2008－CVIM－163 (28)
2008／5／8

Detection of unseen objects in a 3-D dynamic scene

Satoshi KAWABATA† Shinsaku HIURA† Kosuke SATO†
†Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka, 560-8531 Japan
Tel +81-6-6850-6371 kawabata@sens.sys.es.osaka-u.ac.jp

Abstract Detecting an unseen object in 3-D space is one of the important task on the field of computer vision. Especially, background subtraction is the most popular technique to extract the region of an unseen object from 2-D image taken under the static condition. Though, such condition is rare for practical use; A real scene contains moving object which we previously know its existence. Thus the system must have the ability to distinguish unfamiliar object from moving objects. Although the system can successfully extract the region of an unseen object, we have to decide that the object is intruding into certain region or not. In this paper, we propose a method to extract the region of unseen object from dynamically moving background using iterative projection onto eigenspace by kernel PCA of a background image sequence. Additionally, we realize a calibration-free 3-D intrusion detection system with multiple uncalibrated cameras by integrating the extracted region of unseen object in 2-D captured image.

Keywords: Intrusion detection, background subtraction, kernel PCA, visual hull method.

1 Introduction

Nowadays a lot of cameras are used at shops, museums, banks, roads and so on, but taken images are only recorded and mainly used for evidence for postmortem. In the case of very critical missions which requires real-time responses, human would watch many monitors tiled on a wall. To save such labor and cost for monitoring, special sensors are often used. Light beam detectors are used to find undesirable intrusion, but the sensitive lines are not flexible because it is strictly related to the arrangement of the equipments. Passive infra-red sensor which detects the heat of human body is also widely used, but it will detect only the existence of the human body and detailed observation of monitoring area is impossible.

For replacing human observer with a computer, image-based surveillance and monitoring method is extensively researched [1]. In these research, they apply the techniques to restricted admission by image-based person identification, motion analysis of human or car, alert for anomalous events and so on. More recently, a gait analysis has been attracted researchers' attention. For example, Matsumura detects abnormal behavior of human from a trait of target on omni-

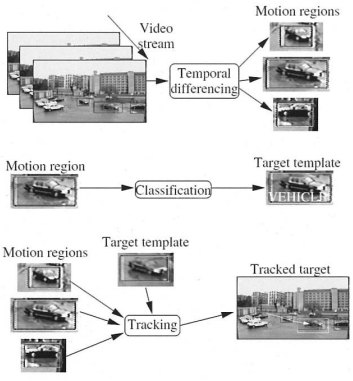


Fig.1 Flow of DARPA VASM system [3]

image by comparing the trait and a transition model has been obtained in advance [2]. Otsu used CHLAC feature for precise gait identification, where the feature is invariant for translation and time.

On the other hand, there are several researches not only using cameras fixed in space, but also intentionally utilizing many cameras to recognize human action [3](Fig. 1). VSAM project constructs the system that is able to classify target objects and track them

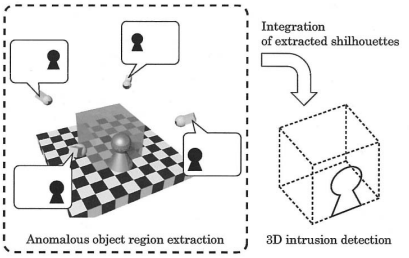


Fig.2 Outline of proposed system

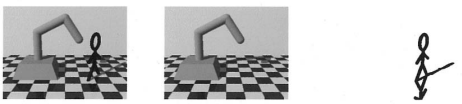


Fig.3 Background subtraction

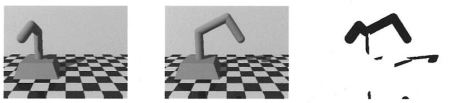


Fig.4 Object extraction under a moving background objects

even though they occlude each other. The system is robust for the occlusion by using templates of objects, however, it cannot handle fine motion nor deformation of object. Thus it isn't suitable to detect a 3-D intrusion.

Therefore, we propose the intrusion detection system for versatile purpose in this paper. The system consists of extraction method of an unseen object region from 2-D image and 3-D intrusion detection method by integrating the regions using multiple uncalibrated cameras(Fig 2).

2 Unseen Object Region Extraction from Captured Image

Background subtraction uses static background captured before monitoring (Fig. 3), therefore it can not handle the moving object as a background (Fig. 4). Real-time background maintenance, such as median filter, is widely used to solve this problem, but fast motion of the background object will be detected as a intruder.

Therefore, some background models have been proposed to adapt the background subtraction method to a kind of dynamic scene.

2.1 Extracting Object from Changing Background

Background subtraction is the method to extract a region from captured image whose amount of change exceeds a predefined criteria by comparing the image with the background model. Thus it is possible to apply a sophisticated background subtraction with improved background model to the scene which contains moving background objects.

Generally, the factors leads to false positive error of an extraction result are as follows:

- Global illumination change by varying position or luminance of light source,
- Local illumination change by a cast shadow or inter-reflection,
- Rapid illumination change by illuminating a spot light,
- Transparent object or the object with similar texture to background,
- Waving leaves, grasses,
- Motion of a background object.

An error caused by illumination changes can be suppressed by choosing an appropriate model and criterion. Meanwhile, it is impossible to extract objects having similar look as background object because they have similar feature to one of background object.

To deal with the scene of waving leaves, the extraction method which make the sensitivity of such variable region lower with the help of statistics. However, this approach tends to increase the false negative in the region.

Motion of background object, the last factor, means background image completely changes every moment. So it is need to estimate the background image at the moment from observed image.

2.1.1 Background Models for Dynamic Scene

In the case of a dynamic scene, it is important to construct a background model from multiple images (ex. image sequence of moving background object, image set of varying illumination etc.) rather than a single background image.

Elgammal et al. represented each pixel value with a mixture of gaussian (MOG) distribution and esti-

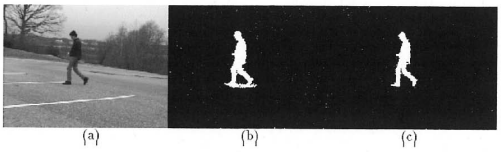


Fig.5 Extraction using MOG [4]

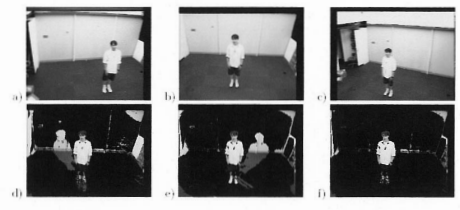


Fig.6 Extraction using a disparity map [5]

mated the parameters from multiple images to construct a background model using EM algorithm [4]. And successfully extracted a human region from the scene contains waving leaves and illumination changes. It is difficult for this model to handle a local illumination change (ex. cast shadow) (Fig. 5 (b)), they normalized image by:

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \frac{1}{R+G+B} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (1)$$

If the system could obtain a depth information about a background, it is possible to achieve the background subtraction free from illumination changes. Ivanov et al. have proposed the extraction method in static scene using a disparity map as the background model, which is robust to shadows [5] (Fig. 6).

2.1.2 Using Correlation or Co-occurrence with Neighboring Pixels

Javed et al. evaluated not only difference value at each pixel but also spatial differential for an object extraction [6] (Fig. 7), which is robust to global illumination change and translate background objects. However, they adapt the translation by updating a background model at the right moment. The extraction method can't use with a continuously moving scene because the background model is outdated until updating it.

