Design of Networked Visual Monitoring Systems

Chin-Hwa Kuo, Tay-Shen Wang and Ping-Huang Wu

Department of Computer Science and Information Engineering
Tamkang University, Tamsui, Taiwan ROC
E-mail: chkuo@mail.tku.edu.tw

Abstract

We design and implement a networked visual monitoring system for surveillance. Instead of the usual periodical monitoring, the proposed system has an *auto-tracking* feature which captures the important characteristics of intruders. We integrate two schemes, namely, image segmentation and histogram comparison, to accomplish auto-tracking. The developed image segmentation scheme is able to separate moving objects from the background in real time. Next, the corresponding object centroid and boundary are computed. This information is used to guide the motion of tracking camera to track the intruders and then to take a series of shots, by following a predetermined pattern. We have also developed a multiple objects tracking scheme, based on object color histogram comparison, to overcome object occlusion and disocclusion issues. The designed system can track multiple intruders or follow any particular intruder automatically. To achieve efficient transmission and storage, the captured video is compressed in the H.263 format. Query based on time as well as events are provided. Users can access the system from web browsers to view the monitoring site or manipulate the tracking camera on the Internet. These features are of importance and value to surveillance.

Key words: surveillance system, visual object segmentation, histogram comparison

1. Introduction

We design and implement a networked visual monitoring system for surveillance in the present paper. The proposed system integrates modern image processing, computer, and networking technologies in building the advance features such as, (1) auto-tracking, (2) remote control on the Internet, and (3) random access and query of stored images. These features are of importance and value in surveillance as well as in many other applications.

Functions of traditional analog visual monitoring systems are well known. The captured images of the monitoring site are recorded at the local end or transferred to a control center. Security guidance is required in monitoring the corresponding video. Such a system usually suffers from the following problems. (1) All captured images are stored *sequentially* in tapes, which do not filter out any unimportant scenes. (2) The motion of the monitoring camera is fixed or moves periodically. Sometimes, it may fail to capture the important features of intruders. (3) Accessory sensors are needed to provide assistance, e.g., automatic alarm. (4) Systems do not provide remote control and multiple accesses in a cost-effective manner.

Research on digital visual monitoring systems has progressed actively in recent years. Change detection scheme is applied in determining the appearance of intruders [21]. As an intruder breaks in, the system is able to trigger an automatic alarm. Furthermore, content-based retrieval technique is also used in video database. The feature points tracking scheme is presented in [3] for particular person-tracking in a crowded environment. In [8], object locations are determined in 3D by multiple cameras. This information is further used in object tracking.

We make use of *image segmentation*, *histogram comparison*, and *motion tracking* in the design of our object tracking scheme. In the case of surveillance, the *object* segmentation is of concern. Discontinuity and similarity properties are two main characteristics in image edge detection. In the *boundary-based* segmentation scheme, frequency discontinuity information is used to extract the region boundaries [4,13,20]. In the *region-based* image segmentation scheme, by taking the advantage of similarity of gray level in two-dimensional domain, one can classify the regions of image [5,6,14]. The above image segmentation schemes may fail to segment an object even with the assistance of texture segmentation [13]. In the case of object segmentation, two-dimensional spatial relationship in an image and temporal relationship in a sequence of image are used. In block matching scheme, if the same motion vector persists in the neighboring blocks, these blocks are viewed as the same object [7]. Or the objects spatial relationship is first allocated by edge detection scheme, then inter-frame change detection is applied to identify the moving object in contiguous frames [9,15,21].

An object's physical characteristics, such as color, shape, and motion vectors, are used in the tracking of multiple objects in a sequence of continuous frames. In [16], object shapes are used as the main feature. It compares the object contours in the contiguous frames and finds out the best match contour to determine the corresponding object. In [22], the feature points and feature lines in every frame are determined. Next, the motion vectors of these feature points are predicted. If the predicted values and the feature points in the next frame are matched, the corresponding object is identified. In [1,11,12], histogram comparison schemes are applied for the location problem in content-based retrieval field.

