PriSurv: Privacy Protected Video Surveillance System Using Adaptive Visual Abstraction

Kenta Chinomi, Naoko Nitta, Yoshimichi Ito, and Noboru Babaguchi

Graduate School of Engineering, Osaka University
2-1 Yamadaoka Suita, 565-0871 Japan
{chinomi,naoko,ito,babaguchi}@nanase.comm.eng.osaka-u.ac.jp

Abstract. Recently, video surveillance has received a lot of attention as a technology to realize a secure and safe community. Video surveillance is useful for crime deterrence and investigations, but may cause the invasion of privacy. In this paper, we propose a video surveillance system named PriSurv, which is characterized by *visual abstraction*. This system protects the privacy of objects in a video by referring to their privacy policies which are determined according to closeness between objects in the video and viewers monitoring it. A prototype of PriSurv is able to protect the privacy adaptively through *visual abstraction*.

1 Introduction

The concern about video surveillance has been growing in recent years in accordance with increasing demands for security. Video surveillance allows us to remotely monitor a live or recorded video feed which often includes objects such as people. It is recognized that video surveillance contributes to realizing a secure and safe community. When video surveillance systems are widely deployed, we are faced with a problem of privacy invasion. Objects' privacy may be exposed in the videos recorded by surveillance cameras set in public spaces. However, their privacy should be protected anytime even while surveillance is in operation. The challenge of video surveillance is to balance security and privacy appropriately.

Over the past few years, several studies have been made on privacy with video surveillance. Newton et al.[1], Cavallaro et al.[2] and Kitahara et al.[3] proposed the methods for protecting objects' privacy by image processing such as blur and mosaic. The methods presented by Wickramasuriya et al.[4] and Senior et al.[5] protect object's privacy based on the authority either of objects or viewers. Zhang et al.[6] proposed the method for embedding privacy information using digital watermarking technology. Embedded information is obtained only by authorized viewers. Considering more appropriate privacy protection, Sekiguchi et al.[7] proposed the system to protect object's privacy in accordance with the requests of viewers and objects. In this system, the viewer's request is prior to the object's request for practicality of the surveillance system.

These methods, however, are insufficient for privacy protection because the difference of each object's sense of privacy is not considered. For example, some objects may want their privacy to be protected from particular viewers, while

S. Satoh, F. Nack, and M. Etoh (Eds.): MMM 2008, LNCS 4903, pp. 144–154, 2008.

others may not care about their privacy at all. This implies that we must adaptively protect objects' privacy.

This work has been motivated by the need to protect objects' privacy considering the difference of each object's sense of privacy. The contribution of this work is as follows:

- To propose an approach to protecting privacy of people adaptively with video surveillance.
- To embody video/image processing for protecting visual privacy.
- To settle video surveillance as a secure social system.

In this paper, we propose Privacy Protected Video Surveillance System PriSurv. PriSurv aims to protect objects' privacy based on their privacy policies which are determined according to closeness between objects and viewers. A mechanism called *visual abstraction* is introduced to control disclosure of visual information according to objects' privacy policies. Objects' privacy policies give types of abstraction operators which are functions to control visual information of objects.

2 System Overview

We assume PriSurv should be used in a small community such as school areas. In the community, a certain number of members are registered and they would monitor each other. The members would sometimes be viewers and other times be objects.

It is noted that PriSurv is characterized by adaptive *visual abstraction*. PriSurv is capable of generating several kinds of images reflecting on the relationship between the objects and the viewers. The appearance of the objects should be open only to the viewers that objects feel close to, and be hidden to the viewers that objects feel distant from. In this way, PriSurv can control the objects' privacy for each individual viewer.

Fig. 1 is an example showing the feature of PriSurv. Let o, v and a denote an object, a viewer and an abstraction operator, respectively. An original image S including the object o is processed through the abstraction operator a, and the viewer v obtains a privacy protected image $S^a_{<o,v>}$. In Fig. 1, 'Taro' is the object and is monitored by four viewers: 'Stranger 1', 'Neighbor 1', 'Neighbor 2' and 'Family 1'. The appearance of 'Taro' should be changed to each viewer because the closeness levels between 'Taro' and four viewers are different. The original image S is processed through *visual abstraction* by the abstraction operators 'Dot', 'Silhouette', 'Mosaic' and 'As-Is', and the abstracted images $S^{Dot}_{<Taro,Stranger1>}$, $S^{Silhouette}_{<Taro,Neighbor1>}$, $S^{Mosaic}_{<Taro,Neighbor2>}$ and $S^{As-Is}_{<Taro,Family1>}$ are sent to 'Stranger 1', 'Neighbor 1', 'Neighbor 2' and 'Family 1', respectively.