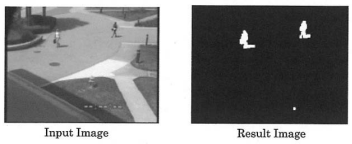


Fig.7 Extraction Using Neighbor Pixel Information [6]

2.1.3 Problems in Implementing on Intrusion Detection System

In an intrusion detection system under a dynamic scene, a mixture of gaussian model, which can handle a local illumination change, is not suitable because a background object dynamically moves. The adaptive approach based on updating has also been proposed, however this only makes time of false detection shorter and doesn't solve the problem. In addition, there is no warranty of correctness of a background model with the updating approach. That is, a fixed background model of a dynamic scene is desirable for an intrusion detection system.

Here, when the whole variation of background image is observable in advance, it is possible to extract the region of unseen object by just referencing an appropriate background image without any parameterization.

3 Object Extraction by Estimation of Background Image using Eigenspace

As mentioned above, the extraction under a dynamic scene can be done by estimating an appropriate background image from newly captured image. Nevertheless, we can't obtain the background image by simply searching in stored images because new input image usually contains an unseen object somewhere, so we have no criteria to measure the similarity between the input image and each stored image.

The eigenbackground is a popular technique to handle a background image set [7]. Once represent a image set as an eigenspace, a different image from the set can't be synthesized using the eigenspace. Therefore, to estimate a background image, it is done by simply projecting input image to an eigenspace then back-projecting to image space. This framework works well only if unseen object region is small. To handle large unseen object in an input image, we have to exclude the region explicitly from the estimation

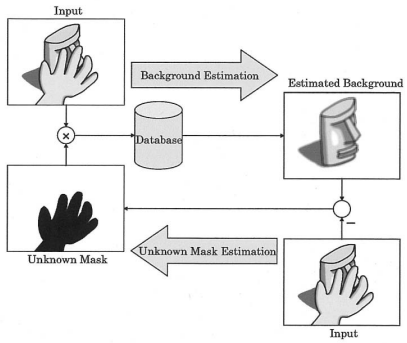


Fig.8 Background estimation and object extraction

process. However, since we have no information the object region to exclude, it is need to iterate the background estimation and the object extraction alternately (Fig. 8).

3.1 Image Reconstruction from Eigenspace

When we have s background images $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_s$, one of background images $\boldsymbol{x}_i$ can be approximated using matrix $E = [\boldsymbol{e}_1 \ldots \boldsymbol{e}_d] \quad (d < s)$ which consist of orthonormal basis $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ calculated using principal component analysis (PCA), as

$$\boldsymbol{x}_i \approx E\boldsymbol{p}_i + \bar{\boldsymbol{x}} \tag{2}$$

where $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ is called as eigenvectors. To simplify the equation, we omit the average image $\bar{\boldsymbol{x}}$ because it is constant.

$$\boldsymbol{x}_i \approx E\boldsymbol{p}_i \tag{3}$$

Inversely, we can calculate the point $\boldsymbol{p}$ in an eigenspace which corresponds to the background image $\boldsymbol{x}$ as

$$\boldsymbol{p} = E^T\boldsymbol{x} \tag{4}$$

where $(E^TE)^{-1}E^T = E^T$, because E is orthonormal. If we have the point $\boldsymbol{p}$, the background image $\boldsymbol{x}$ can be reconstructed using eigenvectors.

$$\boldsymbol{x} \approx E\boldsymbol{p} \quad (= EE^T\boldsymbol{x}) \tag{5}$$

3.2 Background Estimation

As described above, we can reconstruct the background image if we can estimate the point in an eigenspace, $\boldsymbol{p}$. If the background contains not only a perturbation (e.g. illumination variations [7], etc.) but dynamic motion of background objects, we must explicitly exclude a occluded region by an unknown object which causes a considable estimation error of background image. In this paper, we propose the method to estimte $\boldsymbol{p}$ from a image $\tilde{\boldsymbol{x}}$ a part of which is occluded by unknown objects. BPLP method [8] calculates the point to minimize the difference between input and estimated image except the occluded region. This method can be represented using diagonal matrix Σ whose diagonal element is 0 when the corresponding pixel is occluded and 1 not occluded, as

$$\varepsilon^T\varepsilon \xrightarrow{\boldsymbol{p}} \min. \tag{6}$$

$$\varepsilon = (\tilde{\boldsymbol{x}} - E\boldsymbol{p})^T \Sigma^T\Sigma (\tilde{\boldsymbol{x}} - E\boldsymbol{p}) \tag{7}$$

and the solution $\hat{\boldsymbol{p}}$ is given as follows.

$$\hat{\boldsymbol{p}} = \left(E^T\Sigma^T\Sigma E\right)^{-1} E^T\Sigma^T\Sigma\tilde{\boldsymbol{x}} \tag{8}$$

But however, equation (8) takes much time because d^2 times multiplication of image by image must be calculated. So that, we propose an iterative projection method as follows. Instead of equation (8), we use a reccurence equation,

$$\hat{\boldsymbol{p}}_n = E^T\hat{\boldsymbol{x}}_{n-1} \tag{9}$$

where the initial value of estimator $\hat{\boldsymbol{x}}_0$ is an average of all background images, $\hat{\boldsymbol{x}}_0 = 0$. Then we can calculate n^{th} estimated point $\hat{\boldsymbol{p}}_n$ using $n-1^{th}$ estimated background. The background image $\hat{\boldsymbol{x}}_{n-1}$ is a composition of former estimated background and input image $\tilde{\boldsymbol{x}}$,

$$\hat{\boldsymbol{x}}_{n-1} = \Sigma\tilde{\boldsymbol{x}} + (I - \Sigma)\, E\hat{\boldsymbol{p}}_{n-1} \tag{10}$$

where the occluded region of input image is replaced by the former estimated background.

3.3 Updating Occlusion Mask

Both BPLP and our method needs information of the region occluded by intruders, Σ. Unfortunately, this diagonal matrix is unknown because the objective of our research is to estimate the silhouette of the unknown object. Therefore, we calculate the mask Σ by using simple background subtraction as

$$\Sigma(j) = \begin{cases} 1, & \text{for } |\hat{\boldsymbol{x}}(j) - \boldsymbol{x}(j)| < th \\ 0, & \text{for } |\hat{\boldsymbol{x}}(j) - \boldsymbol{x}(j)| \geq th \end{cases} \tag{11}$$

where th is a threshold and (j) denotes each pixel or element of diagonal matrix. Of course, the estimated background $\hat{\boldsymbol{x}}$ is varied when the occlusion mask Σ changes, therefore, background estimation described in section 3.2 and background subtraction by equation (11) must be iterated. The sequence of simultaneous

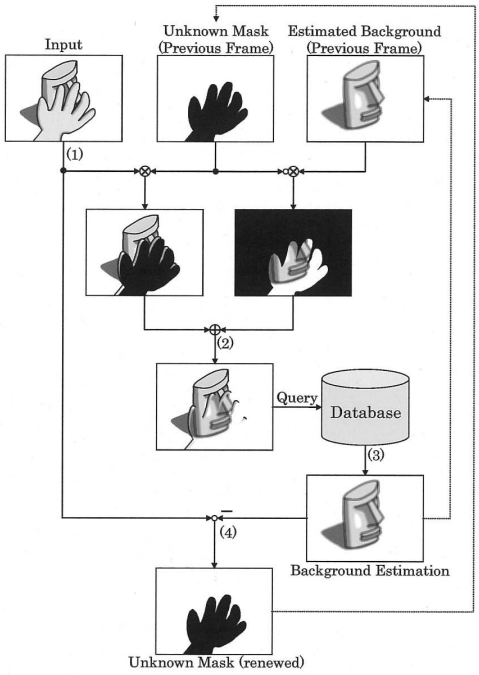


Fig.9 Flow of proposed method

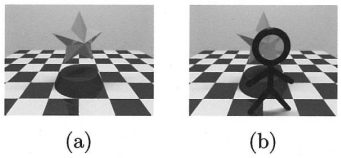


Fig.10 Rendered CG image. (a) background image, (b) input image with moving intruder. In both images, centered star-shaped object rotates.

