig. 6, and it is taken as the burglar.

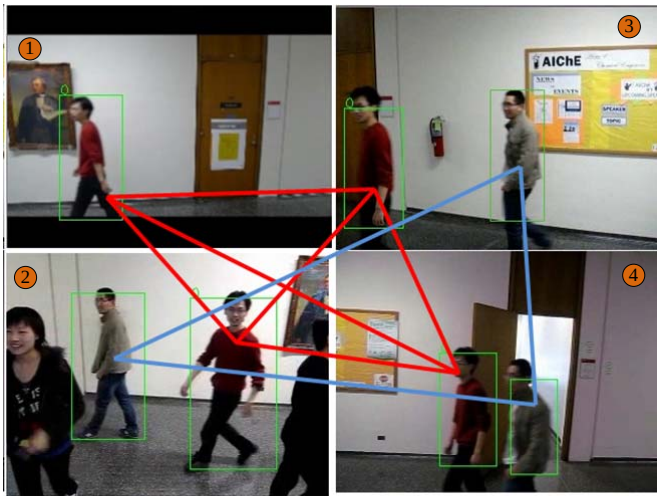


Figure 7: Four selected frames which are respectively captured by camera 1, 2, 3, and 4 in Test 1.

6.2.2 Test 1: Candidate feature with single occurrence in each camera

In test 1, the trajectory of the burglar is depicted in Fig. 5 by using the red curve. When the burglar was within the communication region of the sensor nodes, the video data captured by the corresponding cameras was saved in the laptops. For identifying the burglar, each saved video sequence was first processed independently to select the top-5 objects which appeared in every video sequence, and the color histogram features of every object were figured out at the same time.

Fig. 7 shows the four frames which are respectively selected from the saved video sequences captured by camera 1, 2, 3, and 4. In each video frame, the top-5 objects are indicated by the green rectangles. After selecting the top-5 objects, the selected objects that appeared in the different camera sensing regions were matched based on their statistical features. In Fig. 7, each pair of the successfully matched objects is connected by using a line. As you can see from the figure, there are two cliques which are respectively indicated by the red and blue lines, and this is because that the trajectories of the two objects are the same within a certain period of time. However, other people do not walk with the burglar all the time (which is the basic assumption of our approach), and thus the sizes of the two cliques are different. The object with the maximal clique is the most suspicious burglar. In Fig. 7, the number of vertices in the clique indicated by the red line is 4, and the number of vertices in the clique indicated by the red line is 3. Therefore, our approach takes the person wearing the red clothes as the burglar. This matches the ground truth.

6.2.3 Test 2: Candidate feature with multiple occurrences in some camera

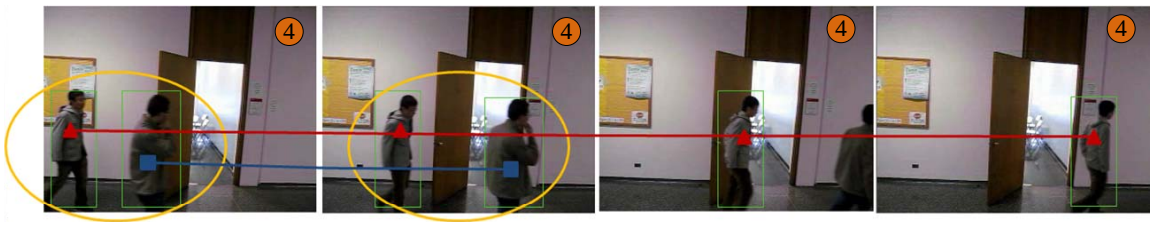


Figure 6: Selected frames in the saved video sequence captured by sensor-camera mate 4.