Fig. 2 shows the system architecture of PriSurv. Viewers need to access the main server through an open network and be authenticated to monitor a surveillance video. The function of each module in Fig. 2 is as follows:

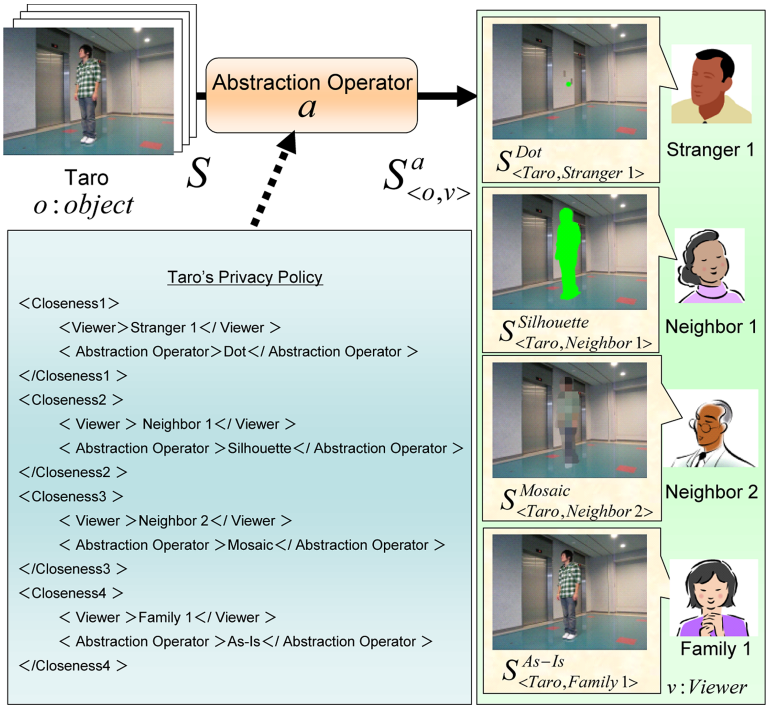

Fig. 1. Feature of PriSurv

– Analyzer
  Analyzer has a function of *object identification* which is a process to label IDs on each object in an image. On condition that each object has an RFID-tag, objects in images are identified with a combination of RFID-tags and video analysis.
– Profile Generator
  Profile Generator facilitates to set profiles of members. Profiles have members' privacy information, which are privacy policies and attributions such as name, age, gender and address. Registered members can set their privacy information simply on a graphical user interface, then Profile Generator converts the settings to XML based syntax. Profiles can be updated only by their owners and can not be accessed by other members.
– Profile Base
  Profile Base stores a set of profiles of the registered members securely in the server.
– Access Controller
  Access Controller reads XML based privacy policies of objects and look up viewers' IDs in objects' privacy policies to determine appropriate abstraction operators. Access Controller sends the types of operators to Abstractor.