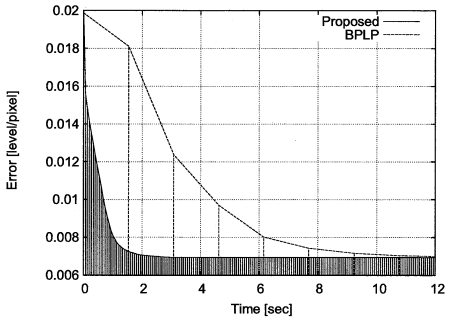


Fig.11 Convergence for fixed image

estimation of background and object region is summarized as follows.

BPLP:

1. capture new image $\tilde{\boldsymbol{x}}$,
2. equation (8) for background estimation using old object region Σ,
3. equation (11) to update object region Σ,
4. go to 1.

Proposed method: (Fig. 9)

1. capture new image $\tilde{\boldsymbol{x}}$,
2. equation (10) to combine input image and old background using old object region Σ,
3. equation (9) to estimate new background image,
4. equation (11) for update object region Σ,
5. go to 1.

Evidently our method has faster frame rate of background subtraction, and it has a potential to faster convergence to the correct region and background.

4 Experiments

4.1 Evaluation using CG images

To compare our method to BPLP equally, we used synthetic images for background and input images. Figure 10(a) shows the background image (ground truth) and 10(b) input image generated by POV-Ray 3.6. In both figures, star-shaped background object rotates just three-sixty through 256 images. The dimension of eigenspace is 58 when accumulation contributing ratio is 95%.

4.1.1 Experiment 1 : Convergence for Fixed Image

In this experiment, we compare the speed of convergence using fixed input image. We used an average image as an initial value.

Figure 11 shows the result of convergence. The error is a variance between estimated image and ground truth. By this result, it is better to update the object region mask rapidly than the slow calculation of the optimal value.

4.1.2 Experiment 2 : Error for real-time sequence

In this experiment, we checked the error for a stream of images. We assumed all 256 images as a sequence of images of 30 fps, therefore the period of

Table1 Average error of estimated background

BPLP	0.028175
Our method	0.014375
Our method (excludes static pixels)	0.010365

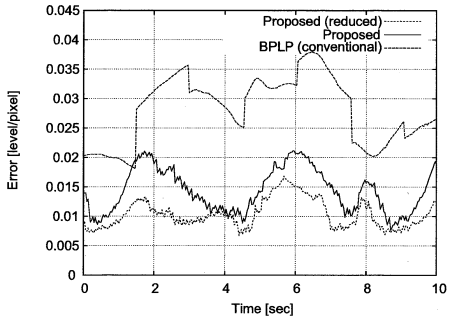


Fig.12 Improvement by exclusion of static pixels

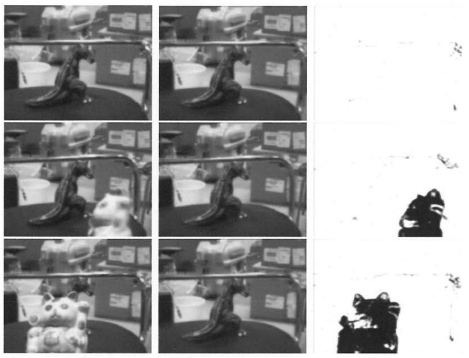


Fig.13 Background Estimation using real images. (a) Input images, (b) Estimated Background, (c) Estimated Object Region.

all images is 8.53sec.

In actual, a part of background image stays static while the background includes dynamic object. Therefore, we exclude such pixels from the estimation. Figure 12 shows the effect of exclusion of static pixels. Since the cycle time of estimation is shorter, the error of estimation becomes smaller. The frame rate of this method is 20 fps to 30 fps. For real-time sequence, our result shows not only less error but also faster cycle of estimation. Table 1 shows the average of the error through the sequence. This results shows our method has better performance for real-time sequence.

4.2 Evaluation using real images

Figure 13 shows the result of our system for the real scene. The background is a miniature of dinosaur rotating on a chair, and the intruding object is a miniature of white cat. The dimension of eigenspace is 46 when the number of background image is 256 and accumulation contributing ratio is 95%. In Figure 13, column (a) shows input images($\tilde{\boldsymbol{x}}$), (b) estimated background ($E\hat{\boldsymbol{p}}_n$), (c) mask image (Σ), respectively. The region of the intruding object is extracted correctly while the background object is moved. Also figure 13 shows the result of extraction of unknown object.

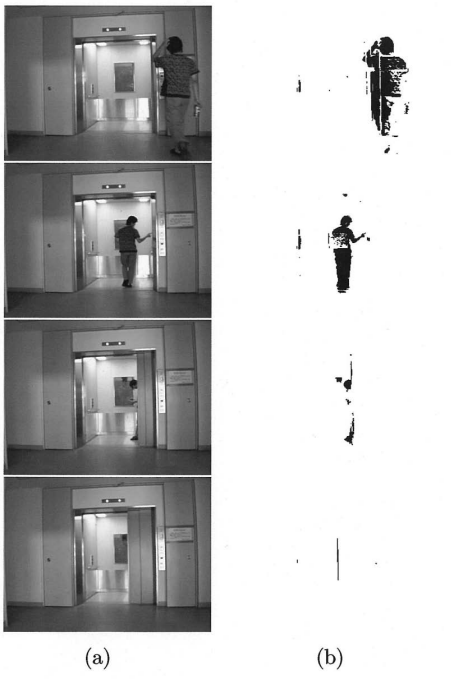


Fig.14 Unknown object extraction using real images. (a) Input images, (b) Estimated Object Region.

5 Nonlinear Eigenspace

Background estimation is based on the assumption that a background image set can be well described as a linear relation of pixels. However, a relationship between background images contains nonlinear correlation in general. In this section, we apply the kernel trick to our iterative projection method for expanding it to handle nonlinear subspaces.

5.1 kernel PCA [9]

Here, nonlinear-projection of a sampled image $\boldsymbol{x}_i$ to higher dimensional feature space by a function $\boldsymbol{\phi}(\boldsymbol{x})$ is:

$$\boldsymbol{x}_i \underset{\boldsymbol{\phi}}{\longmapsto} \boldsymbol{x}_{\phi i}. \tag{12}$$

The covariance matrix in the feature space is represented by:

$$X_\phi X_\phi^{\mathrm{T}} \tag{13}$$

$$X_\phi = \left[\boldsymbol{x}_{\phi 1}, \ldots, \boldsymbol{x}_{\phi s}\right], \tag{14}$$

since the feature space tends to have higher or infinite dimensionality, it is difficult to compute eigenvectors of the space. As think about the SVD of X_ϕ:

$$X_\phi = UDV^{\mathrm{T}}. \tag{15}$$

where, U is a set of eigenvectors of $X_\phi X_\phi^{\mathrm{T}}$, V is a set of eigenvectors of $X_\phi^{\mathrm{T}} X_\phi$. Therefore, the dimension of V doesn't exceed the number of samples while U can have infinite dimensionality.

