Privacy Protecting Data Collection in Media Spaces

Jehan Wickramasuriya, Mahesh Datt, Sharad Mehrotra & Nalini Venkatasubramanian
Department of Information & Computer Science
University of California, Irvine
Irvine, CA 92697-3425, USA
{jwickram,mahesh,sharad,nalini}@ics.uci.edu

ABSTRACT

Around the world as both crime and technology become more prevalent, officials find themselves relying more and more on video surveillance as a cure-all in the name of public safety. Used properly, video cameras help expose wrongdoing but typically come at the cost of privacy to those not involved in any maleficent activity. What if we could design intelligent systems that are more selective in what video they capture, and focus on anomalous events while protecting the privacy of authorized personnel? This paper proposes a novel way of combining sensor technology with traditional video surveillance in building a privacy protecting framework that exploits the strengths of these modalities and complements their individual limitations. Our fully functional system utilizes off the shelf sensor hardware (i.e. RFID, motion detection) for localization, and combines this with a XML-based policy framework for access control to determine violations within the space. This information is fused with video surveillance streams in order to make decisions about how to display the individuals being surveilled. To achieve this, we have implemented several video masking techniques that correspond to varying user privacy levels. These results were achievable in real-time at acceptable frame rates, while meeting our requirements for privacy preservation.

Categories and Subject Descriptors: I.4.8 [Image Processing & Computer Vision]: Scene Analysis – *Object recognition, Sensor fusion, Tracking*; K.6.5 [Management of Computing & Information Systems]: Security and Protection

General Terms: Design, Security

Keywords: Video Surveillance, Privacy, Access Control

1. INTRODUCTION

With the heightened consciousness among the public, private and government organizations for security, surveillance technologies (especially video surveillance) have recently received a lot of attention. Video surveillance systems are being considered/deployed in a variety of public spaces such as metro stations, airports, shipping docks, etc. As cameras go up in more places, so do the concerns about invasion of privacy [14]. Privacy advocates worry whether the potential abuses of video surveillance outweigh its benefits. A fundamental challenge is to design surveillance systems that serve the security needs while at the same time protect the privacy of the individuals.

In this paper, we describe the design of a privacy preserving video surveillance system that monitors subjects in an instrumented space only when they are involved in an access violation (e.g., unauthorized entry to a region). In our system, access control policies specify the access rights of individuals to different regions of the monitored space. Policy violations (detected via use of localization sensors such as RFID tags[1], motion detection, etc) are used to trigger the video surveillance subsystem. Video manipulation techniques such as masking are used to preserve the privacy of authorized subjects when the surveillance system is turned on. To the best of our knowledge, the proposed system is the first of its kind that fuses information from various sensors with video information in implementing a privacy-protecting surveillance framework for media spaces.

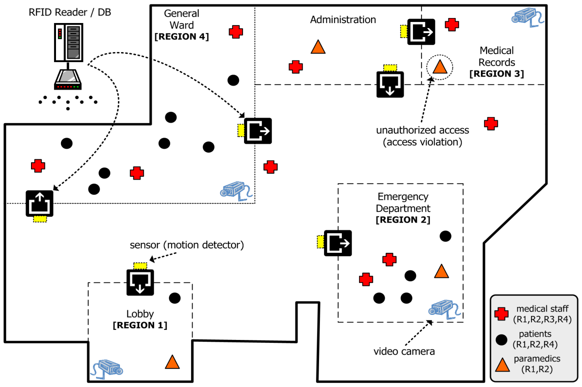

Figure 1: Instrumentation of an hospital's physical space for privacy protecting video surveillance.

Example Scenario: We demonstrate the basic functional-

MM'04, October 10-16, 2004, New York, New York, USA.
Copyright 2004 ACM 1-58113-893-8/04/0010 ...$5.00.

[1] A RFID (Radio-Frequency IDentification) tag is a tiny, relatively inexpensive device capable of transmitting a piece of static information across a distance. RFID tags are currently in use for mediating access to various regions, however it does not provide enough information to pinpoint the object being tracked within that space.

ity of our proposed framework by means of an example that examines security in a hospital setting. Assume the hospital is divided into federated regions which are each covered by video cameras. The RFID sensor information is used in conjunction with the video surveillance subsystem to provide coverage of the monitored regions. Furthermore, access control policies are defined for personnel carrying RFID tags. The enforcement of these policies are used to influence the video surveillance system. Fig. 1 shows a scenario where the hospital consists of four regions, each surveilled by a single camera.

The idea is that if a subject is authorized to be in a particular region (which is determined by their RFID tag) he/she may be hidden in the video. This way, subjects' privacy is maintained until they violate their given rights (e.g. enter a region they are not authorized to be in). In Fig. 1, each region ({R1,R2,R3,R4}) is monitored by a video feed which is triggered by a system that processes the information from RFID readers and predefined access policies. A paramedic entering R3 causes the motion sensor to initiate a read on his/her tag which is forwarded to the RFID reader. This causes an access violation and subsequently triggers the video feed for R3. However, medical personnel present in R3 (who are authorized for all regions) will be masked in the feed according to their given privacy level. Additional constraints can also be enforced such as restricting certain patient/doctor pairs or associating a bed (which can also be tagged with RFID) with a particular patient).

