Visual Surveillance Using Less ROIs of Multiple Non-calibrated Cameras

Takashi Nishizaki, Yoshinari Kameda, and Yuichi Ohta

Graduate School of Systems and Information Engineering,
University of Tsukuba, 1-1-1 Tennoudai, Tsukuba, Ibaraki, 305-8573, Japan
{tanishi, kameda, ohta}@image.esys.tsukuba.ac.jp
http://www.image.esys.tsukuba.ac.jp

Abstract. With a large number of surveillance cameras, it is not an easy task to determine which camera should be monitored and which region of the camera images should be checked so that all the activities and/or events in a scene are examined. We present a new method to realize effective visual surveillance under an environment in which a number of non-calibrated fixed surveillance cameras are being operated. We also show two applications that are useful for surveillance tasks based on our proposed method. One is *"suggestion of associative blocks"*, and the other is *"dominant camera selection"*. Our approach exploits co-occurrence between two regions of interest (ROIs) over the surveillance cameras, and it needs neither calibration nor supervised training. We have conducted preliminary tests with forty cameras installed in a room and a corridor next to the room, and some promising results of the two applications are shown in this paper.

1 Introduction

Recently, there are increasing social demands to observe and detect usual and/or unusual events by exploiting cameras in various environments. Such a surveillance camera system is thought to be useful for security in public areas, road traffic monitoring, and so on. As surveillance cameras are being more and more installed and utilized in a scene for surveillance task, sometimes it becomes impractical and cumbersome to remember their locations and their visible areas. In addition, since surveillance cameras cannot always be set perpendicular to the ground/floor, images from surveillance cameras may not be comfortable to recognize instantly. Therefore, with a large number of surveillance cameras, it is not an easy task for people to recognize which camera should be monitored and which region of the camera images should be checked so that all the activities and/or events in a scene can be examined. This problem becomes prominent especially when a number of cameras are widely scattered because it is impractical to calibrate them consistently.

Fig. 1 shows snapshots of 36 cameras taken simultaneously. The cameras are installed in a room and a corridor adjacent to the room. It is apparently difficult to locate a person in the images. As the number of the surveillance cameras increases, maintaining the consistency of their geometric information (their locations and directions) is cumbersome. Therefore, there is a demand

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3851, pp. 317–327, 2006.

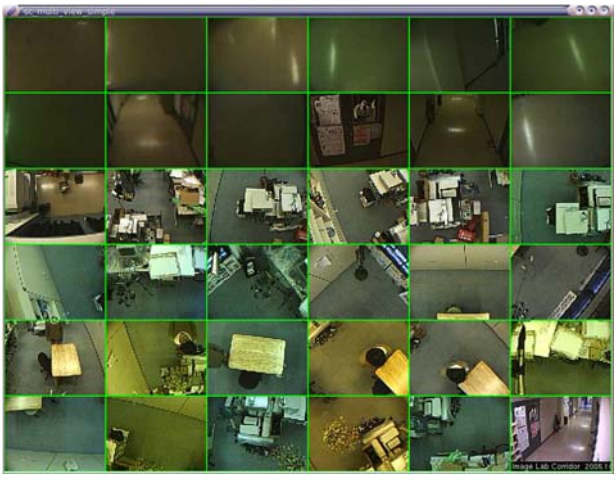

Fig. 1. An Example of Multi-view Videos (36 cameras)

to a sophisticated visual assistance method that can support visual surveillance tasks against a large number of cameras, which are not geometrically calibrated precisely.

We present a new method to realize effective visual surveillance under an environment in which a number of non-calibrated fixed surveillance cameras are being operated. We also show two applications that are useful for surveillance task based on our proposed method. One application of the method is "suggestion of associative blocks". If there is an event that an user should check and he/she has noticed it by checking one region on a surveillance camera on the system, the system can point out regions of different cameras that are helpful to examine the event. The other application is "dominant camera selection". Out of many surveillance cameras, our method can tell which cameras are worth watching in general case.