Introducing the kernel function $k_\phi(\boldsymbol{x}, \boldsymbol{y})$ which value is the innerproduct of $\boldsymbol{x}$ and $\boldsymbol{y}$ in a feature space:

$$k_\phi(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}_\phi^{\mathrm{T}} \boldsymbol{y}_\phi. \tag{16}$$

And we define the following kernel matrix $K_\phi(X, Y)$ using the kernel function:

$$K_\phi(X, Y) = \left\{k_\phi\left(\boldsymbol{x}_i, \boldsymbol{y}_j\right)\right\}_{ij} = X_\phi^{\mathrm{T}} Y_\phi. \tag{17}$$

According to the relation of $X_\phi^{\mathrm{T}} X_\phi = K_\phi(X, X)$, U, a set of eigenvectors of a covariance matrix $X_\phi X_\phi^{\mathrm{T}}$ satisfies following relation.

$$K_\phi(X, X) = VD^2V^{\mathrm{T}} \tag{18}$$

$$U = X_\phi V D^{-1}. \tag{19}$$

So a projection point $\boldsymbol{p}$ corresponding to the point of image $\boldsymbol{x}$ in feature space is computed as:

$$\boldsymbol{p} = U^{\mathrm{T}} \boldsymbol{x}_\phi \tag{20}$$

$$= D^{-1} V^{\mathrm{T}} X_\phi^{\mathrm{T}} \boldsymbol{x}_\phi \tag{21}$$

$$= D^{-1} V^{\mathrm{T}} k_\phi(X, \boldsymbol{x}). \tag{22}$$

This technique is referred to the kernel trick to avoid the calculation in higher dimensional feature space. And popular kernel functions are:

$$k_\phi(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{y}, \tag{23}$$

$$k_\phi(\boldsymbol{x}, \boldsymbol{y}) = \left(1 + \boldsymbol{x}^{\mathrm{T}} \boldsymbol{y}\right)^p, \tag{24}$$

$$k_\phi(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{p^2}\right). \tag{25}$$

5.2 kBPLP

As mentioned in Sec. 5.1, sample background image set can be described as a subspace in higher-dimensional feature space by the kernel PCA. However, it is difficult to perform a backprojection from the feature space to an image space.

Amano et al. have proposed kBPLP nonlinear extension of BPLP using the kernel trick [10]. In the kBPLP, they introduced the following new feature, which stacked pixel value $\boldsymbol{\zeta}$ in occluded region and feature value $\boldsymbol{\xi}_\phi$ of pixel value $\boldsymbol{\xi}$ in observable region.

$$\boldsymbol{y}_i = \begin{bmatrix} \boldsymbol{\xi}_{\phi i} \\ \boldsymbol{\zeta}_i \end{bmatrix}. \tag{26}$$

In the subspace constructed from this feature points $\boldsymbol{y}_i$, since there is a linear relationship between $\boldsymbol{\xi}_\phi$ and $\boldsymbol{\zeta}$, the pixel values in occluded region $\boldsymbol{\zeta}$ can be estimated from feature values of observable region $\boldsymbol{\xi}_\phi$. Especially, when the vector $\Sigma\boldsymbol{x}$, replaced occluded region in an image $\boldsymbol{x}$ with 0, satisfy the followings:

$$\Sigma\boldsymbol{x} = [\ldots, x_{i-1}, 0, x_{i+1}, \ldots], \tag{27}$$

$$\boldsymbol{\xi} = \boldsymbol{x}' = [\ldots, x_{i-1}, x_{i+1}, \ldots], \tag{28}$$

$$k_\phi(\Sigma\boldsymbol{x}, \Sigma\boldsymbol{y}) = k_\phi(\boldsymbol{x}', \boldsymbol{y}'), \tag{29}$$

An estimation equation of occluded values can be represented by:

$$\bar{\Sigma}\hat{\tilde{\boldsymbol{x}}} = \bar{\Sigma} X V \left(V^{\mathrm{T}} K_\phi(\Sigma X, \Sigma X) V\right)^{-1} V^{\mathrm{T}} K_\phi(\Sigma X, \Sigma\tilde{\boldsymbol{x}}). \tag{30}$$

Note that $\bar{\Sigma} = (I - \Sigma)$. In this equation, they use V derived from

$$K_\phi(\Sigma X, \Sigma X) = VD^2V^{\mathrm{T}}, \tag{31}$$

when a occlusion mask Σ changes, the kernel matrix $K_\phi(\Sigma X, \Sigma X)$ and its eigenbases must be recalculated. While bases of BPLP for linear subspace are constant against occlusion mask, it is difficult for kBPLP to estimate image in realtime due to the computational cost.

5.3 Modified kBPLP

The BPLP with linear eigenspace is the method that uses a eigenspace by PCA of sample images as the model, and estimates occluded values by fitting occluded image to the subspace. On the other hand, since kBPLP reconstructs a subspace according to occluded region, the method is not a direct extension of BPLP as to this point.

Hence we consider the following feature Y, which has whole pixel value X and its feature value X_ϕ.

$$Y = \begin{bmatrix} X_\phi \\ X \end{bmatrix} = UDV^{\mathrm{T}} = \begin{bmatrix} U_\phi \\ U_X \end{bmatrix} DV^{\mathrm{T}}. \tag{32}$$

At this time, we obtain:

$$Y^{\mathrm{T}}Y = K_\phi(X,X) + X^{\mathrm{T}}X, \tag{33}$$

And we call it $K_Y(X,X)$. Using this matrix, the point $\boldsymbol{p}$ on the eigenspace corresponding to $\boldsymbol{x}$ is computed by,

$$\boldsymbol{p} = U^{\mathrm{T}}\boldsymbol{y} = D^{-1}V^{\mathrm{T}}K_Y(X,\boldsymbol{x}). \tag{34}$$

In addition, we can compute a backprojection linearly using this feature $\boldsymbol{p}$ as follows:

$$\boldsymbol{x} \approx U_X\boldsymbol{p} = XVD^{-1}\boldsymbol{p}. \tag{35}$$

Similarly, an estimated point $\hat{\boldsymbol{p}}$ of given occluded image $\Sigma\tilde{\boldsymbol{x}}$ can be computed as

$$\hat{\boldsymbol{p}} = \left(D^{-1}V^{\mathrm{T}}K_Y(\Sigma X, \Sigma X)VD^{-1}\right)^{-1} D^{-1}V^{\mathrm{T}}K_Y(\Sigma X, \Sigma\tilde{\boldsymbol{x}}). \tag{36}$$

So we can reconstruct an image from a partially occluded image using eq. (35) and (36). However, this estimation uses V, D derived from

$$K_Y(\Sigma X, \Sigma X) = VD^2V^{\mathrm{T}}, \tag{37}$$

and has heavy kernel matrix computation in eq. (36). Therefore, we apply the iterative projection technique to this estimation to realize faster estimation.

5.4 Rapid Calculation of Modified kBPLP using Iterative Projection

In the modified kBPLP, the estimation process is done linearly because there is a linear relationship between a feature vector $\boldsymbol{y}$ and image vector $\boldsymbol{x}$. So we can apply the same framework of BPLP and the iterative projection method to modified kBPLP.