Primary contributions: Our primary contribution in this paper is a novel approach to combining localization sensors (in particular RFID technology) with traditional video surveillance to build a framework for privacy-protecting data collection in media spaces. We have built and tested such a surveillance system based on ideas discussed in this paper over the past few months. Our system consists of (1) localization component that utilizes off-the-shelf sensor hardware (i.e RFID, motion detection) to determine location of subjects; (2) policy framework that supports access control specification using XML; (3) a video processing component that implements several video processing techniques for motion detection and object masking that can be applied in real time at interactive frame rates. The above components are integrated to realize a fully implemented video surveillance system that autonomously detects anomalous events while protecting the privacy of authorized personnel who may appear in these video streams.

The remainder of this paper is organized as follows; Section 2 describes our system architecture and outlines the role of RFID and video processing techniques in realizing this framework. In Section 3 we introduce our XML-based policy framework for access control. Section 4 discusses privacy protecting video processing techniques. Section 5 describes our implementation of the framework and presents accompanying results. In Section 6 we discuss related work and conclude with future work in Section 7.

2. SYSTEM ARCHITECTURE

Fig. 2 depicts a high-level outline of the system architecture. The infrastructure comprises of the following components:

- **Sensing Module:** Processes data from incoming sensors. More specifically, a RFID control component deals with RF-related messaging arriving at the readers. Data from motion detection sensors are also processed here. A video input module handles the incoming video stream data from the various surveillance cameras.
- **Data Management Module:** Consists of a XML-based policy engine for access control. This policy engine interacts with a database subsystem consisting of profile and policy databases for the users.
- **Auxiliary Services:** A service library contains modules that provide auxiliary services on the sensed information (including the incoming video stream(s)). These include obfuscation, motion detection and object tracking modules. For example masking may be applied to the video stream before it is passed to the output module, if the subject has been authorized by the policy engine.
- **Output Module:** Handles customized reporting, logging and video rendering functionality.

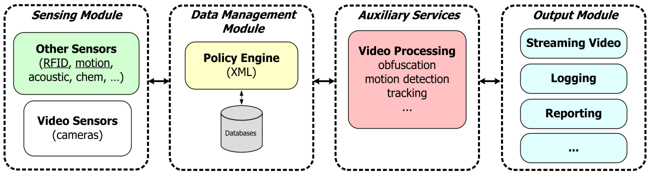

Figure 2: System architecture.

In particular, we utilize RFID sensor information (together with motion detection sensors) for localization of objects within the media space.

Radio Frequency Identification (RFID) Technology: RFID technology provides the ability to interrogate data content without contact and the necessity of the line-of-sight communication[2]. RFID is a means of storing and retrieving data through electromagnetic transmission to a RF compatible integrated circuit [12]. A typical system consists of a tag (or a set of tags), a reader/receiver that can read and/or write data to these tags, and optionally a field generator (for relaying information from a region). The communication occurs at a pre-defined radio frequency and utilizes a protocol to read and receive data from the tags via *inductive coupling.* Our system is instrumented as follows; each protected region is equipped with a field generator. The boundaries between adjacent spaces can either be physical, as in a door or wall separating two rooms, or virtual, as in a non-physical partition used to separate parts of a large room. Since we are interested in entry (and exit) to a region, each field generator is equipped with a motion detector which triggers a read of the region when motion is detected. If there is no tag information associated with the motion, the signal sent to the reader is categorized as unauthorized and video surveillance of the region is triggered. Tags are distributed to personnel, and a database stores the access rights associated with each tag, desired user privacy levels (which are subsequently mapped to video masking techniques). When entry into a region is detected, the tag information is read (if present) and that information is passed to the RFID control module which forwards an authorization request to the policy engine. The policy decision for the object is then passed to

[2] Thanks to recent manufacturing methods, estimates suggest that the cost of these tags will drop to the vicinity of five cents per unit by 2005 [15] making them a very attractive solution.

the video processing module, which uses this information in rendering the video object.

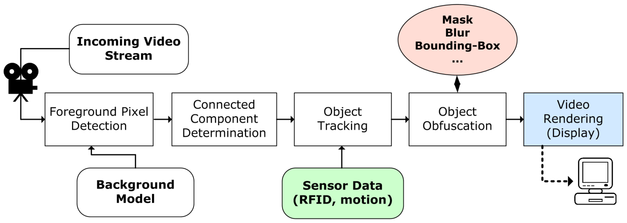
Incoming Video Stream
Mask
Blur
Bounding-Box
...
Foreground Pixel Detection
Connected Component Determination
Object Tracking
Object Obfuscation
Video Rendering (Display)
Background Model
Sensor Data (RFID, motion)

Figure 3: Flow chart of the video subsystem.

Video Processing Subsystem: A flow chart identifying the main components of the current video processing subsystem is depicted in Fig. 3. Our video analysis software is based on detection of moving foreground objects in a static background. The choice of a relatively simple algorithm is motivated by the need for real-time processing. Using a background model, the pixel detection module passes the result to a simple 4-connected component algorithm to cluster pixels into blobs. The object tracker identifies objects entering the field of view of the camera and localizes them in the video frame using information passed to it by the RFID control module. Depending on access control policy, the stream is further processed using one of a number of masking techniques (Fig. 4) before being displayed by the rendering module. This entire process is tuned to maintain a frame rate of approximately 30fps. It should be noted that the video camera capture facility is motion triggered. This way, video is only captured when there is activity in the region being monitored making it much easier to focus on events of interest and save storage if video data is being archived.