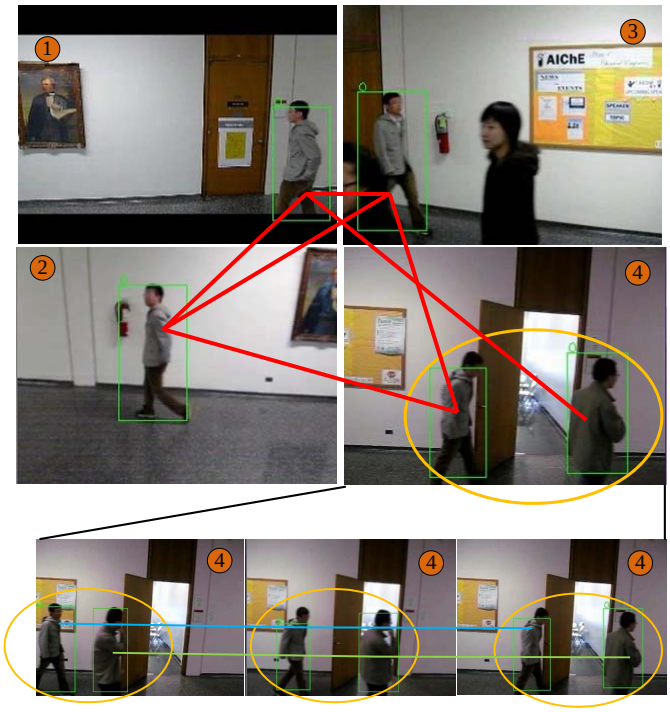


Figure 8: Selected frames which are respectively captured by camera 1, 2, 3, and 4 in Test 2.

In test 2, the burglar wearing gray clothes walked along the blue curve trajectory which is also depicted in Fig. 5. Fig. 8 shows the frames which are respectively selected from the saved video sequences captured by camera 1, 2, 3, and 4 in test 2. After selecting the top-5 objects from the four video sequence respectively, there are two objects that have the quite similar color histogram features in the video sequence captured by camera 4. Therefore, as shown in the figure, some errors happen in the object matching process, and the 4-vertex clique can not be formed based on the visual information of mate 1, 2, 3, and 4. However, in the video sequence captured by camera 1, 2, and 3, there is only one person wearing the gray clothes, and no matching error happens due to the considerable differences of the statistical features of the different top-5 objects. Consequently, a clique with 3 vertices is formed, and it is the maximal clique in this test. Finally, the burglar is catched successfully.

6.2.4 *Test 3: Candidate feature with zero occurrence in some camera*

Because the communication region of a TelosB node is different from the sensing region of a web camera, the video sequence recorded in the laptop may not contain the burglar in some situations. In test 3, the burglar also wears gray clothes, and the trajectory of the burglar is depicted by using a red curve in the right part of Fig. 9. As shown in the figure, when the burglar entered the communication region of the sensor node 3, the laptop began to save the video data captured by camera 3. However, before entering the sensing region of camera 3, the burglar went back along the path which it had just walked along. Therefore, the video sequence recorded in laptop 3 does not contain the burglar at all. At this time, once an object whose statistical feature is quite similar with that of the burglar is within the sensing region of camera 3, like the case shown in the left part of Fig. 9, some errors of the object matching may happen, and thus there is no a 4-vertex clique in this situation. Though this disturbing factor can reduce the accuracy of successfully catching the burglar to a certain extent, we can still make the right decisions as long as there is enough correct sensing information. Like the situation given in Fig. 9, we can use the visual information obtained from camera 1, 2, and 4 to find the burglar successfully.

6.3 Performance Evaluation

In this subsection, we evaluate the accuracy of our approach under the various experimental situations. First, we show the relationship between the accuracy of identifying the burglar and the number of burglar-contained cameras, and it is depicted in Fig. 10. When there is only one camera whose visual information contain the burglar, the accuracy is quite low. In this case, the burglar can only be identified based on the occurrence number of objects in the recorded video sequence and this judgment condition is trustless in many scenarios. Since the walking speeds of different persons are different and the communication region of sensor nodes and sensing region of cameras are inconsistent, the time of the burglar appearing in the video sequence may not be the longest one among that of all the objects in the video

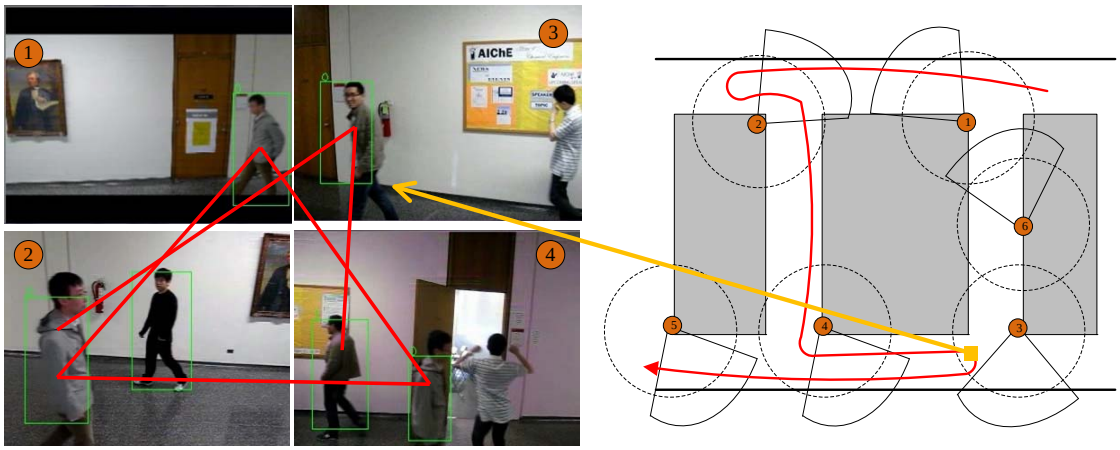


Figure 9: Trajectory of burglar and selected frames which are respectively captured by camera 1, 2, 3, and 4 in Test 3.