Regarding an input image $\tilde{\boldsymbol{x}}$ as fully observable ($\Sigma = I$), the projected point on feature space $\hat{\boldsymbol{p}}$ and the backprojected point of it $\hat{\hat{\boldsymbol{x}}}$ can be computed as same as eq. (34) and eq. (35),

$$\hat{\boldsymbol{p}} = D^{-1}V^{\mathrm{T}}K_Y(X,\tilde{\boldsymbol{x}}) \tag{38}$$

$$\hat{\hat{\boldsymbol{x}}} = U\hat{\boldsymbol{p}} \tag{39}$$

$$= XVD^{-2}V^{\mathrm{T}}K_Y(X,\tilde{\boldsymbol{x}}) \tag{40}$$

$$= XK_Y^{-1}(X,X)K_Y(X,\tilde{\boldsymbol{x}}) \tag{41}$$

In this case, we use V, D from

$$K_Y(X,X) = VD^2V^{\mathrm{T}}. \tag{42}$$

Then we iterate the background estimation by projecting to nonlinear eigenspace and backprojection, and replace pixel value in occluded region with estimated using the following recurrence equation:

$$\hat{\boldsymbol{p}}_k \leftarrow D^{-1}V^{\mathrm{T}}K_Y(X,\hat{\boldsymbol{x}}_k) \tag{43}$$

$$\hat{\boldsymbol{x}}_{k+1} \leftarrow \Sigma\tilde{\boldsymbol{x}} + (I-\Sigma)XVD^{-1}\hat{\boldsymbol{p}}_k. \tag{44}$$

Comparing a proposed equation (43) and one of modified kBPLP (36), the matrix affect to $K_Y(X,\hat{\boldsymbol{x}}_k)$ is constant whenever occlusion mask Σ changes, and computation of the kernel matrix and inversion is only needed once when sample images are given. Putting it all together, we update the estimated image using the followings.

$$\hat{\hat{\boldsymbol{x}}}_k \leftarrow XK_Y^{-1}(X,X)K_Y(X,\hat{\boldsymbol{x}}_k) \tag{45}$$

$$\hat{\boldsymbol{x}}_{k+1} \leftarrow \Sigma\tilde{\boldsymbol{x}} + (I-\Sigma)\hat{\hat{\boldsymbol{x}}}_k \tag{46}$$

The proposed algorithm is:

0. Initialize: set $\Sigma_0 = I, \hat{\boldsymbol{x}}_0 = \boldsymbol{0}$,
1. Capturing input image $\tilde{\boldsymbol{x}}$,
2. Replacing former occluded region $(I-\Sigma)$ in the input image with previously estimated image $\hat{\hat{\boldsymbol{x}}}$ (eq. (46)),
3. Updating estimated image $\hat{\hat{\boldsymbol{x}}}$ using the image $\hat{\boldsymbol{x}}$ in step 2, (eq. (45)),
4. Updating occlusion mask Σ by comparing input image and estimated background,
5. Back to step 1.

6 Experiments

At first, using CG sequence same as linear method, we compare the proposed method and modified kBPKP at view of square error and computational time.

Table2 num of dim. of subspace

$k_Y(\boldsymbol{x},\boldsymbol{y})$	p	dim.	c.p.
$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$	–	182	0.990218
$(1+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y})^p+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$	3	219	0.990456
$\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{p^2}\right)+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$	1.5	200	0.990303

In advance, we define the matrix X composed from given background sequence $\{\boldsymbol{x}_i\}_{i=1}^s$.

$$X=[\boldsymbol{x}_1-\bar{\boldsymbol{x}},\ldots,\boldsymbol{x}_s-\bar{\boldsymbol{x}}], \tag{47}$$

where $\bar{\boldsymbol{x}}$ denotes mean image of the sequence, i.e. $\frac{1}{s}\Sigma\boldsymbol{x}_i$. Then we obtained the sample data by normalizing X so as to Frobenius norm of X $\|X\|_{\mathrm{F}}$ is equal to s.

$$X:=\frac{s}{\|X\|_{\mathrm{F}}}X=\frac{s}{\sqrt{\mathrm{tr}(X^{\mathrm{T}}X)}}X \tag{48}$$

In kernel PCA, the relation of the number of dimension and cumulative proportion is differ kernel function by function. So we decided the dimension of eigenspace to the smallest number to exceed 0.99 of c.p.. The threshold to extract unseen object is set to $\tau=0.13$.

The specification of experimental setup is as follows:

PC Dual Xeon 3.6 GHz, 2GB RAM
OS Gentoo Linux (kernel: 2.6.22)
Compiler GCC 4.1.2 (glibc-2.6.1)

6.1 Evaluation using CG

Table 2 shows the dimension of eigenspace with each kernel function.

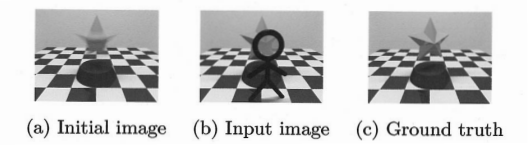
6.1.1 Convergence Speed in Given Occlusion Mask

We used a mean image (Fig. 15(a)) as initial estimated image. And adopt 3-degree polynomial kernel as kernel function k_ϕ, which shows the best performance in preliminary experiment. Thus,

$$k_Y(\boldsymbol{x},\boldsymbol{y})=(1+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y})^3+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}. \tag{49}$$

We input a fixed image (Fig. 15(b)) as input for the iteration, and calculate the error between an estimated image at each time and ground truth image (Fig. 15(c)). The error ϵ is given as follows, where ground truth $\boldsymbol{x}_g$, estimated image $\hat{\boldsymbol{x}}$, and number of pixel n.

$$\epsilon=\sqrt{\frac{(\hat{\boldsymbol{x}}-\boldsymbol{x}_g)^2}{n}} \tag{50}$$



(a) Initial image (b) Input image (c) Ground truth

Fig.15 Test images

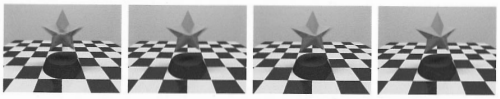


Fig.16 Estimated image by proposed method (poly-3)

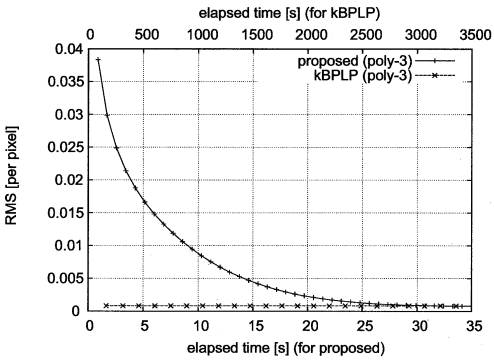


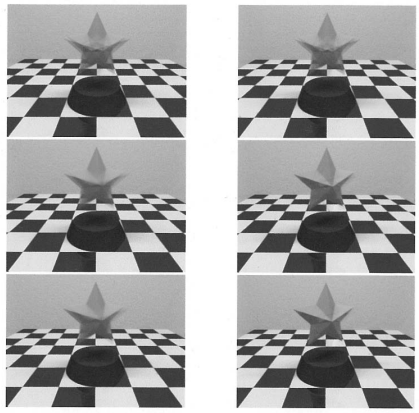
Fig.17 Estimation error in fixed mask

Fig. 16 shows the estimation result of proposed method. We can confirm that the iteration converges to ground truth from the figure. Fig. 17 is the estimation errors of proposed and modified kBPLP. The plotted points describe one iteration and estimation result of proposed method approaches to the estimation result of modified kBPLP. In this figure, horizontal axis denotes the elapsed time, above tics are for kBPLP and bottom for proposed. From the figure, proposed method converged in about 30 seconds while modified kBPKP costs about 160 seconds to estimate the first result. Thus proposed method is 5 times faster in the case of known occlusion mask.

6.1.2 Estimation with Updating Occlusion Mask

In this section, we compare the both method under occlusion region is unknown. We set $\Sigma=I$ (no occluded region) as initial state.