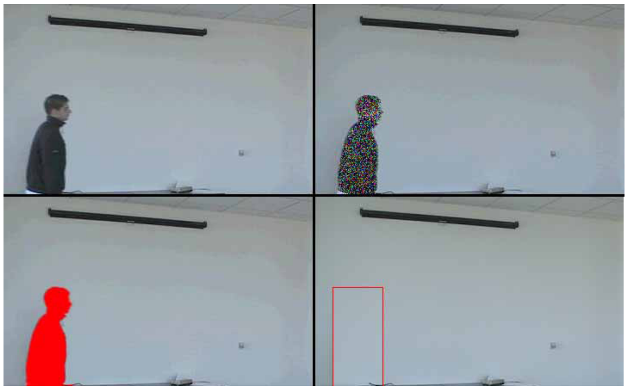

Figure 4: Privacy-protecting masking techniques for video utilized by our system. They represent different levels of user privacy. A frame of the original video is shown (top-left), followed by a noise/blur filter (top-right). A pixel-coloring approach is shown (bottom-left) followed by a bounding-box based technique (bottom-right) which hides details such as gender, race or even dimensions of the person in question.

Further details about the subsequent components are provided in the following sections.

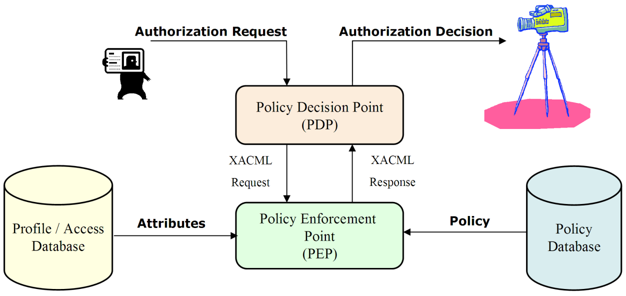

Figure 5: XACML-based Policy Framework.

```
<!−− Access request by tag 1001 −−>
<Subject>
 <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:
 subject:subjectid"
 DataType="urn:oasis:names:tc:xacml:1.0:data−type:
 rfc822Name">
   <AttributeValue>tag1201</AttributeValue>
 </Attribute>
</Subject>
```

Figure 6: Subject specification in XACML.

3. SPECIFICATION OF ACCESS CONTROL POLICY

Here we present an approach for specifying and enforcing security policies in the context of our architecture. The access control model is crucial to our system and allows specification of spatial constraints (i.e. which regions a person has access to). Our access control policy allows the implementor to specify policies which dictate the manner in which video surveillance is conducted in a physical space. In other words, the policy decisions drive the video subsystem. For example, a member of the janitorial staff cleaning an office suite at 2 A.M. might be considered 'normal'. However, if corporate policy prohibits entry to particular parts of the building after midnight, this event may be considered a potential security breach and need to be further investigated.

We specify security policies using the eXtensible Access Control Markup Language (XACML), which are processed by an enforcement engine which provides mediated access to a database. XACML [6] is utilized to define the access policies as well as carry out enforcement on these policies. XACML is a standard, general purpose access control policy language defined using XML. It is flexible enough to accommodate most system needs, so it may serve as a single interface to policies for multiple applications and environments. In addition to defining a policy language, XACML also specifies a request and response format for authorization decision requests, semantics for determining policy applicability and so on. The components of the access control model are the video-based objects, the potential users, and modes of access which can be modeled as a traditional authorization rule of the form $< s, o, m >$, where subject s is authorized to access object o under mode m, where the mode is associated with a particular privacy level. In the following section we give a general description of the type of policies supported by our system and then give specific examples of their specification in XACML (A simple example of subject specification is shown in Fig. 6)[3].

[3] In reality, any type of policy specification framework can be implemented and applied to our general set of primitives.

Access Control Framework: Here we present a framework for access control specification in the context of our system. For the purposes of this paper, we outline a simplified specification of the framework. Assume a set of regions R = { r_1, ... , r_n } which are monitored over time, a set of corresponding video streams, V = { v_1, ..., v_n } to these regions. Assume that the region r_i corresponds to the video stream v_i. But note that in a general setting more than one video stream can correspond to a region. There is a set of objects O = { o_1,....,o_m }, which are being surveilled (e.g. people). These objects can be static (e.g. inventory) or mobile. Each of these objects may have a RFID tag associated with it, which in effect serves as its credential and basis for authorization to a particular region. The mapping between a particular tag and the corresponding object is stored in a profile database. We specify a set of tags, T = { t_1, ..., t_x } and use T as the subject of the authorization rules. Furthermore, each tag also has associated with it a privacy level P, where P $\in$ L = $\{L_0, ..., L_N\}$ (Fig. 4). These privacy levels specify the sensitivity at which video information pertaining to a particular object should be collected or viewed. The privacy level of a tag is determined at the time the tag is issued and can be modified later by an administrator. In our system, we use four levels of increasing sensitivity (L_0 corresponds to no privacy and L_3 to total privacy) and map a masking technique to each of these levels. The various video masking techniques are discussed in the next section. Finally, to aid in specifying group-based authorization rules, we introduce an object association list, OA $\subseteq$ O. This is a useful concept, not in the traditional notion of access groups but for associating inventory (i.e. equipment etc.) with users authorized to use them. We use the more general notion here that members of OA are simply other objects, however in practice these objects are of a static nature.