Our approach exploits co-occurrence between two regions of interest (ROIs) over the surveillance cameras, and it needs neither calibration nor supervised training. We divide camera images into small blocks. Each small block has foreground regions when it captures motions in a scene. Our system first eliminates redundant blocks that apparently do not contribute to event recognition. The elimination algorithm consists of two stages. In the first stage, the blocks that never detect motions are eliminated. Then, in the second stage, we exploit PCA to sweep out the blocks that do not contribute to describe events. We call the remaining blocks regions of interest (ROIs). The system then calculates co-occurrence of any pair of ROIs in which foreground regions related with an event are found simultaneously. Once the co-occurrence matrix is obtained, it can determine a set of ROIs that should be taken care of when an event in focus is found in a certain ROI. In other words, the ROIs are associative to the specified ROI. In addition, dominant camera selection can be conducted based on the co-occurrence matrix.

The rest of the paper is formed as follows. In section 2, recent reseaches related with our research are mentioned. Section 3 explains our surveillance system. Section4 describes the elimination algorithm of (small) blocks in surveillance camera images. In section 5, we show two applications, *"suggestion of associative blocks"* and *"dominant camera selection"*. The concluding remarks are shown in section 6.

2 Related Works

There are many visual surveillance systems for human tracking, traffic monitoring, and detection of unusual objects. In order to cover large area and/or to track objects in complex motion, surveillance systems uses multiple cameras. As shown in previous works, the multi-camera surveillance systems usually rely on manual camera calibration [1][2][3][4] or complex automated calibration method[5]. The surveillance systems with calibrated cameras can surely provide accurate geometry of objects in an environment. However, manual camera calibration is too cumbersome to cope with large-scale surveillance systems, and it is impractical to apply fragile automated calibration methods to such systems. Therefore, there are many demands for surveillance methods that only assume rough geometry information of cameras.

In order to track moving objects on surveillance videos, and to know where to see in videos for surveillance tasks, correspondences in videos captured by cameras are thought to be useful. Therefore, many methods that have correspondence models and estimate correspondences of locations or trajectories of moving objects have been proposed [6][7][8][9]. We also use the correspondences to calculate co-occurrence of objects observed in multiple cameras. Our proposed method is different on the point that it is a monitoring support method and can be applied to a large-scale camera system easily.

3 Surveillance System

In this section, we present a framework of our multi-camera surveillance system, and discuss image features to be used for estimating correspondence.

3.1 Camera Network System

Fig. 2 shows a framework of our system and Fig. 1 shows an example of multi-view videos captured by the system. The system consists of multiple network-cameras(web-cameras) and multiple PCs for image processing. We employed the off-the-shelf web cameras because of the following advantages.

- Installation flexibility: One camera only requires one LAN cable (that also provides power to each camera) at its mounting point.
- Process scalability: We can change the number of cameras assigned to one PC easily.

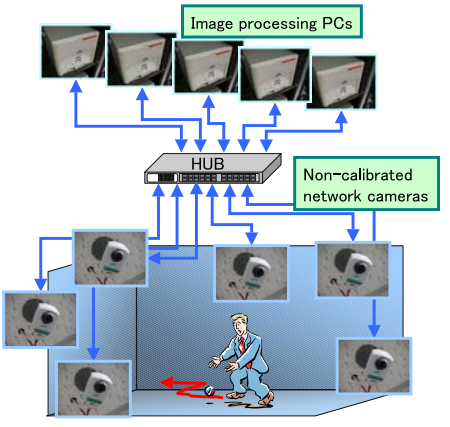

Fig. 2. A Framework of Our Surveillance System

On the other hand, they have following problems.

- It is difficult to synchronize videos.
- Frame rate of captured videos is unstable because it may be affected by network congestion.

To estimate co-occurrence matrix and apply it for the surveillance, we must take care of these factors. In our current implementation, however, synchronization of cameras does not matter because the cameras can output 10.0 to 30.0 fps and the motions in a scene are not so fast compared with the frame rate. Network congestion can also be avoided if network hubs are properly connected so as not to exceed the maximum bandwidth for each connection. Our system consists of 45 cameras and 22 PCs currently. We are planning to extend the system to 80 cameras.

3.2 Image Feature

We consider only basic image features so that our method can cover any kinds of events. Cameras capture images in RGB format and the system divides them into small rectangular blocks. An image is divided into R small blocks (currently, $R = 64$).