We integrate visual image object segmentation, object histogram comparison and motion tracking, real-time transmission, and digital storage/query techniques in the designing of a novel networked visual monitoring system. The developed system has the ability to perform the following functions:

(1) Auto-tracking: A tracking camera tracks the intruder motion at the monitoring site. Furthermore, it takes a series of shots to capture the main characteristics of the intruders. In other words, instead of periodically scanning the monitoring site, the designed system captures the intruder's features actively. If no intruders appear, the system will not record any useless images.

(2) Web-based remote control and monitoring: Users are able to view the monitoring site from common web browsers on the Internet. Authorized users are allowed to set the tracking mode and to manipulate the tracking camera to the desired location remotely.

(3) Digital storage/query: The system provides two types of query. One is based on time. The other is based on event. Thus, the events of interest can be retrieved randomly and efficiently.

The paper is outlined as follows. In section 2, we present the architecture of the proposed visual monitoring system. Visual object segmentation and tracking schemes are given in section 3. A real-time image segmentation scheme is developed to detect scene change and to extract moving objects. Then, an *object histograms comparison* scheme is developed to detect the motion of objects. We implement a prototype of our design in section 4. Windows NT and Windows 98 are used as a platform for server end and client ends, respectively. We conclude our present work and point out future research direction in section 5.

2. System Overview

Based on the experiences learned from traditional visual monitoring systems, the following issues are of concern.

(1) real time and remote access capability,
(2) capture and store intruder's important features,
(3) automatic alarm,
(4) fast query of stored images, and
(5) remote control of the tracking camera.

Conventional analog or simple digital systems fail to satisfy the above goals in a cost-effective manner. In this paper we make use of modern digital technology in accomplishing the above requirements.

The system architecture of the present networked visual monitoring system is shown in Fig. 1. There are at least two cameras in the corresponding monitoring site. One is the global camera, which is in charge of capturing the global view of interest. The others are tracking cameras, it

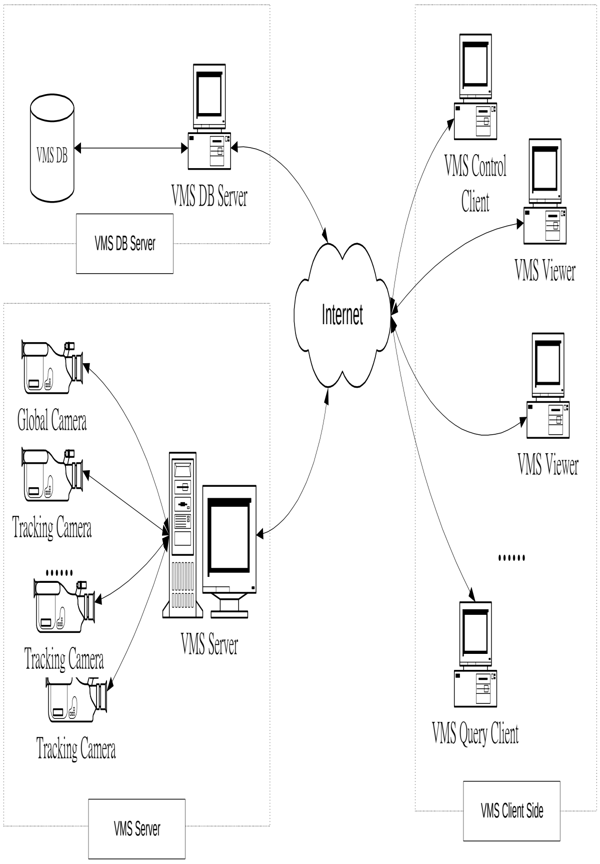


Figure 1. System architecture of networked visual monitoring system

may be one or more cameras, which is controlled by VMS server to track the intruders. In order to keep the recorded images safely, the captured images are stored in VMS server as well as VMS DB server. In the case of users requesting real time image monitoring, the system connects to VMS Server. On the other hand, the system connects to VMS DB server if users query stored images.