Therefore we can associate with a tag, t_i (and therefore an object) a set $Rules := (Ruleset, d)$ which contains a *Ruleset* of authorization rules that allow or deny an action as well as an element d that defines the default ruling of this rule set. Default rulings can be one of $\{+, -, \emptyset\}$, corresponding to 'allow', 'deny' and 'don't care'. An $AR_i \in Ruleset$ is a 5-tuple of the form $< r, p, oa, ts, res >$, where r $\in$ R, p $\in$ L, oa $\subseteq$ OA, ts is a temporal constraint (e.g. a time range, $[ts_{min}, ts_{max}]$ corresponding to the validity of the rule) and res $\in \{+, -\}$ corresponding to 'allow' or 'deny'. A 'deny' result would imply that the object in the corresponding video stream would be unaltered and shown (p = L_0), whereas 'allow' would cause the stream to be altered so as to protect the identity of the object in question, depending on p. Additionally, authorization rules are subject to a *deny-takes-precedence* evaluation strategy. That is if an AR_i, $< r, p, oa, ts, - > \in AR'$ exists, all tuples $< r, p, oa, ts, + > \in AR'$ are removed.

Some typical examples are given below, following a generic authorization request of the form $(tag, region, timestamp)$, which specifies the tag in question, the region it has entered, as well as the time of the request.

1. **Person with no tag(s)**, $(\emptyset, r_i, ts)$: A person entering a region with no tag, will not have any rules associated with him/her aside from the default ruling, which will be 'deny'. This corresponds to a privacy level of L_0, meaning the person will be shown.
2. **Person with a valid tag**, (t_i, r_i, ts): A person entering a region with a valid tag, will satisfy spatial and temporal constraints and return an 'allow' decision together with the corresponding privacy level, p. $\forall AR,\ t_i \rightarrow AR_i, \exists AR_i\ such\ that\ r_i = r,\ \wedge\ (ts_{min} \leq ts \leq ts_{max})\ \wedge\ (res_i = +)$.

```
<!-- Only allow entry into region from 8am to 1pm -->
<Condition FunctionId=”urn:oasis:names:tc:xacml:1.0:function:and”>
 <Apply FunctionId=”urn:oasis:names:tc:xacml:1.0:function:
 time-greater-than-or-equal”
  <Apply FunctionId=”urn:oasis:names:tc:xacml:1.0:function:
  time-one-and-only”>
   <EnvironmentAttributeSelector DataType=”http://www.w3.org/2001/
   XMLSchema#time”
     AttributeId=”urn:oasis:names:tc:xacml:1.0:environment:
     current-time”/>
  </Apply>
   <AttributeValue DataType=”http://www.w3.org/2001/
   XMLSchema#time”>08:00:00</AttributeValue>
 </Apply>
 <Apply FunctionId=”urn:oasis:names:tc:xacml:1.0:function:
 time-less-than-or-equal”
 <Apply FunctionId=”urn:oasis:names:tc:xacml:1.0:function:
 time-one-and-only”>
  <EnvironmentAttributeSelector DataType=”http://www.w3.org/2001/
  XMLSchema#time”
    AttributeId=”urn:oasis:names:tc:xacml:1.0:environment:current-time”/>
 </Apply>
 <AttributeValue DataType=”http://www.w3.org/2001/
 XMLSchema#time”>13:00:00</AttributeValue>
 </Apply>
</Condition>
```

Figure 7: An example of a time-bounded authorization condition specified in XACML.

3. **Person with an invalid tag**: Assuming the tag has been successfully authenticated, two possible violations may cause a 'deny' result from an authorization rule. (1) The access rights associated with the current tag specify that the requesting region is unauthorized, causing a *spatial access violation*. (2) The access rights associated with the current tag specify that the timestamp associated with the request does not satisfy the time bounds associated with the requesting region [4]. Fig. 7 shows an authorization rule that enforces a time constraint on a region expressed in XACML.
4. **Group-based authorization**: Here we associate a temporal threshold δ, with each tag request (i.e. entry into a surveilled region). If multiple tag events are detected within this threshold, they are treated as a group authorization request. In which case, the respective object association lists are cross-checked.

4. PRIVACY PROTECTING VIDEO PROCESSING TECHNIQUES

Our video subsystem (Fig. 3) relies on techniques that detect moving foreground objects from a static background in an indoor setting. When the video camera is turned on, the initialization process learns the background for a scene (approximately 100 frames). Each background pixel is modeled as one Gaussian distribution with two elements $[E, \sigma_i]$ defined as follows:

$$E_i = [\mu_{R(i)}, \mu_{G(i)}, \mu_{B(i)}]$$
$$\sigma_i = [\sigma_{R(i)}, \sigma_{G(i)}, \sigma_{B(i)}]$$

where $\mu_{R(i)}, \mu_{G(i)}, \mu_{B(i)}$ and $\sigma_{R(i)}, \sigma_{G(i)}, \sigma_{B(i)}$ are the arithmetic means and standard deviations of the i^{th} pixel's red, green and blue channels respectively, computed over N still

[4] We adopt two possible approaches to handle a violation of this manner. Either the person is immediately unmasked in the current region, or remains masked until subsequent reentry into the region causes reevaluation of the associated access rights.

background frames. The processed mean and standard deviation images are stored in main memory, after which each incoming frame is grabbed and goes through pixel level analysis to distinguish moving foreground pixels from the background. The selection process is illustrated in Algorithm 1. TCD is the color distortion factor computed for each pixel channel as μ +/− (3*σ).