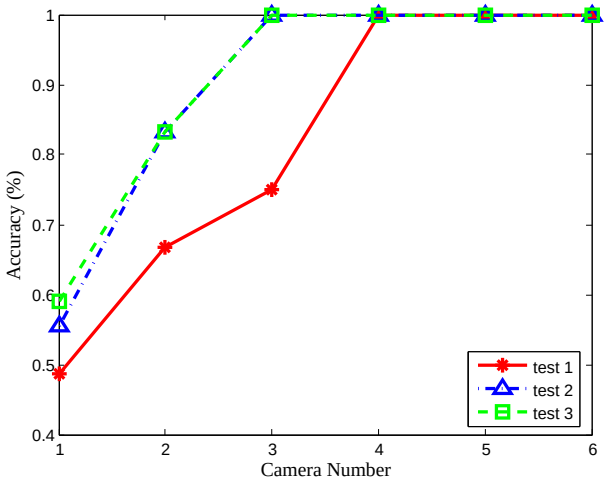


Figure 10: Accuracies of 3 tests versus number of cameras.

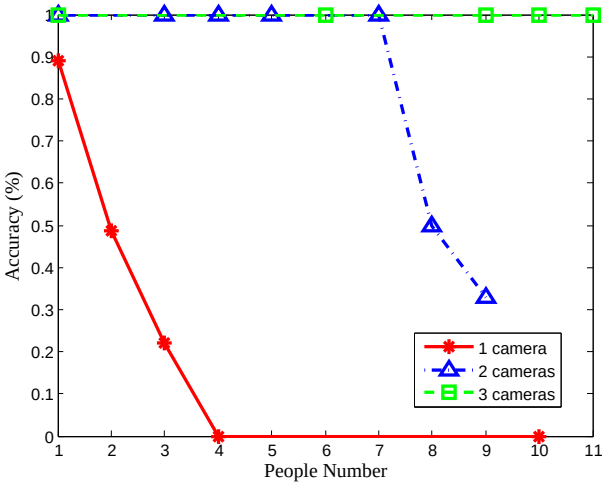


Figure 11: Accuracies based on sensing information obtained from different number of cameras versus the number of people.

sequence, or even is 0.

As the number of useful cameras increases, the accuracy improves observably. When there are 4 useful cameras whose visual information is recorded to identify the burglar, the identification accuracies of all the 3 tests are 100%. Therefore, our burglar identification approach can obtain a good identification performance at a quite lower storage or communication cost. In addition, from the viewpoint of the identification speed, the small number of the needed camera means that the burglar can be successfully catched very fast.

Second, we evaluate the influence of the number of people on the accuracy of our burglar identification approach. For the reasonable performance evaluation, we also conduct a few tests, and the accuracy is the average of those obtained from all the tests. Fig. 11 depicts the tests' result that shows the relationship between the accuracy and the number of people under the cases of 1, 2, and 3 cameras. In the case of 1 camera only, when there is only one object in the saved video sequence, the accuracy is high. However, the accuracy is reduced significantly as the number of people is increased, and it cannot identify the burglar successfully when there are more than 4 objects in the saved video sequence. In the case of 2 cameras, when there are small number of objects (e.g., less than 7 in our tests) in the two saved video sequences, the burglar can be catched successfully. The identification accuracy begin to decrease when the number of people is increased to some value (e.g., 8 in our tests). In the case of 3 cameras, our approach can identify the burglar successfully in any situation (in which the number of people is actually from 1 to 11). Therefore, 3 cameras are enough to iden-