Server End

Data flow diagram of VMS server is shown in Fig. 2. It consists of five units: (1) camera control unit, (2) real-time image transmission unit, (3) query unit, (4) storage unit, and (5) encoding unit. There are two types of data flow in the server end. One is commands coming from the client ends. The other is inherent in the server end. A command from the client ends is first passed to the command demultiplex unit. Depending on the types of request, this command may be transferred into the camera control unit, real-time image transmission unit, or query unit for further processing. Meanwhile, in the server end, the captured images are transferred to the storage unit and real-time image transmission unit.

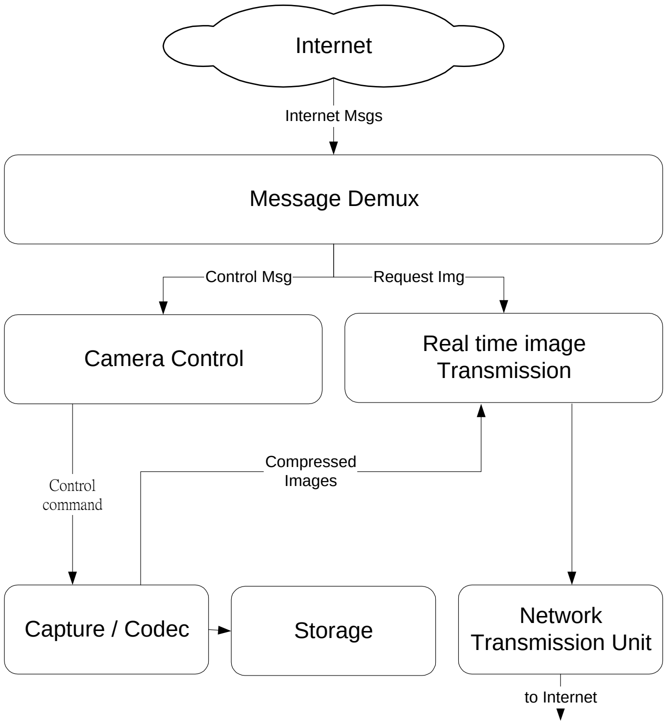


Figure 2. Data flow of VMS server

Client End

The proposed networked visual monitoring system is able to view and control the monitoring site from the client ends. A data flow diagram of a VMS client is shown in Fig. 3. Three types of request from the user end, (1) view monitoring site, (2) control camera, and (3) query of stored image. The corresponding units are as (1) decoder unit, (2) camera control analysis unit, and (3) query interface, respectively. These requests are transmitted to the Internet from the network transmission unit. As the message responds from the server, a message demultiplex unit relays these messages to the corresponding unit for further processing.

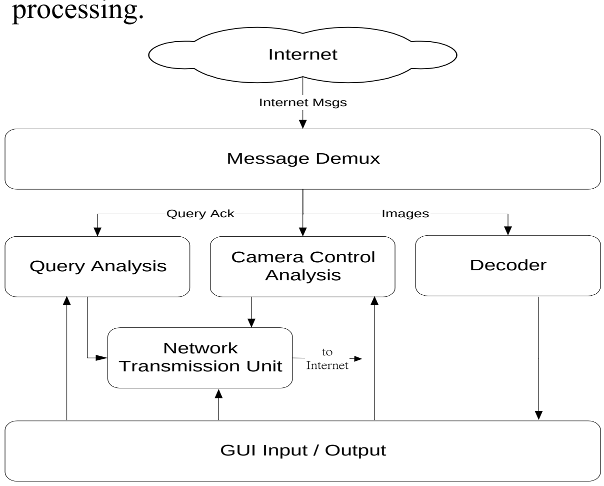


Figure 3. Data flow in the client end of VMS

Table 1 Notations

Notations	Description
F_0	The initial frame of the monitoring sequence
F_{n-1}, F_n	Consecutive two frames, F_n represents the current frame, F_{n-1} denotes the previous one
F_B	Background image of monitoring site
$P_R(x,y,n)$, $P_G(x,y,n)$, $P_B(x,y,n)$	Color intensity at the pixel (x,y) in Frame n of red (R), green (G) or blue (B) respectively. When $n=b$, that is F_B.
$P_B(x,y)$	Color intensity at the pixel (x,y) in Frame F_B ,including red, green and blue