Algorithm 1 : The Pixel Selection Process

```
for each (i$^{th}$) pixel in the current frame do
    for each R,G,B channel do
        if one of i$^{th}$ pixel channel value is NOT within the TCD then
            select pixel as foreground
        end if
    end for
end for
```

Once the foreground pixels are identified from the background, we pass it to a simple 4-connected component algorithm [8] to cluster pixels into blobs in one pass (from bottom left to top right). We use a threshold to discard blobs with few numbers of pixels (these are considered noise). Empirical analysis was carried out on a number of video clips (for our setting) to determine appropriate threshold values for this purpose (discussed in further detail in Section 5.2). For this application, the threshold was set to the minimum number of pixels a person's body can occupy in a 320x240 image. In addition, attributes such as the center of gravity, maximum, minimum x and y range, and the area in pixels are assigned to each blob.

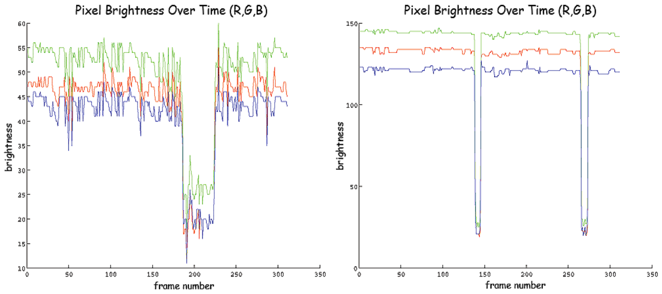

Figure 8: Pixel brightness over time for a sampled point in a scene where 1) one person enters [left]; 2) two people enter [right]. In 1), the pixel of interest is against a non-static background (note the erratic variation in brightness). It can also be seen that the person spends more time at the sampled point. In 2) it can be seen that the background is less dynamic.

Object Tracking: Our tracker maintains a list of objects present in the scene. Each object (person) has the following parameters which are updated for each frame; 1) Minimum and maximum x and y pixel range to identify a bounding box around each person; 2) The area a person occupies in the frame (in pixels) to determine if the object is legitimate or simply noise; 3) The person's center of gravity (cog); 4) A velocity vector for a person; 5) The future cog, which is computed as follows (where V is the velocity vector, and $disTime$ is the displacement time for the object);

$$futurecog_x = cog_x + (V_x * disTime)$$
$$futurecog_y = cog_y + (V_y * disTime)$$

6) The current frame number. This is used to identify whether a person in the list is present in the scene, and accordingly whether they will be retained or removed from the object list; 7) A merge list to indicate whether a person has merged with other persons. Two people in the scene are determined to have merged if they point to the same blob in the current frame. Splitting occurs when the same two merged people in the previous frame point to two different blobs in the current frame, upon which the merge list is updated.

After each frame gets processed into legitimate blobs, the tracker tries to match each person in the list with a candidate blob in the incoming frame using motion vector prediction [11]. In essence if a person's cog falls within one of the candidate blobs' bounding box in the current frame, then the information is assigned to that particular blob (cog, min/max parameters, area etc.). If a match is not found between an object and all the blobs in the current frame, the object is removed from the list and assumed to have exited the scene. If a new blob does not have a match in the object list, a new object is created and assigned authorization information (masking level) by the RFID subsystem. An outline of the algorithm is shown in Algorithm 2. During the rendering process, the blob's corresponding masking level is applied to the outgoing video stream. In the case of a *merge*, the highest masking level among the persons involved in the merge is applied to the corresponding blob. For example if a person p_1, with privacy level L_1 merges with a person p_2, with privacy level L_2, then the merged blob is masked using L_2.

Algorithm 2 : The Tracking Algorithm

```
$peopleList = \emptyset$
for each incoming frame, $f$ do
    for each candidate blob, $b$ in $f$ do
        for each person, $p \in peopleList$ do
            $findMatch(p,b)$
        end for
        if match is not found for $b$ then
            create a person object $p$ corresponding to $b$
            $privacyLevel := fuseRFID(p)$
            $p.maskingLevel := privacyLevel$
            $peopleList.add(p)$
        end if
        $p.frameNumber := f$
    end for
    for person, $p \in peopleList$ do
        if $p.frameNumber$ != $f$ then
            $peopleList.remove(p)$
        end if
    end for
    for persons, $p$ and $q \in peopleList$ do
        if $p$ and $q$ point to the same blob then
            add $p$ to $q.mergeList$
        else if $p \in q.mergeList$ then
            remove $p$ from $q.mergeList$
        end if
    end for
end for
```

5. IMPLEMENTATION & RESULTS

In this section we describe the experimental setup and hardware used, further implementation details as well as results which illustrate the functionality of our video processing subsystem.

5.1 Experimental Setup

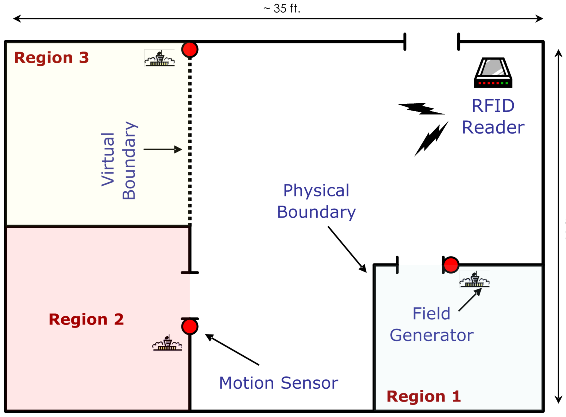

Figure 10: Experimental setup for RFID equipment.