Fig. 3. Image Feature of One Captured Image. The left image is an input image and the right image is a display of its saved data.

In our research, the system extracts foreground regions by calculating background subtraction for input images. It calculates the mean intensity value of the foreground regions in each small block, and stores it in INT-type 32 bit data format. Fig. 3 shows an image of one saved data. The data is saved in compressed format. The data size depends on situations. For example, we captured a scene for about 290 hours by 4 cameras, and the compressed data size was about 1.3 GBytes.

4 ROI Selection

In this section, we present a method to select ROIs from the small blocks.

4.1 Data Structure

We define an event vector $\mathbf{x}(t)$ at time t $(1 \leq t \leq T)$, where T is a number of observed event vectors.

$$\mathbf{x}(t) = \{\ x_1(t),\ \cdots\ ,\ x_i(t),\ \cdots\ ,\ x_N(t)\ \} \tag{1}$$

The size(number of dimensions) of the vector is $N = C \times R$, where C is a number of cameras and R is a number of small blocks in an image. Each event vector represents which camera and where in the captured image objects are observed. Each element $x_i(t)$ denotes a feature of an object detected in a region i at time t, and represents what object is observed there. We use the mean intensity values of the small blocks in an image as image features. Note that other features can also be applied in our method.

We call the small blocks that can contribute to event recognition "regions of interest (ROIs)". ROIs are obtained by eliminating redundant blocks among all the N blocks, and the elimination algorithm consists of two stages. In the first stage, the blocks that never detect foreground regions are eliminated. In the second stage, we exploit principal component analysis (PCA) to sweep out the blocks that do not contribute to describe events.

4.2 Block Elimination Based on Foreground Region Detection

First, a mean vector $\mathbf{M}$ is calculated for the input N dimensional event vector $\mathbf{x}(t)$ for $1 \leq t \leq T$ before the block elimination process starts. If M_i that is a mean of features observed in a small block i is zero, the block i is eliminated because it means the region i never detects any motions for all the T frames. Then, we get N' $(N' \leq N)$ dimensional vector $\mathbf{x}'(t)$ by eliminating the blocks that are useless to detect foreground regions.

4.3 Block Elimination by PCA

After the first stage, we apply principal component analysis (PCA) to $\mathbf{x}'$ by using the variance-covariance matrix $\mathbf{V}'$. PCA is a multivariate procedure that rotates

the data in a multi-dimensional space so that variances projected onto the new axes have large variability. It is mainly used for dimensionality reduction. The resultant new rotated axes are called principal axes of $\mathbf{x}'$, and after applying the PCA, principal axes $z_k (1 \leq k \leq N')$ are given by linear combinations of the original variables as shown in the following equations.

$$\mathbf{z} = \mathbf{A}\,\mathbf{x}' \tag{2}$$

$$z_k = a_{1k}x'_1 + a_{2k}x'_2 + \cdots, a_{N'k}x'_{N'} \tag{3}$$

We select a set of variables $\{x'_j\}$ that have larger weight $\{a_{jk}\}$ for more significant principal axes $\{z_k\}$. The principal axes $\{z_k\}$ that have higher contribution ratio are thought to be useful both to recognize and to classify the original data. The followings are the detailed description of block elimination algorithm using PCA.

Step 1: Sorts the principal axes z_k by contribution ratios p_k. A contribution ratio p_k indicates how the principal component z_k represents data better, and it is represented by a variance λ_k of z_k.

$$p_k = \frac{\lambda_k}{\sum_{l=1}^{N'} \lambda_l} \tag{4}$$

Step 2: Calculates accumulated contribution ratio by

$$c_k = \sum_{l=1}^{k} p_l \tag{5}$$

and selects the principal components z_k whose accumulated contribution ratios are larger than a threshold c_{th}. Currently, c_{th} is set to 0.9. We denote the selected principal axes by $\{z'_k\}$.

Step 3: Given a principal axis z'_k, the method calculates the mean value $\bar{a}_k$ of $\{a_{jk}\}$ that are coefficients of $\{x'_j\}$. A score s_j of the variable x'_j obtains the contribution ratio p'_k of z'_k when the coefficient a_{kj} is larger than $\bar{a}_k$.