3. Objects Segmentation and Tracking

As shown in Fig. 4, the image processing procedure of the designed visual monitoring system is divided into three phases, namely, *pre-process*, *segmentation*, and *tracking*. The captured images from the global camera are fed into a pre-process block to filter out image noise. Next, an object segmentation scheme is followed, which is in charge of segmentation of moving objects from the background. Once the motion objects are detected, the centroid and boundary of objects are computed. This information is passed to the control mechanism of the tracking camera to guide the movement of a tracking camera. The tracking camera not only follows the motion trajectory of the intruder, but also performs camera zoom-in and zoom-out to capture every detail of the intruder. Note that to accomplish the above task, processing the captured images in *real-time* is an important issue. In other words, the computation complexity of motion object segmentation and tracking schemes are of concern. In this section, the designed real-time object segmentation scheme and object tracking scheme based on histogram comparison are described.

3.1 Visual object segmentation

In the case of many surveillance applications, most of the time the background scene remains the same. We use this feature in designing an object segmentation scheme. To facilitate the presentation, notations used in the present paper are listed in Table 1. We denote F_B as the background image and F_n as the present image. As shown in Fig. 5, an object appears in the captured frames F_{n-1} and F_n. Object segmentation is achieved if we are able to find out the distinguishable region in F_{n-1} and F_n from F_B. Furthermore, object tracking is accomplished if we identify the motion of objects between F_n and F_{n-1}. A flow chart of object segmentation in the present paper is depicted in Fig. 6. It consists of *image segmentation* process and *background update* process.

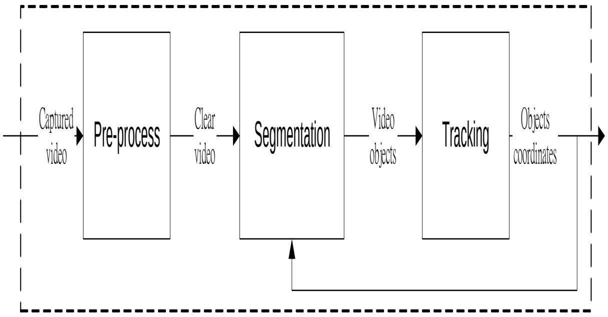


Figure.4 Image processing procedure of visual monitoring systems

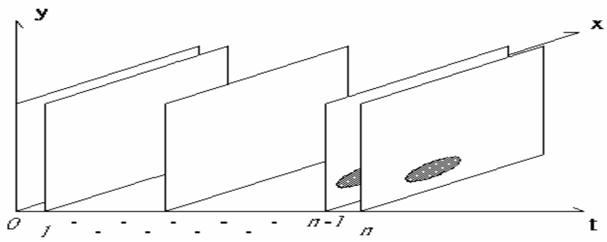


Figure.5 moving object in continuous frames

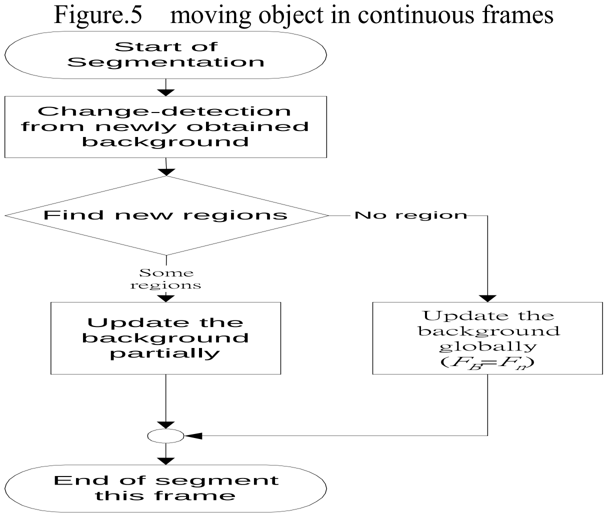


Figure 6. Visual object segmentation procedure

$$P'(x,y)=\max\left(\left|P_R(x,y,n)-P_R(x+1,y,n)\right|,\left|P_G(x,y,n)-P_G(x+1,y,n)\right|,\left|P_B(x,y,n)-P_B(x+1,y,n)\right|\right)$$