For the experiments described here we used Canon Optura 20 DV Cameras, capturing video at a resolution of 320x240. One RFID reader, tags, and field generator relays were used for the instrumented regions. All processing was carried out on a uniprocessor Pentium 4 at 2.60 Ghz with 1 GB of RAM equipped with a Pinnacle firewire video capture card under Windows XP Professional. Microsoft's DirectX SDK (8.0) was utilized for interacting with the camera. The RFID reader was connected to the computer via a four-wire RS-232 interface, and we used medium range RFID tags that transmit and receive at 916.5 Mhz and 433 Mhz respectively. The TrakTags utilized in the experiments have a range of approximately 275ft (85m) and the field generators possess a wake-up range of between 6.5-108ft (2-33m). The instrumented space is illustrated in Fig. 10. The range of the field generator relays are hardware adjustable, and were calibrated to detect activity in the regions of interest. We outfitted an office workspace by creating two regions with physical boundaries (rooms) and one open area which was partitioned by one of the relays via a virtual boundary. Motion sensors were used to monitor activity in and out of the chosen regions. As the motion detector senses movement, it wakes up the field generator which transmits a constant 433 MHz field to detect any tags in the region. This information is sent to the reader. Field generators and motion sensors were positioned at the entry points to the regions being studied. Each region was monitored by a single camera for the purpose of these experiments.

The RFID event handling process involves detection of a tag in a region, checking the associated access control policy and relaying the result to the video subsystem. On detection of a tag, the tag ID, reporting field generator ID (effectively the zone identification) and timestamp is obtained. A separate thread in the video process waits for this information on a socket and depending on the policy decision, renders the object (which may be masked). Fig. 11 depicts the high-level RFID event handling procedure for (non-group) events.

```
long TagEvent(long status, HANDLE funcId,
              rfTagEvent_t* tagEvent, void* userArg)
{
  if (tagEvent−>eventType == RF_TAG_DETECTED) {
    if (PolicyCheck(GenerateXACMLRequest
    (tagEvent−>tag−>id,tagEvent−>fGenerator,enterTime)))
      signalVideoThread(videoSocket, szBuf);
  }
}
```

Figure 11: RFID event handling.

5.2 Results & Observations

We demonstrated the functionality of our framework by testing a variety of cases. In the context of each of the scenarios outlined in Section 3, we also tested the functionality of our tracking and masking algorithms. Interaction between multiple people with varying access rights were tested, especially merging and splitting between masked and unmasked objects. Fig. 9 illustrates the interaction between authorized and unauthorized people in video. Here we can see the merging/splitting process as they pass each other in view of the camera. In evaluating the performance of our implementation, we investigated the overheads involved in our techniques. We observed that for a single camera the overhead of the various masking techniques (shown in Fig. 4) were negligible due to the limited field of view of the camera and hence the number of people interacting simultaneously at any given time. We chose a resolution of 320x240 as it strikes a good balance between detail and performance (both in terms of framerate and storage space if the video was being archived). Doubling the resolution to 640x480 affected the framerate significantly as expected, as on average just under 20fps was achieved. At our target resolution of 320x240, we achieved a constant framerate of 30fps in all tested cases.

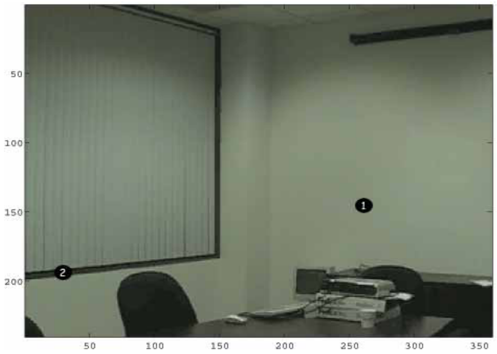

Figure 12: Example scene used for analysis.

We carried out some video analysis prior to implementing our video subsystem. The scene depicted in Fig. 12 was used for this analysis. The scenario consisted of one person entering the scene from the right and moving through the room to point 2 as shown in the figure. As outlined in Section 4, we utilize a pixel selection process to distinguish moving foreground pixels from the background, and model each background pixel as a Gaussian distribution. Fig. 13 shows this distribution for the two areas highlighted in Fig. 12. In each case, the pixels corresponding to the person entering the room is highlighted. The other set of pixels (the domi-

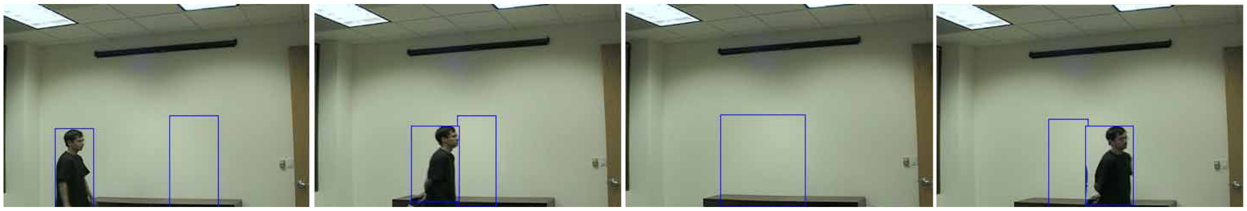

Figure 9: Interaction between authorized and unauthorized personnel in the space

nant peak) represents the background pixels. It can be seen for point 2, the foreground pixels (i.e. the clothing worn by the person in the scene) are very close to the sample point (window sill). Hence, the threshold used (in this case 3*σ) becomes very important in distinguishing the two. We are developing more advanced techniques to make the pixel detection process more robust.