Step 4: Apply Step 3 to all the principal components $\{z'_k\}$. Calculate the mean value $\bar{s}$ of the scores $\{s_j\}$. Then, select the variables x'_j whose score s_j are higher than $\bar{s}$. Finally, block j corresponding to x'_j is regarded as a ROI.

4.4 Experimental Results of ROI Selection

Fig. 4 shows a layout of cameras in an experiment environment, and Fig. 5,6 show resultant ROIs. In the figures, the small blocks marked in a bright color are ROIs, where people walking around in a room were observed frequently. In the case of 12 cameras(Fig. 6), 525 blocks were selected from 768 blocks in the elimination process of the first stage, and 253 blocks were selected in the elimination process of the second stage. Some ROIs extracted in the case of 4 cameras(Fig. 5) were eliminated in the case of 12 cameras(Fig. 6) because they became less important than the other ROIs in the case of 12 cameras.

In the experiments, Calculation of ROI extraction was conducted on a PC of Pentium4 2.80 GHz, and its memory size is 1.0 GByte. We applied the ROI

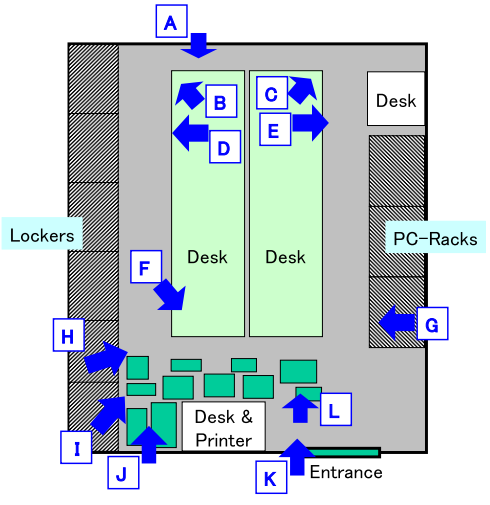

Fig. 4. Camera Layout. Alphabets indicate camera names and arrows show their directions.

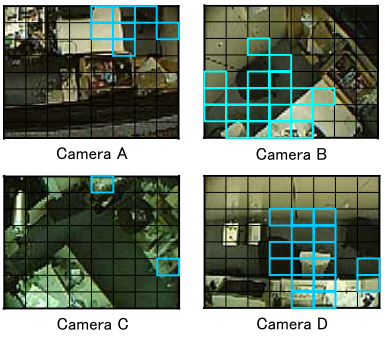

Fig. 5. Resultant ROIs (4 cameras)

extraction method to a scene of two hour length. In the case of 4 cameras, the calculation needed 181.70 seconds; 48.27 seconds to calculate a mean vector $\mathbf{M}$, 133.30 seconds to calculate a variance-covariance matrix $\mathbf{V}'$, and 0.13 seconds to eliminate redundant blocks. In the case of 12 cameras, the calculation spent 1347.19 seconds; 147.7 seconds to calculate a mean vector, 1190.6 seconds to calculate a variance-covariance matrix, and 8.89 seconds to eliminate redundant blocks.

Currently, we are exploring an on-line clustering method for event vectors using extracted ROIs. We expect that the clustering method can be applied to a large-scale camera network because our redundancy elimination algorithm reduces the data size to be processed to a great extent.

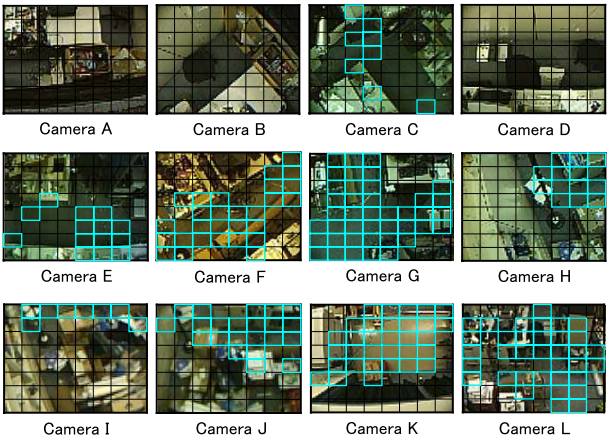

Fig. 6. Resultant ROIs (12 cameras)

5 Visual Surveillance Support Applications

Once the ROIs are calculated, the system can provide various support functions for visual surveillance. Two useful applications are introduced in this section.