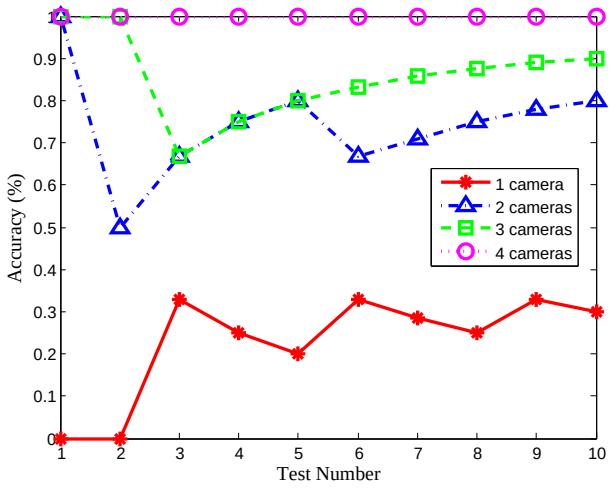


Figure 12: Accuracies based on sensing information obtained from different number of cameras versus number of tests.

tify the burglar in about 10 people's situation, and this also means that the proposed approach can achieve a quite good identification performance by using a small number of cameras. In all tests here we assume that there is only one stolen property, *i.e.*, one secondary wireless module will send out the alarm message.

Third, we evaluate the stability of our target identification system. Fig. 12 shows the average identification accuracy versus different numbers of tests conducted. The x-axis is the number of tests conducted, and the y-axis is the average accuracy of identification over all the tests conducted so far. In this figure, when the test number is small, the average identification accuracies of the 1, 2, and 3 cameras' cases vary significantly. As the number of the tests increases, the average accuracies of all the cases are gradually stabilized. 1 camera's case has the lowest average accuracy which is always below 50%. The 2 and 3 cameras' cases have good identification performance in most situations, and the identification of target is always successful in the 4 cameras' case. Therefore, the stability of our target identification system is good enough even in the case with a small number of cameras.

7. RELATED WORK

Numerous approaches have been proposed for object tracking in the literature. A number of algorithms for tracking moving objects across multiple stationary cameras have been recently proposed [20] [23] [22]. Kang *et al.* [19] presents novel approaches for tracking moving objects observed by multiple, stationary or moving cameras. Previous approaches are limited to the case of synchronized cameras for ensuring correspondence across views. In [29], the author proposed an approach for space and time self-calibration of cameras. Tracking moving objects from a non-stationary camera often assumes the accurate estimation of the camera motion [8] [18] [6]. In [30], the global estimation of trajectories and bounding boxes using Tensor Voting based tracking approach was proposed. It achieves smooth and continuous trajectories and bounding boxes, ensuring the minimum registering error. Very recently, Guha *et al.* [15] present AutoWitness, a system to deter, detect, and track personal property theft. Together with novel hardware design, they use model of city streets and Viterbi decoding to estimate the most likely path.

RFID has been used for tracking asserts since RFID was introduced. Typical approaches include hierarchical RFID tracking system [21], active RFID tracking [7], supply chain tracking [10]. These systems typically are not designed to pinpoint the suspicious target that may carry the stolen asserts.

8. CONCLUSION

In this paper, we present a complete system for tracking a lost property using wireless sensor networks and digital cameras. Our system can provide efficient automatic tracking of the property without sacrificing the privacy of the owner of the object, and effectively pinpoint the suspicious target (a person or a car) using novel object classification and matching methods. Our extensive experiments show that our system can pinpoint the suspicious targets with a surprisingly good accuracy almost 100%. Our system can be complemented by some existing approaches to further improve its efficiency and effectiveness. It also gracefully deal with the case when the reader may miss some alarm messages from some secondary wireless module, or when some surveillance cameras do not have associated wireless module reader. A future work is to extend the design to cope with scenarios such as environment with bad visibility when it is difficult to classify and match objects using videos.

9. REFERENCES

[1] Brickhouse security gps tracking system. http://www.brickhousesecurity.com/gps-tracking-system.html.

[2] F. UCR. Burglary - crime in the united states - 2008. http://www.fbi.gov/ucr/cius2007/offenses/property_crime/burglary.html.

[3] http://www.tinyos.net/.

[4] Live view gps asset tracker. http://www.liveviewgps.com/all+gps+tracking+products.html.

[5] 802.15.4: Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE P802.15* (2003).

[6] AYER, S., SCHROETER, P., AND BIGN, J. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *European Conference on Computer Vision* (1994), pp. 316–327.

[7] BHANAGE, G. D., ZHANG, Y., ZHANG, Y., TRAPPE, W., AND HOWARD, E. Rollcall : The design

for a low-cost and power efficient active rfid asset tracking system. 2521–2528.

[8] BLACK, M. J., AND ANANDAN, P. The robust estimation of multiple motions: Affine and piecewise smooth flow fields. *Computer Vision and Image Understanding* (1996).