Fig. 18 shows the estimation result of each method. Though modified kBPLP computes better result per



(a) iterative projection (b) modified kBPLP

Fig.18 Estimated image with updating mask

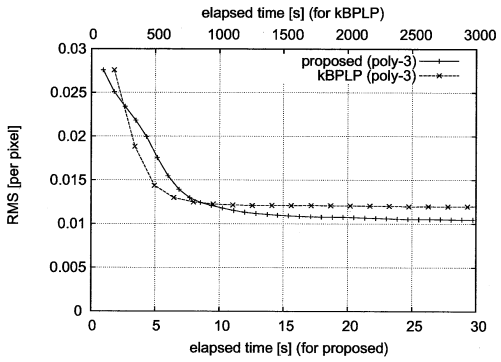


Fig.19 Estimation error in revising mask

iteration, they lives different time scale. In Fig. 19, the time scale for modified kBPLP (above) is 100 times longer than proposed method (below), and proposed method almost converges in 10 seconds, while modified kBPLP takes more than 500 seconds. When the occlusion mask is not given, the final result can differ each other, since the method uses a different mask in iteration.

6.2 Estimation of Real Scene

In this section, we apply proposed method to a real sequence taken by DV camera. And we compare the linear iterative method and the nonlinear one, here.

6.2.1 Experimental Condition

At first, we extract 256 frames without human from a sequence taken in front of a lift as background images (Fig. 20(a)). Then we compute a mean image

(a) Background sequence (b) Input sequence

Fig.20 Background sequence and input sequence.

of them and eigenimages, we decide the dimension of eigenspace so as to satisfy the 0.99 of c.p.. We have 112 dimensional eigenspace (c.p.: 0.990273) at this moment.

We input sequences (Fig. 20(b)) and estimated background image and unseen object region. In this case, we switch input frames per iteration, this is equivalent to compute the result in 1/30 seconds.

6.2.2 Experimental Results

Fig. 21 shows each estimation result at same frames, these frames are one of parts where the linear method tends to much estimation error. The frames exist around the frame in 3rd row of Fig. 20(b). There is an object having similar intensity of lift's door, thus we find stripe-like estimation error with liner estimation method. On the other hand, nonlinear version has lesser error comparing to the linear method.

Fig. 22 shows extraction result of unseen object. Constructing the subspace from a sequence without human, proposed method successfully extract the hu-

(a) Innerproduct (linear)　(b) Gaussian ($p = 0.7$)

Fig.21 Estimated image of conventional and proposed

(a) Innerproduct　(b) Gaussian ($p = 0.7$)

Fig.22 Mask image of conventional and proposed

man region while moving door is ignored.

7 3-D Intrusion Detection System

Intrusion detection techniques (e.g., person–machine collision prevention, off-limits area observation, etc.) are important for establishing safe, secure societies and environments. Today, equipment which detects the blocking of a light beam, referred to as a light curtain, are widely used for this purpose. Although the light curtain is useful to achieve very safe environments which were previously considered dangerous, it is excessive for widespread applications. For example, the light curtain method requires us to set equipment at both sides of a rectangle for detection, which leads to higher cost, limited shape of the detection plane and set-up difficulty. In the meantime, surveillance cameras have been installed into many various environments; however, the scenes observed by these cameras are used only for recording or visual observation by distant human observers, and they are merely used to warn a person in a dangerous situation or to immediately halt a dangerous machine. There are many computer enhancements that recognize events in a scene [3], but it is difficult to completely detect dangerous situations, including unexpected phenomena. Furthermore, we do not have sufficient knowledge and methodologies to use the recognition result from these systems to ensure safety. Therefore, our proposed system simply detects an intrusion in a specific area in 3D space using multiple cameras. We believe this system will help establish a safe and secure society.

As mentioned above, flexibility and ease in setting up the equipment and detection region are important factors to the cost and practical use. However, there are two problems in image based intrusion detection: one is the necessity of the complex and nuisance calibration for a multiple camera system, and the other is the intuitiveness for defining a restricted area. Thus, we propose a method to complete the calibration and the restricted area definition simultaneously by simply moving a colored marker in front of the cameras.

8 Characterization and Simplification of the Intrusion Detection Problem

In the last decade of computer vision, there have been many studies to measure or recognize a scene taken by cameras in an environment. In particular, methods to extract or track a moving object in an image have been investigated with great effort and have rapidly progressed. In most of this research, the region of an object can be detected without consideration of the actual 3D shape. Therefore, although these techniques may be used for rough intrusion detection, they cannot handle detailed motion and deformation, such as whether a person is reaching for a dangerous machine or an object of value. On the other hand, there has been other research to reconstruct the whole shape of a target object from images

taken by multiple cameras. Using this method, it is possible to detect the intrusion of an object in a scene by computing the overlapping region of the restricted area and target object. This approach is not reasonable because the reconstruction computation generally needs huge CPU and memory resources, and, as described later, the approach involves unnecessary processes to detect an intrusion. In addition, it is not easy for users to set up such a system because the cameras must be calibrated precisely. Thus, we resolve these issues by considering two characteristics of the intrusion detection problem.

The first is the projective invariance of the observed space in intrusion detection. The state of intrusion, that is, the existence of an overlapping region of a restricted area and object, is invariant if the entire scene is projectively transformed. Hence, we can use weak calibration, instead of full calibration, to detect an intrusion. Furthermore, setting the restricted area can be done simultaneously with the calibration, because the relationship between the area and cameras can also be represented in a projective space. Although the whole shape of an intruding object has projective indefiniteness, it doesn't affect the detection of intrusion.

The second characteristic is that a restricted area is always a closed region. Consequently, we do not have to check the total volume of a restricted area; it is sufficient to observe only the boundary of the restricted area. This manner of thinking is one of the standard approaches for ensuring safety, and is also adopted by the above-mentioned light curtain. Our system detects an intrusion by projecting the silhouette on each camera image onto the boundary plane, then computing the common region of all the silhouettes. This common region on the boundary plane is equivalent to the intersection of the reconstructed shape of an object by the visual hull method and the shape of the boundary plane.

9 Detection of an Intruding Object

9.1 The Visual Hull Method

To decide if an object exists in a specific area, the 3D shape of the object in the scene must be obtained. We adopt the visual hull method for shape reconstruction. In the visual hull method, the shape of an object can be reconstructed by computing the intersection of all cones, which are defined by a set of rays through the viewpoint and one point on the edge of the silhouette on an image plane. This method has the advantage that the texture of an object does not affect the reconstructed shape, because there is no necessity to search the corresponding points between images. However, this method tends to reconstruct a shape larger than the real shape, particularly with concave surfaces. Also, an invisible area from any of the cameras can also make it impossible to measure the shape. Although this is a common problem for image-based surveillance, our approach is always safe because the proposed system handles the invisible area as a part of the object.

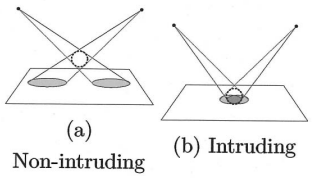


Fig.23 Intrusion detection based on the existence of an intersection

Although the visual hull method has great merit for intrusion detection, it needs large computational resources for the set operation in 3D space. Therefore, it is difficult to construct an intrusion detection system that is reasonable and works in real time.

9.2 Section Shape Reconstruction on a Sensitive Plane

As mentioned above, it is sufficient to observe only sensitive planes, the boundary of a restricted area, for intrusion detection. Accordingly, only the shape of the intersection region on a sensitive plane is reconstructed by homography based volume intersection [11]. In this case, the common region of projected silhouettes on the plane is equivalent to the intersection of the visual hull and the plane. Therefore, when an object exceeds a sensitive plane, the common region appears on the plane (Fig. 23). In this way, the 3D volumetric intrusion detection problem is reduced to efficient processes of inter-plane projection and common region computation in 2D space.