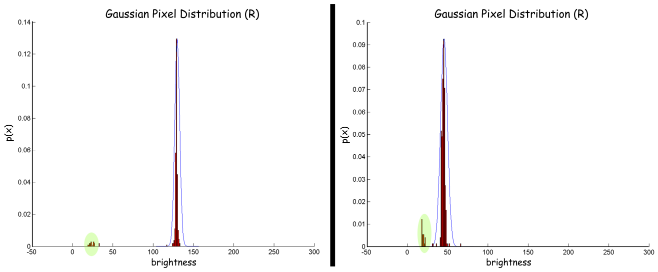

Figure 13: Gaussian pixel distribution (red channel only) for point 1 (left) and point 2 (right) in Fig. 11. Here a person enters the scene and moves from point 1 to 2. The pixel distribution for the two points in the video stream are shown.

System Deployment: Even though we have realized a fully-functional implementation of our framework, the deployment of a such a system should eventually be pushed to the camera level. In our implementation, we tightly coupled the processing capabilities (of the PC) to the camera and processed everything in real-time with no archival of the original video data (only the processed video stream is available). Ideally this processing capability should reside at the camera level to make the privacy preservation in media spaces more acceptable to the end-users. Optimization of the algorithms used here for object tracking and masking for privacy is a key component of such a realization as well as the possible use of MPEG-4 video. MPEG-4 has a superior compression efficiency, advanced error control, object based functionality and fine grain scalability making it highly suitable for streaming video applications. Recently MPEG-4 has emerged as a potential front runner for surveillance applications because of its layered representation, which is a natural fit for surveillance tasks as they are inherently object-based. It is also desirable to find people moving in the scene, independent of the background. We are also conducting scalability analysis on our techniques for larger regions.

6. RELATED WORK

Recently, there has been a increased interest in RFID-related research, both in the academic and commercial sectors. Particularly, solutions examining the threat to consumer privacy of RFID technology have proposed techniques to protect unwanted scanning of RFID tags attached to items consumers may be carrying or wearing. For example, [10] propose the use of 'selecting blocking' by 'blocker tags' to protect consumer privacy threatened by the pervasive use of RFID tags on consumer products. This enables consumers to 'hide' or 'reveal' certain RFID tags from scanning when they want to. In this paper, we use RFID technology for localization of subjects appearing in media spaces. Therefore, this research is extremely useful and complementary to our infrastructure as it addresses security concerns relating to the RFID hardware itself.

Privacy concerns in video surveillance have not really been addressed in video processing research. Furthermore, these techniques require efficient implementations to process real-time video streams (usually MPEG-1 or MPEG-2). Variations of background subtraction has been used as a technique for foreground/background segmentation for long video sequences [7]. As a relatively simple method, it works fairly well in most cases but its performance depends heavily on the accuracy of the background estimation algorithms. We utilize a form of background subtraction for our video subsystem but it is not the focus of the work here. Issues dealing with illumination changes, shadows, dynamic backgrounds are not addressed here, but have been investigated in other work. The addition of these techniques will serve to make our video subsystem more robust. Relevant areas in real-time motion tracking, image/face recognition [20] and video data indexing have been studied but rarely [19] infused with techniques to preserve privacy. [13] proposed a quasi-automatic video surveillance approach based on event triggers to generate alarms and overcome the drawbacks of traditional systems. We are adopting a similar vision in providing surveillance only when activity is taking place within a region. A new approach known as experimental sampling was proposed in [18] which carries out analysis on the environment and selects data of interest while discarding the irrelevant data. Our framework utilizes different modalities of sensor data in providing information that assist the video subsystem in detecting and classifying anomalies.

The area of work dealing with privacy preservation in media spaces is relatively new, and a lot of the related work is in the domain of computer-supported corporative work (CSCW) [5, 19]. Of particular interest is the work presented by Boyle et. al [2] which utilized blur and pixelization filters to mask sensitive details in video while still providing a low-fidelity overview useful for awareness. Specifically they analyze how blur and pixelize video filters impact both awareness and privacy in a media space. However, the limitation of these techniques are that the filters are not applied to the individual objects in the video but to the entire video frame, which makes enforcing separate policies and distinguishing between authorized and unauthorized personnel impossible. In our approach, we apply the processing techniques on a per-object basis and apply this effect in real-time as the object is tracked in the space. Previous work utilizing eigenspace filters [3] proposed a way to mask out potentially sensitive action associated with an individual by using a set of pre-defined base images to extract a representation of the person (face) by taking an inner product of the video images with those base images stored in a database. This technique though useful, relies on capturing and storing base images of the potential subjects, which may be both infeasible as well as against the notion of trying to store as little identifiable information about individuals in the space as possible.

There has been a large body of work on policy specification and access control in XML [1, 4]. The majority provide run-time checking of access control policies and fine-grained specification of policy. We utilize these techniques, and adopt a XACML type architecture [6] to specify and enforce access control in our system. XACML is an attractive option that presents standardized interfaces for request/response formats and can describe an access control policy across many applications. Additionally, a single XACML policy can be applied to many resources. This helps avoid inconsistencies and eliminates duplication of effort in creating policies for different resources.