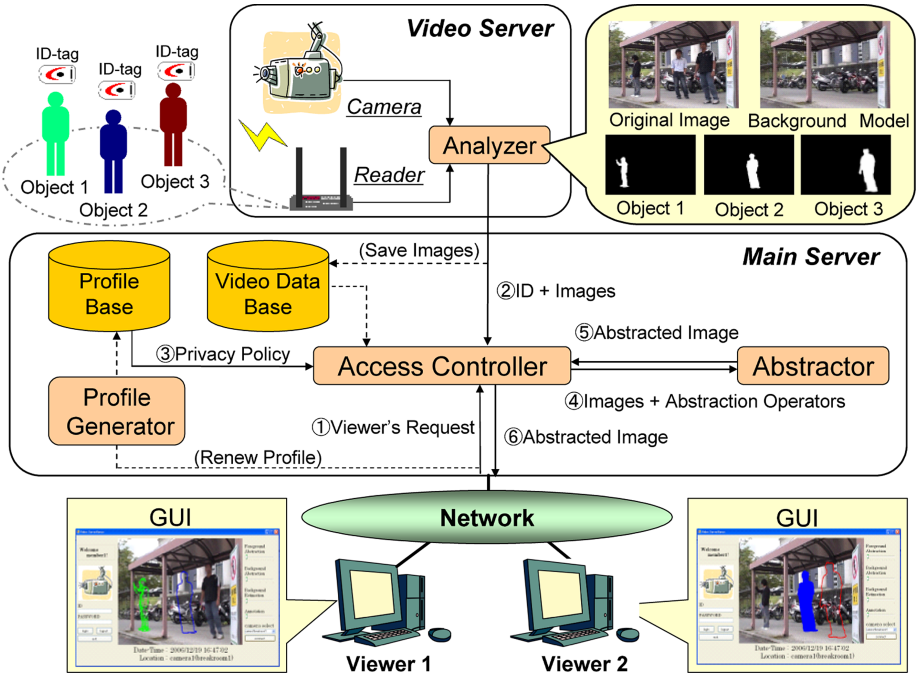

Fig. 2. System Architecture

– Abstractor
Abstractor is an image processor for *visual abstraction* by the abstraction operators specified from Access Controller.
– Video Data Base
Video Data Base stores past video data and supplies them to viewers through *visual abstraction* when necessary.

3 Object Identification

PriSurv needs to firstly identify objects in original images. Here, assuming that every member has his own RFID-tag, the surveillance space is divided into N × N areas and we estimate in which area each RFID-tag and object in images are present. IDs are then assigned to all objects in images by integrating these estimation results. Fig. 3 shows the process of object identification. The process consists of three steps:

1. Area estimation of RFID-tags carried by objects
2. Area estimation of objects in images
3. Integration of area estimation results

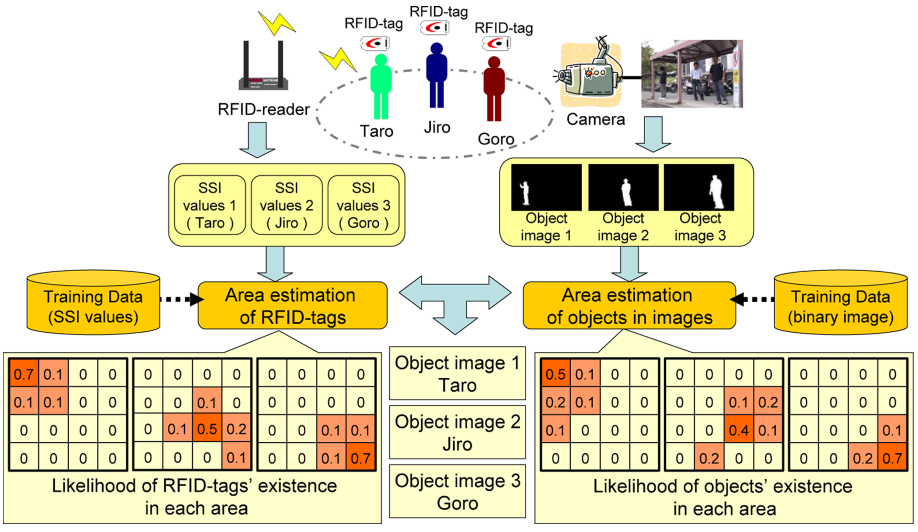

Fig. 3. Object Identification

These three steps are described below.

RFID-tags are useful for object identification since they provide the information about who and where their carriers are. Each tag emits a signal encoded with a unique ID number, which is received by an RFID-reader. As a result, the SSI (Signal Strength Indicator) values of the signals received by all RFID-readers set in the surveillance space are measured for each RFID-tag. We firstly prepare a training data by collecting a set of these SSI values when RFID-tags are at arbitrary locations in each area. When SSI values are measured from an unknown location, the likelihood of the RFID-tag's existence in each area is calculated by examining their k-nearest neighbors in the training data.

In order to identify objects in the original image, each object has to be extracted separately. Therefore, we firstly estimate a background model image by Gaussian mixuture model and extract a foregound image per original image, where each detached region is considered as an object. Since the appearance of the object should be different depending on the spatial relationship between the object and the camera, we firstly prepare a training data by collecting binary images of each object, when the object is at arbitrary locations in each area. When a new original image is obtained, the likelihood of each object in the image to exist in each area is calculated by examining their k-nearest neighbors in the training data.