$$P''(x,y)=\begin{cases}1 & if \quad P'(x,y)\geq t\\ 0 & otherwise\end{cases} \quad (1)$$

$$S_{i,k}=\frac{\sum_{j=l_s}^{l_e}\sqrt{(P_R(i,j,n)-P(i,j,b))^2+(P_G(i,j,n)-P(i,j,b))^2+(P_B(i,j,n)-P(i,j,b))^2}}{l_e-l_s+1},k=0,1,...,m \quad (2)$$

$$S_k=\begin{cases}forground\ , & if \quad S_{i,k}\geq t\\ background\ , & if \quad S_{i,k}<t\end{cases}$$

The image segmentation process consists of the following steps.

(1) Edge detection

In the initial phase, F_0 is taken as background image F_B. The background image F_B and any present image F_n are fed into a high-pass filter, as Eq. (1), to find out the edges.

The output of the first stage of the high pass filter $P'(x,y)$ at point (x,y) is equal to the maximum color difference of R, G, and B components. Then, this value feeds into the second stage. If $P'(x,y)$ greater than a predetermined threshold t, the output of the high pass filter is equal to 1, otherwise it is equal to zero. The collection of the points equal to 1 in the corresponding two-dimensional space constitutes the edges of F_n. We refer this image as *edge image*.

(2) Image segmentation

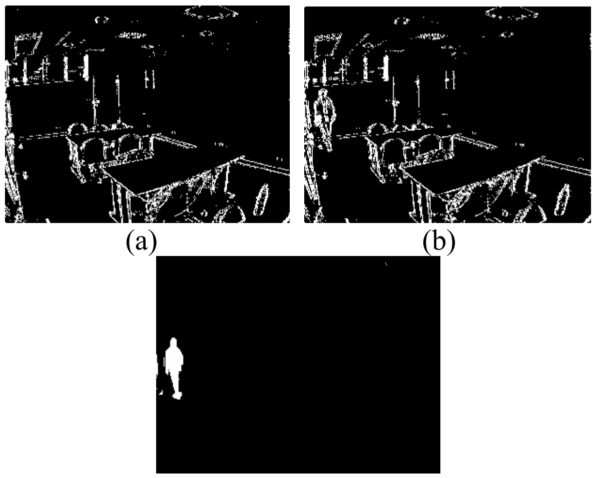
We scan this resulting new edge image from left to right and up to down. The collection of contiguous zeros or ones constitutes a *segment*. Each scan line may consist of many segments. The locations and length of each segment are recorded. Then, we compare the corresponding segment contents of F_B and F_n, as Eq. (2), to identify the noticeable segment.

where $S_{i,k}$ denotes the k-th segment of the i-th scan line, l_s and l_e represent the segment starting and ending locations of the corresponding segment respectively. Combing these segments, we extract the corresponding regions. Note that the connected segments form a *region*, which may contain more than one object. In the case of noisy conditions, scanning the image from a vertical direction can further reduce the error. After the segmented diagram of F_n is compared with F_B, the same process will be applied again for F_B to compare with F_n.

(3) Region labeling

We compute the area of each region. The region will be ignored, if its area is less than a predetermined threshold. Each region detected is tagged with unique ID. Next, the corresponding region centroid and boundary locations are computed. The object boundary locations are used to guide the motion of a tracking camera. This includes camera moving direction and zoom-in/zoom-out processes to capture the features of intruders.

An example of the above is shown in Fig. 7. Fig. 7a and Fig. 7b represent the edge images of the background image F_B and the current image F_n, respectively. The result of image segmentation of the proposed scheme is shown in Fig. 7c. Note that there are three regions. Two regions are too small, which will be eliminated as we mentioned in region labeling step.