9.3 Vector Representation of the Silhouette Boundary

The visual hull method only uses information of the boundary of a silhouette. Therefore, the amount of data can be decreased by replacing the bound-

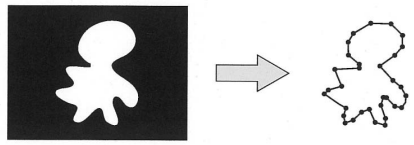


Fig.24 Vector representation of silhouette contours

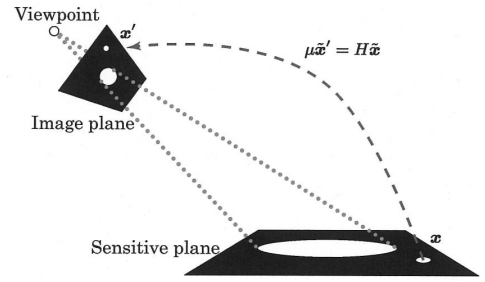


Fig.25 Homography between two planes

ary with vector representation by tracking the edge of the silhouette in an image (Fig. 24). In the vector representation, the projection between planes is achieved by transforming a few vertices on the edge. It is easy for the common region computation to decide whether each vertex is inside or outside the other contour. With this representation, we are able to reduce the computational costs for the transformation and common region calculation, and it is not necessary to adjust the resolution of the sensitive plane to compute the common region with sufficient preciseness. In a distributed vision system, it is possible to reduce the amount of communication data because many camera-connected nodes extract silhouette contours and one host gathers the silhouette data and computes the common region.

9.4 Procedure of the Proposed System

For summarization, intrusion detection on the boundary is realized by the following steps:

1. Defining sensitive planes.
2. Extracting the silhouette of a target object.
3. Generating the vector representation from the silhouette.
4. Projecting each silhouette vector onto sensitive planes.
5. Computing the common region.
6. Deciding the intrusion.

In the next section, we discuss step 1.

10 Construction of a Restricted Area

Using the following relationship, the silhouette of an object on an image plane can be transformed onto a sensitive plane. Let $\boldsymbol{x}(\in \Re^2)$ be the coordinate of a point on a sensitive plane. The corresponding point on the image plane can be calculated, as follows:

$$\mu\tilde{\boldsymbol{x}}' = H\tilde{\boldsymbol{x}}, \tag{51}$$

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{52}$$

where $\tilde{\boldsymbol{x}}$ is the notation of homogeneous coordinates of $\boldsymbol{x}$. Matrix H is referred to as a homography matrix, which has only 8 DOF for the scale invariant.

From Eq. (52), the homography matrix can be determined by more than four pairs of corresponding points which are specified by a user. However, this method is a burden to users, who must set up the system in proportion to the product of the number of cameras and the number of sensitive planes. Also, it is not easy for users to define an arbitrarily restricted area without a reference object. Therefore, in the next section, we introduce a more convenient method for setting a sensitive plane.

10.1 Relation of the Homography Matrix and Projection Matrix

Instead of specifying the points on an image from a camera view, it is easy to place a small marker in the real observed space so that we obtain the corresponding points using cameras. However, in this case, it is difficult to point out the four points on a plane in real 3D space. Therefore, we consider the method in which users input enough 'inner' points of the restricted area so that the system automatically generates a set of sensitive planes which cover all the input points. Now, when we know the projection matrix P, which translates a coordinate in a scene onto an image plane, the relationship between $\boldsymbol{X}$, a point in 3D space, and $\boldsymbol{x}$, a point on an image plane, is given by

$$\lambda\tilde{\boldsymbol{x}} = P\tilde{\boldsymbol{X}}. \tag{53}$$

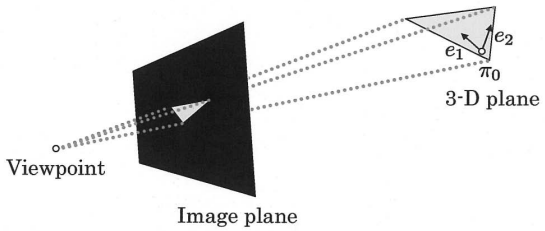


Fig.26 A plane in 3D space projected onto the image plane

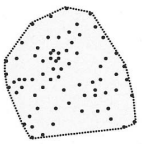


Fig.27 Points and their convex hull (2D case)

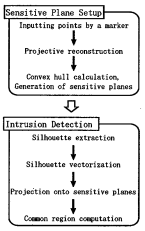


Fig.28 Flow chart of the proposed system

Likewise, as shown in Fig. 26, a point on the plane Π in 3D space is projected onto the image plane as follows.

$$\lambda\tilde{\boldsymbol{x}} = P(\alpha\tilde{\boldsymbol{e}}_1 + \beta\tilde{\boldsymbol{e}}_2 + \tilde{\boldsymbol{\pi}}_0) \tag{54}$$

$$= P\begin{bmatrix}\tilde{\boldsymbol{e}}_1 & \tilde{\boldsymbol{e}}_2 & \tilde{\boldsymbol{\pi}}_0\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\\ 1\end{bmatrix} \tag{55}$$

where $\boldsymbol{e}_1, \boldsymbol{e}_2$ are bases of Π in 3D, and $\boldsymbol{\pi}_0$, (α, β) are the origin and parameter of Π, respectively.

From Eq. (55), we can compute the homography matrix between an arbitrary plane in 3D and the image plane by

$$H = P\begin{bmatrix}\tilde{\boldsymbol{e}}_1 & \tilde{\boldsymbol{e}}_2 & \tilde{\boldsymbol{\pi}}_0\end{bmatrix}. \tag{56}$$

Therefore, when we know the projection matrices of the cameras and are given three or more points on a plane in 3D, it is possible to define the plane as a sensitive plane, except in a singular case (e.g., all points are on a line.). For example, the three adjacent points $\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{X}_2$ make one plane:

$$\begin{cases}\boldsymbol{e}_1 := \boldsymbol{X}_1 - \boldsymbol{X}_0,\\ \boldsymbol{e}_2 := \boldsymbol{X}_2 - \boldsymbol{X}_0,\\ \boldsymbol{\pi}_0 := \boldsymbol{X}_0.\end{cases} \tag{57}$$

As mentioned above, a set of homogeneous matrices can be automatically generated from each given camera projection matrix and the vertices of the sensitive planes in 3D space. However, in our problem, we assume both the camera parameters and 3D points are unknown. Therefore, we have to calculate both by the projective reconstruction technique [12] using the given corresponding points between cameras.

10.2 Generation of Sensitive Planes from Reconstructed Inner Points

Now we have the projection matrices and many reconstructed 3D points which reside in the restricted area, so we have to determine enough pairs of 3D points as the vertices of the sensitive planes. We compute the convex hull, which handles all the input points for generating sensitive planes. The system defines a restricted area as the boundary of the convex hull computed using qhull [13] (Fig. 27). The reconstructed points, except on the boundary, are removed because they do not make a sensitive plane.

11 Experiment

We implemented the proposed intrusion detection method in a multiple-camera system. From the users' view, the system has two phases: one is setting the sensitive planes and the other is executing intrusion detection (see Fig. 28). Since the latter phase is completely automated, users need only to input corresponding points with a simple marker. Therefore, any complicated technical process, such as calibration of the multiple camera system, is already managed for setting the actual sensitive plane.

In this experiment, we confirm the proposed method of sensitive plane generation and intrusion detection in projective space. The system consists of three cameras (SONY DFW-VL500) and a PC (Dual Intel Xeon @ 3.6 [GHz] w/ HT). We set the cameras at an appropriate position so that each camera can observe the whole region to detect an intrusion (Fig. 29).