[9] BOPPANA, R. B., AND HALLDRSSON, M. M. Approximating maximum independent sets by excluding subgraphs. In *Scandinavian Workshop on Algorithm Theory* (1990), pp. 13–25.

[10] CAO, Z., DIAO, Y., AND SHENOY, P. Architectural considerations for distributed rfid tracking and monitoring. NetDB 2009.

[11] CASARES, M., VELIPASALAR, S., AND PINTO, A. Light-weight salient foreground detection for embedded smart cameras. *Computer Vision and Image Understanding 114*, 11 (2010), 1223–1237.

[12] COMANICIU, D., RAMESH, V., AND MEER, P. Real-time tracking of non-rigid objects using mean shift. pp. 142–149.

[13] FEIGE, U. Approximating maximum clique by removing subgraphs. *Siam Journal on Discrete Mathematics 18* (2004), 219–225.

[14] GAY, D., LEWIS, P., BEHREN, R., WELSH, M., BREWER, E., AND CULLER, D. The nesC language: A holistic approach to network embedded systems. *Proc. ACM SIGPLAN Conf. Program. Language Des. Implementation* (Jun. 2003).

[15] GUHA, S., PLARRE, K., LISSNER, D., MITRA, S., KRISHNA, B., DUTTA, P., AND KUMAR, S. AutoWitness: locating and tracking stolen property while tolerating GPS and radio outages. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems* (2010), ACM, pp. 29–42.

[16] GUHA, S., PLARRE, K., LISSNER, D., MITRA, S., KRISHNA, B., DUTTA, P., AND KUMAR, S. Autowitness: Locating and tracking stolen property while tolerating gps and radio outages. In *ACM Sensys* (2010).

[17] HORPRASERT, T., HARWOOD, D., AND DAVIS, L. S. A statistical approach for real-time robust background subtraction and shadow detection. *EEE ICCV'99 Frame-Rate Workshop* (1999).

[18] IRANI, M., ANANDAN, P., BERGEN, J., KUMAR, R., AND HSU, S. Efficient representations of video sequences and their applications. *Signal Processing-image Communication 8* (1996), 327–351.

[19] KANG, J., COHEN, I., AND MEDIONI, G. Multi-views tracking within and across uncalibrated camera streams. In *First ACM SIGMM international workshop on Video surveillance* (2003), ACM, pp. 21–33.

[20] KANG, J., COHEN, I., AND MEDIONI, G. G. Continuous tracking within and across camera streams. In *Computer Vision and Pattern Recognition* (2003), pp. 267–272.

[21] LIANG CHEN, J., CHIAO CHEN, M., WU CHEN, C., AND CHUNG CHANG, Y. Architecture design and performance evaluation of rfid object tracking systems. *Computer Communications 30* (2007), 2070–2086.

[22] MITTAL, A., AND DAVIS, L. S. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In *European Conference on Computer Vision* (2002), pp. 18–36.

[23] ORWELL, J., REMAGNINO, P., AND JONES, G. A. Multi-camera color tracking. In *Computer Vision and Pattern Recognition* (1999).

[24] POLASTRE, J., R., S., AND D., C. Telos: Enabling ultra-low power wireless research. *Proceedings of the 4th international symposium on Information processing in sensor networks* (2005).

[25] SENEM, V., JASON, S., CHENG-YAO, C., WAYNE, H., ET AL. A Scalable Clustered Camera System for Multiple Object Tracking. *EURASIP Journal on Image and Video Processing 2008* (2008).

[26] SENIOR, A., HAMPAPURA, A., TIANA, Y.-L., BROWNA, L., PANKANTIA, S., AND BOLLE, R. Appearance models for occlusion handling. *Image and Vision Computing 24*, 11 (Nov. 2006), 1233–1243.

[27] SHRIVASTAVA, N. Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms. In *in SenSys'06: Proc. 4th Internat. Conf. on Embedded Networked Sensor Systems, 2006* (2006), ACM Press, pp. 251–264.

[28] SINGH, J., AND MADHOW, U. Tracking multiple targets using binary proximity sensors. In *In Proc. Information Processing in Sensor Networks* (2007), ACM Press, pp. 529–538.

[29] STEIN, G. P. Tracking from multiple view points: Self-calibration of space and time. In *Computer Vision and Pattern Recognition* (1999), pp. 1521–1527.

[30] ZHANG, H., AND MALIK, J. Learning a discriminative classifier using shape context distance. In *Proc. of the IEEE CVPR* (2003), IEEE, pp. 242–24.