5.1 Suggestion of Associative Blocks

One application of the method is suggestion of associative blocks. If there is an event that a user should examine and he/she notice it by having checked a certain region on one surveillance camera, the system can select associative ROIs that are helpful to examine the event.

To select the associative ROIs, we calculate the co-occurrence between two ROIs m, n. A feature value observed in a ROI m is shown as $y_m(t)$ $(1 \leq t \leq T)$, and a set of y_m is shown as a following vector.

$$\mathbf{y}_m = \{\ y_m(1),\ \cdots\ ,\ y_m(t),\ \cdots\ ,\ y_m(T)\ \} \tag{6}$$

We calculate a correlation value c_{mn} $(0 \leq c_{mn} \leq 1)$ by the following equations, and use it as a measure of the co-occurrence between two ROIs.

$$c_{1mn} \ = \ \frac{\mathbf{y}_m \cdot \mathbf{y}_n}{|\mathbf{y}_m||\mathbf{y}_n|} \ = \ \frac{\sum^T y_m(t) y_n(t)}{|\mathbf{y}_m||\mathbf{y}_n|} \tag{7}$$

$$c_{2mn} \ = \ \begin{cases} \frac{|\mathbf{y}_m|}{|\mathbf{y}_n|} \ if\ |\mathbf{y}_m| < |\mathbf{y}_n| \\ \frac{|\mathbf{y}_n|}{|\mathbf{y}_m|} \ \text{otherwise} \end{cases} \tag{8}$$

$$c_{mn} \ = \ c_{1mn}\ c_{2mn} \tag{9}$$

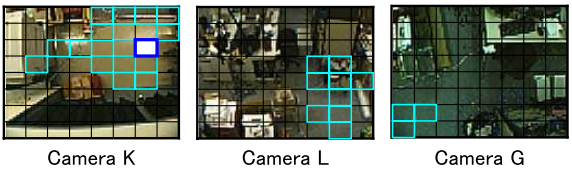

Fig. 7. Associative ROIs (1). The regions with bright frames have high co-occurrences with a white region.

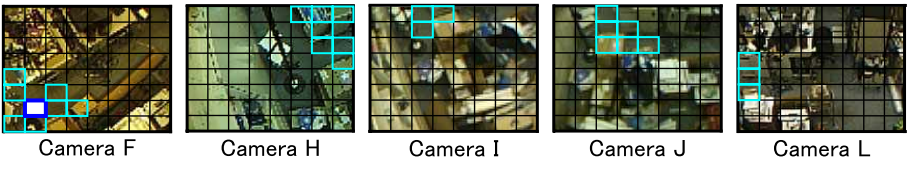

Fig. 8. Associative ROIs (2). The regions with bright frames have high co-occurrences with a white region.

We calculated the co-occurrence matrix of the 12 cameras shown in section 4.4. In Fig. 7 and Fig. 8, the regions with bright frames have high co-occurrences with the white region shown in Camera K and F respectively. The result means that if a user is interested in some motions in white block, the system suggests the user to check the brightly framed blocks too because it is likely to find something in them when some motions are found in the white block.

5.2 Dominant Camera Selection

The other application is dominant camera selection, which means the system can tell which cameras are worth watching in general case. It is useful as the number of cameras C becomes larger.

First, the system sorts all the pairs of any ROIs by their co-occurrences. Then, it selects the upper pairs that have higher co-occurrences, and increments the score u_c of the camera c $(1 \leq c \leq C)$ when the camera c has a block in the pairs. The system selects cameras whose scores are higher than the mean score $\bar{u}$. The selected cameras will be dominant for recognition purpose.