7. FUTURE WORK & CONCLUSIONS

In this paper, we address the challenge of providing a framework for privacy-protecting data collection in video spaces. We designed and implemented a fully functional system, and were able to achieve real-time results at an acceptable frame rate while achieving our goals for privacy protection of users. With this foundation in place, there are a number of improvements and extensions to the system some of which are currently being worked on. We are building support for additional types of sensor data (e.g. acoustic sensors) and expanding the policy framework to capture more complex policies that deal with additional attributes and information from other types of sensors.

Further optimization of the video processing techniques used in our framework are also areas for future research. For this application, our experimental setting was an indoor space. However, this problem becomes much more complex in an open, outdoor environment where it is very difficult to define the background as it can change very frequently. Additionally, the number of objects in view may be significantly greater so scalability becomes a concern. The notion of fusing together multiple cameras in gathering information about a region is also of interest in such a setting. More specific enhancements to the system can be made at the pixel level by using a mixture of Gaussian distributions [16] to deal with lighting changes, long term scene change and ambient motion. For example, dealing with static objects in scenes where they becomes part of the background and other moving objects like window shutters etc. Combining motion prediction with histogram-based template matching [17] of a person's blob-color can improve accuracy significantly in tracking people, especially when they merge and split. Additionally, the security and privacy concerns associated with RFID hardware itself is a concern which is beyond the scope of this paper. However, as RFID technology evolves and tags have the ability to perform some level of computation, authentication schemes utilizing low-cost cryptographic techniques [9] can be utilized.

Acknowledgements

Support of this research by the National Science Foundation under Award Numbers 0331707 and 0331690 is gratefully acknowledged. We would also thank Mohanned Alhazza for his help in implementing the video processing component of the infrastructure.

8. REFERENCES

[1] Bertino, E., Castano, S., Ferrari, E., and Mesiti, M. Controlled Access and Dissemination of XML Documents. *In WIDM'99 ACM* (1999).

[2] Boyle, M., Edwards, C., and Greenberg, S. The Effects of Filtered Video on Awareness and Privacy. *In Proc. of CSCW'00* (2000), pp. 1–10.

[3] Crowley, J., Coutaz, J., and Berard, F. Things That See. *In Communications of the ACM, 43:3 (March)* (2000), ACM Press, New York, NY, pp. 54–64.

[4] Damiani, E., and di Vimercati et. al, S. A Fine-Grained Access Control System for XML Documents. *In ACM Transactions on Information and System Security (TISSEC)* (2002), vol. 5, num 2, pp. 169–200.

[5] Dourish, P., and Bellotti, V. Awareness and Coordination in Shared Workspaces. *In CSCW'92, Toronto* (1992), ACM Press, New York, NY, pp. 107–114.

[6] Godik, S., and (Eds), T. M. eXtensible Access Control Markup Language (XACML) 1.0 Specification Set. OASIS Standard, 2003.

[7] Harville, M., Gordon, G., and Woodfill, J. Foreground Segmentation using Adaptive Mixture Models in Color and Depth. *In Proc. of the IEEE Workshop on Detection and Recognition of Events in Video* (2001).

[8] Horn, B. K. *Robot Vision.* McGraw-Hill Higher Education, 1986.

[9] Juels, A. Minimalistic Cryptography for Low-Cost RFID Tags. *In Submission* (2003).

[10] Juels, A., Rivest, R. L., and Szydlo, M. The Blocker Tag: Selective Blocking of RFID tags for Consumer Privacy. *In Proc. of ACM CCS* (2003).

[11] Lipton, A., Fujiyoshi, H., and Patil, R. Moving Target Classification and Tracking From Real-time Video. *In IEEE Image Understanding Workshop* (1998), pp. 129–136.

[12] M. Chiesa, R. Genz, F. H. e. a. RFID: A Week Long Survey on the Technology and its Potential. Tech. rep., Harnessing Technology Project, Interaction Design Institute Ivrea, 2002.

[13] Marchesotti, L., Marcenaro, L., and Regazzoni, L. A Video Surveillance Architecture for Alarm Generation and Video Sequences Retrieval. *In ICIP2002* (2002).

[14] Privacy International. Privacy International: Video Surveillance. http://www.privacyinternational.org/issues/cctv/index.html.

[15] S.E. Sarma. Toward the Five-Cent Tag. Technical Report, MIT-AUTOID-WH-006, MIT AutoID Center, 2001.

[16] Stauffer, C., and Grimson, W. E. L. Learning Patterns of Activity Using Real-Time Tracking. *In IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (2000), 747–757.

[17] Stevens, M. R., Culbertson, W. B., and Malzbender, T. A Histogram-based Color Consistency Test for Voxel Coloring. *In Proc. of the 16th International Conference on Pattern Recognition (IAPR)* (2002).

[18] Wang, J., Kankanhalli, M. S., Yan, W., and Jain, R. Experiential Sampling for video surveillance. *In First ACM SIGMM international workshop on Video surveillance* (2003), ACM Press, pp. 77–86.

[19] Zhao, Q., and Stasko, J. Evaluating Image Filtering Based Techniques in Media Space Applications. *In CSCW'98, Seattle)* (1998), ACM Press, New York, NY, pp. 11–18.

[20] Zhao, W., Chellappa, R., Phillips, P. J., and Rosenfeld, A. Face recognition: A literature survey. *In ACM Comput. Surv. 35*, 4 (2003), 399–458.