Finally, the calculated two likelihood values in each area are multiplied to calculate the likelihood of each RFID-tag and object to co-exist in the corresponding area. The ID of each RFID-tag is assigned to the object with the highest likelihood.

4 Privacy Policy

It is effective to set privacy policies based on closeness between objects and viewers [9]. Therefore, each registered member classifies other members into several closeness groups and sets an abstraction operator to each group. The members who are not classified in any group are grouped as a non-classified group.

Fig. 4 shows an example of Taro's privacy policy. This privacy policy is described in XACML which is a language for expressing access control policies. Note that the example policy is simplified for readability.

5 Visual Abstraction

Privacy protection is achieved by hiding object's visual information (e.g. expression, clothe, etc.). However, excess *visual abstraction* makes video surveillance meaningless. In PriSurv, both safety and privacy protection are needed. Therefore we implement several abstraction operators to control visual information gradually. We provide the following 12 operators as shown in Fig. 5.

- As-Is
  An object is not abstracted. All visual information is visible. The abstraction level is the lowest of all operators.
- See-through
  Pixel values of an object image and a background model image are blended. The background is visible through the object.
- Monotone
  Color information is hidden. An object is expressed in black and white.
- Blur
  An object is blurred. Details of the object are hidden.
- Mosaic
  An object is mosaicked. Details of the object are hidden.
- Edge
  An object is expressed by an extracted edge in one color.
- Border
  An object is expressed by an extracted border line in one color. The background is visible in the object's region.
- Silhouette
  An object is expressed by a silhouette painted out in one color. The background is not visible in the object's region.
- Box
  An object is expressed by a painted box. We can recognize the height and the width of the object.
- Bar
  An object is expressed by a painted bar. We can recognize the height of the object.
- Dot
  An object is expressed by a painted dot. We can recognize only the location of the object.

```
-<PolicySet Policy SetId="Taro's PolicySet">
    -<Policy PolicyId=" Closeness Level 1">
        -<Rule RuleId="WatchRule 1">
            -<Subject>Stranger1</Subject>
         </Rule>
        -<Obligation ObligationId=" Abstractor 1">
            -<Attribute Assignment>Dot< /AttributeAssignment >
        </Obligation>
    </Policy>
    -<Policy PolicyId=" Closeness Level 2">
        -<Rule RuleId="WatchRule 2">
            -<Subject>Neighbor 1</Subject>
         </Rule>
        -<Obligation ObligationId=" Abstractor 2">
            -<Attribute Assignment>Silhouette< /AttributeAssignment >
        </Obligation>
    </Policy>
    -<Policy PolicyId=" Closeness Level 3">
        -<Rule RuleId="WatchRule 3">
            -<Subject>Neighbor 2</Subject>
         </Rule>
        -<Obligation ObligationId=" Abstractor 3">
            -<Attribute Assignment>Mosaic< /AttributeAssignment >
        </Obligation>
    </Policy>
    -<Policy PolicyId=" Closeness Level 4">
        -<Rule RuleId="WatchRule 4">
            -<Subject>Family 1</Subject>
         </Rule>
        -<Obligation ObligationId=" Abstractor 4">
            -<Attribute Assignment>As-Is< /AttributeAssignment >
        </Obligation>
    </Policy>
</PlicySet>
```

Fig. 4. Example of Privacy Policy

– Transparency
  All visual information of an object is hidden as if the object were not there. The abstraction level is the highest of all operators.

Visual information controlled by each operator is different as illustrated in Table 1. In this figure, '◯', '△' and '×' mean disclosed, partially-hidden and