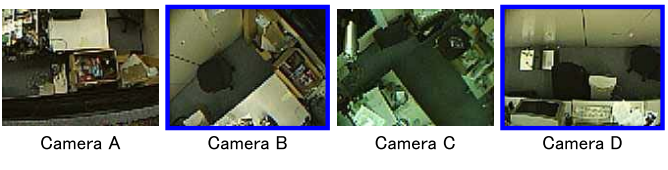

Fig. 9. Dominant Cameras (2 cameras were selected from 4 cameras)

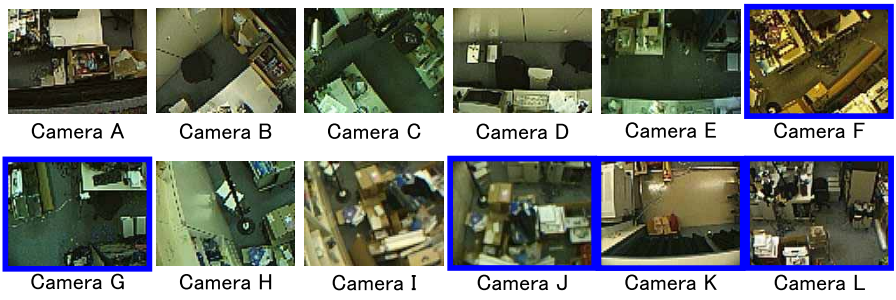

Fig. 10. Dominant Cameras (5 cameras were selected from 12 cameras)

Fig. 9 and Fig. 10 are the experimental results of the dominant camera selection. In the case where there are only 4 cameras (A, B, C, and D), the 2 cameras (B and D) are suggested to be checked for coming events (Fig. 9). On the other hand, if the system has 12 cameras, the system then suggests to check the 5 cameras (F, G, J, K, and L) (Fig. 10).

6 Conclusion

We presented a method to realize effective visual surveillance support under an environment in which a number of non-calibrated fixed surveillance cameras are being operated. Our method divides all camera images into small blocks and selects some blocks that can capture what kind of event is going on. We exploited PCA-based region selection algorithm, and succeeded in presenting useful data expression that can result in achieving two promising visual surveillance support applications; "suggestion of associative blocks" and "dominant camera selection".

As future works, we should examine the relevance of the ROI selection algorithm and the co-occurrence calculation method. In addition, we need to verify the proposed methods with large-scale camera network for very long time period.

References

1. M. D. Beynon, D. J. Van Hook, and M. Seibert abd A. Peacock, "Detecting abandoned packages in a multi-camera video surveillance system," in *IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, 2003, pp. 221–228.
2. T. Chang, S. Gong, and E. Ong, "Tracking multiple people under occlusion using multiple cameras," in *11th British Machine Vision Conference*, 2000.
3. N. T. Nguyen, S. Venkatesh, G. A. W. West, and H. H. Bui, "Hierarchical monitoring of people's behaviors in complex environments using multiple cameras," in *16 th Int. Conf. on Pattern Recognition*, 2002, pp. 13–16.
4. G. Wu, Y. Wu, L. Jiao, Y. Wang, and E. Y. Chang, "Multi-camera spatio-temporal fusion and biased sequence-data learning for security surveillance," in *11th ACM Int. Conf. on Multimedia*, 2003, pp. 528–538.

5. G. Stein, R. Romano, and L. Lee, "Monitoring activities from multiple video streams: Establishing a common coordinate frame," in *IEEE Transactions on Pattern Analysis and Machine Intelligence August 2000 (Vol. 22, No. 8)*, 2000, pp. 258–767.
6. O. Javed, K. Sha que, and M. Shah, "Appearance modeling for tracking in multiple non-overlapping cameras," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2005, vol.II, pp.26–33.
7. P. KaewTraKulPong and R. Bowden, "A real-time adaptive visual surveillance system for tracking low resolution colour targets in dynamically changing scenes," in *Image and Vision Computing, Volume 21, Number 10*, 2003, pp. 913–929.
8. V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 1999, vol.II, pp.253–259.
9. C. Stauffer and K. Tieu, "Automated multi-camera planar tracking correspondence modeling," in *Computer Vision and Pattern Recognition*, 2003, pp. 259–266.