11.1 Input of Sensitive Plane using a Colored Marker

We use a simple red colored marker to input corresponding points among all image planes. First, the user specifies the color of the marker by clicking on the area of the marker, then the system computes the mean and the variance of the area. According to the Mahalanobis distance between an input color at

Fig.29 Cameras and observed space

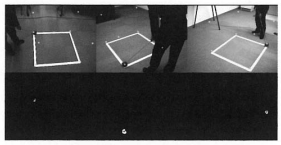


Fig.30 Setting of restricted area (top: camera view, bottom: extracted marker position)

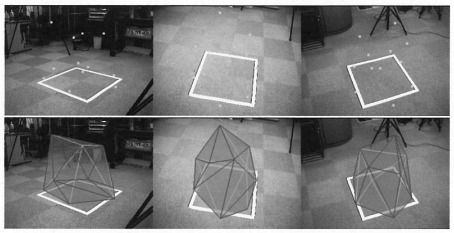


Fig.31 Inputted points and generated convex hull

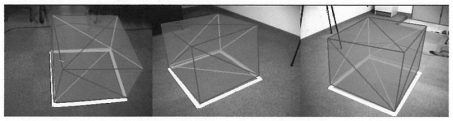


Fig.32 Generated sensitive planes

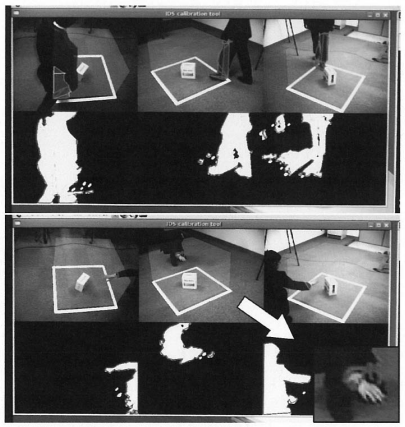


Fig.33 Detection result (top: intrusion of a leg, bottom: intrusion of a wrist, reaching for the object)

each pixel and the reference color, the system extracts similar pixels by thresholding the distance. For noise reduction, the center of gravity of the largest region is calculated as the marker position (Fig. 30). The user in a real scene moves the marker position to set up the restricted area.

Fig. 31 shows an example of the sensitive planes generated from inputted points. In this case, 16 sensitive planes are generated from 10 of 12 inputted points, and remaining two points of them are removed because they are inside of the convex hull.

11.2 Intrusion Detection

In this experiment, we input eight points on the vertices of a hexahedron. Fig. 32 depicts the generated set of sensitive planes from the input points. In this case, 12 planes are generated by the proposed method. The result of the intrusion detection is shown in Fig. 33.

In our implementation, we use a statistical background subtraction method [14] to extract a silhouette of the object from an image. The silhouette is transformed into vector representation by tracking the edge and projected onto each sensitive plane. Then, the system computes the common region on each sensitive plane. In the figure, the leg or wrist of the intruder is detected on the boundary of the restricted area. Although one can see some false positive extraction areas of the silhouette (e.g., the shadow cast in the image of the top row, third column), our method has a robustness against such noise because of the common region computation of all extracted silhouettes.

12 Conclusions

In this paper we propose the intrusion detection system for versatile purpose. The system consists of extraction method of an unseen object region from 2-D image and 3-D intrusion detection method by integrating the regions using multiple uncalibrated cameras.

On unseen object region extraction, we modeled a dynamic scene, which contains moving object as a background, by eigenvectors derived by PCA of a background sequence. This approach successfully enables simultaneous estimation of background image and extraction of unseen object region in realtime.

The linear estimation method works well in most case, however, sometimes it have much estimation er-

ror in particular situation because of nonlineality of image sequences. Therefore, we extend the method to the higher dimensional feature space using nonlinear mapping and kernel trick. This proposed method indicates more stable results in real scene.

On the other hand, we introduced an intrusion detection system for an arbitrary 3D volumetric restricted area using uncalibrated multiple cameras. Although our algorithm is based on the visual hull method, the whole shape of intruding object does not need to be reconstructed; instead, the system can efficiently detect an intrusion by perspective projections in 2D space.

In general, an intricate calibration process for a distributed camera system has been necessary, but the proposed system automatically calibrates the cameras when users input corresponding points through the restricted region setting. Furthermore, the user does not need any previous knowledge about cameras because of the projective reconstruction. Also, any combination of cameras having varying intrinsic camera parameters can be used. Therefore, non-expert users can intuitively operate the proposed system for intrusion detection by only setting the cameras in place.

References

[1] Hu, W., Tan, T., Wang, L. and Maybank, S.: A survey on visual surveillance of object motion and behaviors, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 34, No. 3, pp. 334–352 (2004).

[2] Matsumura, A., Iwai, Y. and Yachida, M.: Tracking Peopole and Action Recognition from Omnidirectional Images, *Systems and Human Science - For Safety, Security, and Dependability Selected Papers of the 1st International Symposium (SSR2003)*, Ch.36, pp. 491–500 (2005).

[3] Collins, R. T., Lipton, A. J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., Burt, P. et al.: A system for video surveillance and monitoring (VSAM project final report), Technical report, CMU Technical Report CMU-RI-TR-00 (2000).

[4] Elgammal, A., Harwood, D. and Davis, L.: Nonparametric model for background subtraction, *FRAME-RATE Workshop, IEEE* (1999).

[5] Ivanov, Y., Bobick, A. and Liu, J.: Fast Lighting Independent Background Subtraction, *International Journal of Computer Vision*, Vol. 37, No. 2, pp. 199–207 (2000).

[6] Javed, O., Shafique, K. and Shah, M.: A Hierarchical Approach to Robust Background Subtraction Using Color and Gradient Information, *Proc. Workshop on Motion and Video Computing*, pp. 22–27 (2002).

[7] Oliver, N., Rosario, B. and Pentland, A.: A Bayesian Computer Vision System for Modeling Human Interactions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 831–843 (2000).

[8] Amano, T. and Sato, Y.: Image Interpolation using BPLP Method on the Eigenspace (in Japanese), *IEICE Journal*, Vol. J85-DII, No. 3, pp. 457–465 (2002).

[9] Schölkopf, B., Smola, A. and Müller, K.-R.: Kernel principal component analysis, *Advances in Kernel Methods-Support Vector Learning*, pp. 327–352 (1999).

[10] Amano, T. and Sato, Y.: Image Interpolation by the High Dimensional Nonlinear Projection Using kBPLP Method(in Japanese), *IEICE Journal*, Vol. J86-D-II, No. 4, pp. 525–534 (2003).

[11] Wada, T., Wu, X., Tokai, S. and Matsuyama, T.: Homography Based Parallel Volume Intersection: Toward Real-Time Volume Reconstruction Using Active Cameras, *Proc. Computer Architectures for Machine Perception 2000*, pp. 331–339 (2000).

[12] Mahamud, S. and Hebert, M.: Iterative Projective Reconstruction from Multiple Views, *Proc. CVPR*, Vol. 2, pp. 430–437 (2000). SC, U.S.A.

[13] Barber, C. B., Dobkin, D. P. and Huhdanpaa, H.: The Quickhull Algorithm for Convex Hulls, *ACM Trans. Mathematical Software (TOMS)*, Vol. 22, No. 4, pp. 469–483 (1996). http://www.qhull.org.

[14] Horprasert, T., Harwood, D. and Davis, L. S.: A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection, *Proc. ICCV Frame-Rate Workshop (ICCV'99)*, pp. 1–19 (1999).