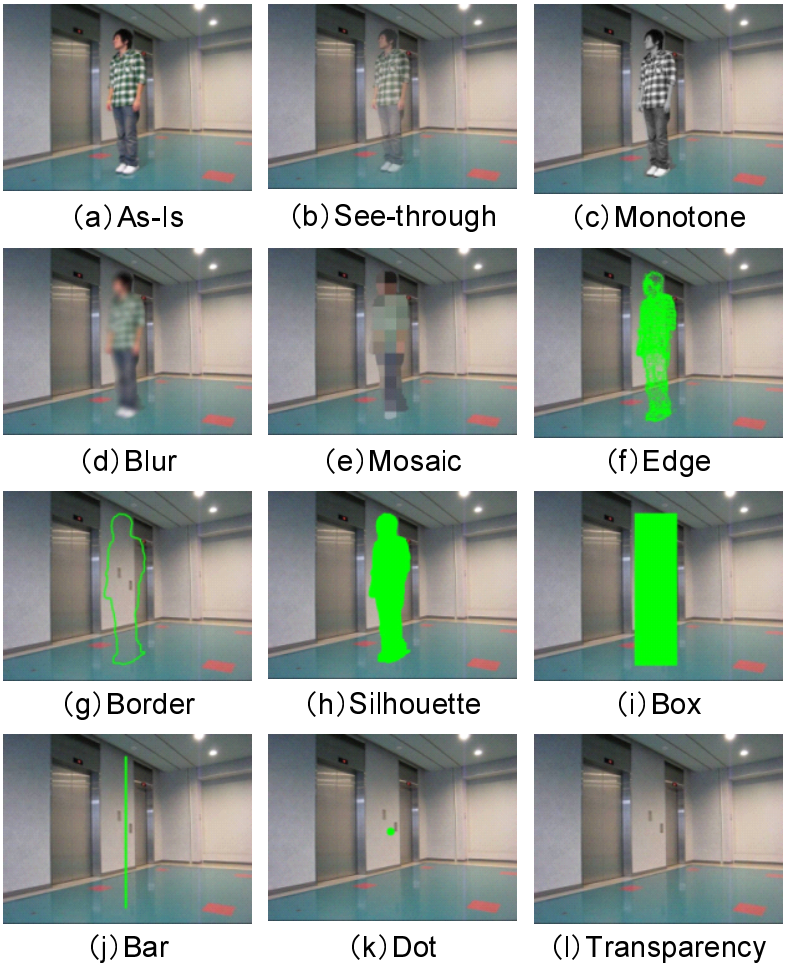

Fig. 5. Visual Abstraction

hidden, respectively. The operators are shown in increasing order of the abstraction level from top to bottom. In addition, the abstraction operators can be applied to a part of the object, i.e. his/her head or body.

6 Formulation

We formulate a process of generating privacy protected images.

An $M \times N$ original image can be represented as

$$S = \{\boldsymbol{s}_{ij}\}\ (i = 1, 2, \cdots, M; j = 1, 2, \cdots, N) \tag{1}$$

where $\boldsymbol{s}_{ij} = (s^r_{ij}, s^g_{ij}, s^b_{ij})$ is a three-dimensional vector of RGB pixel values. In the original image S, let B denote the background region and F denote the

Table 1. Visual Information

	existence	location	height	width	silhouette	hairstyle	clothes	expression
As-Is	○	○	○	○	○	○	○	○
See-through	○	○	○	○	○	○	○	○
Monotone	○	○	○	○	○	○	△	○
Blur	○	○	○	○	○	△	△	×
Mosaic	○	○	○	○	○	△	△	×
Edge	○	○	○	○	○	△	△	×
Border	○	○	○	○	○	×	×	×
Silhouette	○	○	○	○	○	×	×	×
Box	○	○	○	○	×	×	×	×
Bar	○	○	○	×	×	×	×	×
Dot	○	○	×	×	×	×	×	×
Transparency	×	×	×	×	×	×	×	×

foreground region. Then a background image S_B and a foreground image S_F can be expressed as

$$S_B = \left\{ \boldsymbol{b}_{ij} \,\middle|\, \begin{array}{l} \boldsymbol{b}_{ij} = \boldsymbol{0} \quad (i,j) \in \bar{B} \\ \boldsymbol{b}_{ij} = \boldsymbol{s}_{ij} \ (i,j) \in B \end{array} \right\} \tag{2}$$

$$S_F = \left\{ \boldsymbol{f}_{ij} \,\middle|\, \begin{array}{l} \boldsymbol{f}_{ij} = \boldsymbol{0} \quad (i,j) \in \bar{F} \\ \boldsymbol{f}_{ij} = \boldsymbol{s}_{ij} \ (i,j) \in F \end{array} \right\} \tag{3}$$

where $\bar{B}$ is the complement region of B in S and $\bar{F}$ is the complement region of F in S. If there are N_o objects, F is divided into subregions F_n corresponding to objects $o_n (n = 1, 2, \cdots, N_0)$. Then, an object's image S_{F_n} is represented as follows:

$$S_{F_n} = \left\{ \boldsymbol{f}^n_{ij} \,\middle|\, \begin{array}{l} \boldsymbol{f}^n_{ij} = \boldsymbol{0} \quad (i,j) \in \bar{F}_n \\ \boldsymbol{f}^n_{ij} = \boldsymbol{s}_{ij} \ (i,j) \in F_n \end{array} \right\} \tag{4}$$

$\boldsymbol{b}_{ij}$, $\boldsymbol{f}_{ij}$ and $\boldsymbol{f}^n_{ij}$ are three-dimensional vectors of RGB pixel values. Using S_B and $S_{F_n} (n = 1, 2, \cdots, N_0)$, we represent the original image S in terms of stratified images as

$$S = S_B + S_{F_1} + S_{F_2} + \cdots + S_{F_{N_o}} \tag{5}$$

where the operator + adds the pixel values of the same coordinate in S_B and each S_{F_n}.

We use the background model image $\tilde{S}_B$, estimated by Gaussian mixture model, as the background of the privacy protected image. Let a_0 denote the abstraction operator for the background model image $\tilde{S}_B$ and a_n denote the abstraction operator for the image S_{F_n} of the object o_n. We define two sets $\boldsymbol{a} = (a_0, a_1, a_2, \cdots, a_{N_o})$ and $\boldsymbol{o} = (o_1, o_2, \cdots, o_{N_o})$. Now we can say that access control based on objects' privacy policies is a process to determine a set of abstraction operators $\boldsymbol{a}$ to a given set of objects $\boldsymbol{o}$.

Visual abstraction by $\boldsymbol{a}$ on objects' images S_{F_n} and the background model image $\tilde{S}_B$ generates abstracted images as follows:

$$S_{F_n}^{a_n} = \left\{ \boldsymbol{f}_{n,ij}^{a_n} \;\middle|\; \begin{array}{ll} \boldsymbol{f}_{n,ij}^{a_n} = \mathbf{0} & (i,j) \in \bar{F}_n^{a_n} \\ \boldsymbol{f}_{n,ij}^{a_n} = a_n(\boldsymbol{s}_{ij}) & (i,j) \in F_n^{a_n} \end{array} \right\}$$

$$\tilde{S}_B^{a_0} = \left\{ \tilde{\boldsymbol{b}}_{ij}^{a_0} \;\middle|\; \begin{array}{ll} \tilde{\boldsymbol{b}}_{ij}^{a_0} = \mathbf{0} & (i,j) \in \bar{\tilde{B}}^{a_0} \\ \tilde{\boldsymbol{b}}_{ij}^{a_0} = a_0(\tilde{\boldsymbol{b}}_{ij}) & (i,j) \in \tilde{B}^{a_0} \end{array} \right\}$$

where $F_n^{a_n}$ is the region of the foreground image in which *visual abstraction* is executed and $\tilde{B}^{a_0}$ is the visible region of the background model image after *visual abstraction* for the foreground images. The privacy protected image $S^{\boldsymbol{a}}_{<\boldsymbol{o},v>}$, which is sent to the viewer v, is expressed as follows:

$$S^{\boldsymbol{a}}_{<\boldsymbol{o},v>} = \tilde{S}_B^{a_0} + S_{F_1}^{a_1} + S_{F_2}^{a_2} + \cdots + S_{F_{N_o}}^{a_{N_o}}$$

7 Implementation and Evaluation

Fig. 6 shows a graphical user interface of PriSurv prototype. The viewer is able to select the type of abstraction operators and the surveillance camera. Assume that a viewer 'yuta' is monitoring three objects: 'object 1', 'object 2' and 'object 3', from the left. In the shown image $S^{\boldsymbol{a}}_{<\boldsymbol{o},v>}$, v, $\boldsymbol{o}$ and $\boldsymbol{a}$ are as follows: $v = yuta$, $\boldsymbol{o} = (object1, object2, object3)$, $\boldsymbol{a} = (Box, Silhouette, Mosaic)$.

We experimented our object identification method with a 4.75m × 4.5m room as a surveillance space. The room is evenly divided into 16 square areas and has 4 RFID-readers, each of which obtains two SSI values. When two people consistently walk around in the room, the proposed method was able to identify both objects with the accuracy of 81%.

We also evaluated the performance of visual abstraction. The frame rate of the video should depend on the image size, the number of objects, and the type

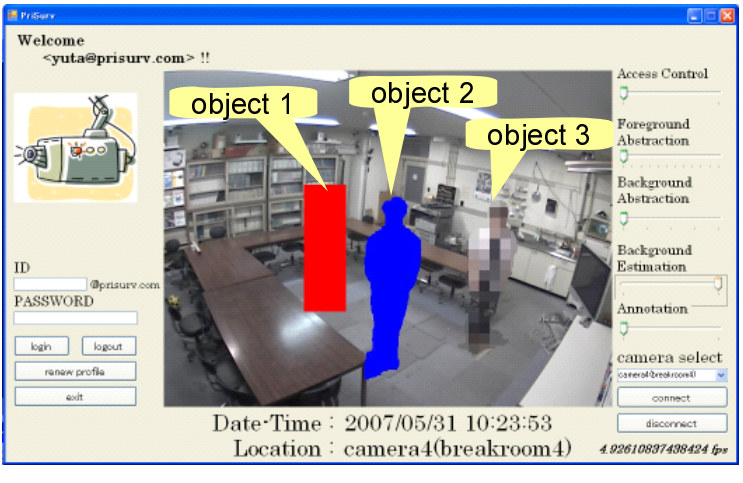

Fig. 6. Graphical User Interface

of the abstraction operator. When the size of images is 640 × 480, the average frame rate was 15.0 fps without any object. When there is one object whose region size is about 150 × 420, the average frame rate was the highest at 12.2 fps and the lowest at 8.8 fps when the abstraction operator was Transparency and See-through, respectively.

8 Conclusion

In this paper, we have proposed Privacy Protected Video Surveillance System PriSurv, focusing on *visual abstraction*. PriSurv refers objects' privacy policies, which are determined according to objects' closeness with viewers, and determines abstraction operators to hide visual information of objects. This method enables us to adaptively protect each object's privacy.

To make PriSurv more practical, we need to improve and integrate *object identification*. We also need to consider network security because there is a risk for the invasion of privacy by eavesdroppers.

Acknowledgments. This work was supported in part by a Grant-in-Aid for scientific research and by SCOPE.

References

1. Newton, E.M., Sweeny, L., Malin, B.: Preserving Privacy by De-Identifying Face Images. IEEE Trans. Knowledge and Data Engineering 17(2), 232–243 (2005)
2. Cavallaro, A., Steiger, O., Ebrahimi, T.: Semantic Video Analysis for Adaptive Content Delivery and Automatic Description. IEEE Trans. Circuits and Systems Video Technology 15(10), 1200–1209 (2005)
3. Kirahara, I., Kogure, K., Hagita, N.: Stealth Vision for Protecting Privacy. In: Proc. 17th International Conference on Pattern Recognition, vol. 4, pp. 404–407 (2004)
4. Wickramasuriya, J., Alhazzazi, M., Datt, M., Mehrotra, S., Venkatasubramanian, N.: Privacy-Protecting Video Surveillance. In: Proc. SPIE International Symposium on Electronic Imaging, vol. 5671, pp. 64–75 (2005)
5. Senior, A., Pankati, S., Hampapur, A., Brown, L., Tian, Y.-L., Ekin, A., Connell, J., Shu, C.F., Lu, M.: Enabling Video Privacy through Computer Vision. IEEE Security and Privacy Magazine 3(3), 50–57 (2005)
6. Zhang, W., Cheung, S.S., Chen, M.: Hiding Privacy Information in Video Surveillance System. In: ICIP2005. Proc. IEEE International Conference on Image Processing, pp. 868–871 (2005)
7. Sekiguchi, T., Kato, H.: Proposal and Evaluation of Video-based Privacy Assuring System Based on the Relationship between Observers and Subjects. IPSJ Trans. on Computer Security Proping up Ubiquitous Society 47(8), 2660–2668 (2006)
8. Stauffer, C., Grimson, W.E.L.: Learning Patterns of Activity using Real-Time Tracking. IEEE Trans. Pattern Analysis and Machine Intelligence 22, 747–757 (2000)
9. Koshimizu, T., Toriyama, T., Babaguchi, N.: Factors on the Sense of Privacy in Video Surveillance. In: Proc. Workshop on Capture, Archival and Retreival of Personal Experiences, pp. 35–43 (2006)