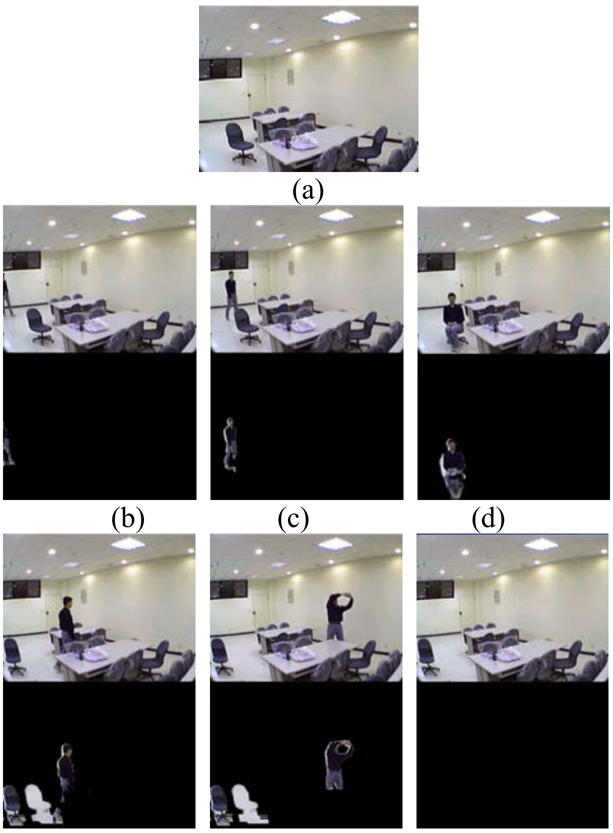


(a) (b) (c)

Figure 7. Segmentation example (a) F_B in frame No.30,(b) F_n in frame No.47, (c) Segmentation result from F_B and F_n

The background image of the monitoring site may change in some cases. For instance, an intruder moves a chair, or new objects were brought in and left in the monitoring site. Thus, we require a background update process. The scheme consists of updating partially and updating globally cases in the background update process. As shown in Fig. 6, if there is no object found in the monitoring site, the present image F_n will be used as new F_B, i.e., updating globally. This process takes care of gradual change of lighting conditions. If some objects do not move for a certain time, it will be viewed as a portion of background image. Thus, this object is pasted on F_B in the corresponding location to form a new background image, i.e., updating partially.

In Fig. 8, we demonstrate the proposed image segmentation scheme from a series of captured images. The background image is given in 8a. From Fig. 8b to 8d, an object appears in the monitoring site. The segmentation scheme successfully extracts the moving object. In Fig. 8e, a chair in the background image was moved. Thus, another two objects appear. One is the chair, another is the space it left. In Fig. 8f, the background portion inside the armpits, looking like a doughnut, can also be separated by using the above scheme. Note that if the objects do not have any movement in a predetermined time, they are viewed as background image, which is shown in Fig. 8g.

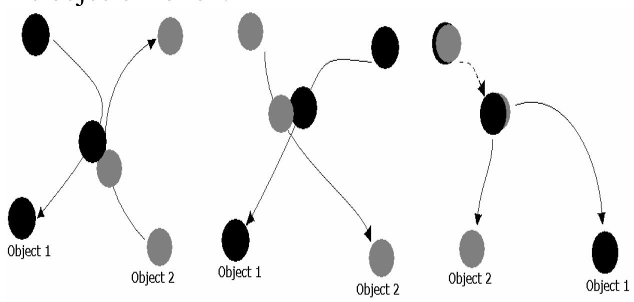


(e) (f) (g)

Figure 8. Results of image segmentation scheme

3.2 Object tracking

When the object segmentation is achieved, the object tracking is an easy issue for a single object. However, there are cases in which multiple objects may appear in the monitoring site at the same time. In order to capture the features of each intruder, we also need to handle the multiple objects situation. As mentioned, we make use of the computed region boundary to guide the movement of the tracking camera. In the case of multiple objects, two common issues are important, one is *occlusion* and the other is *disocclusion*. A simple example is depicted in Fig. 9. As shown in Fig. 9a, two objects separate at the beginning. At some instant they close together, overlap, and then separate again. Some object features may be temporarily hidden when overlap occurs. We indicate the other similar cases in Fig. 9b and 9c. In reality, there are cases which are much complex than we just showed. Therefore, a tracking scheme must overcome the above difficulties. In the present design, we make use of *histogram comparison* and *object spatial locations* to track the objects motion.



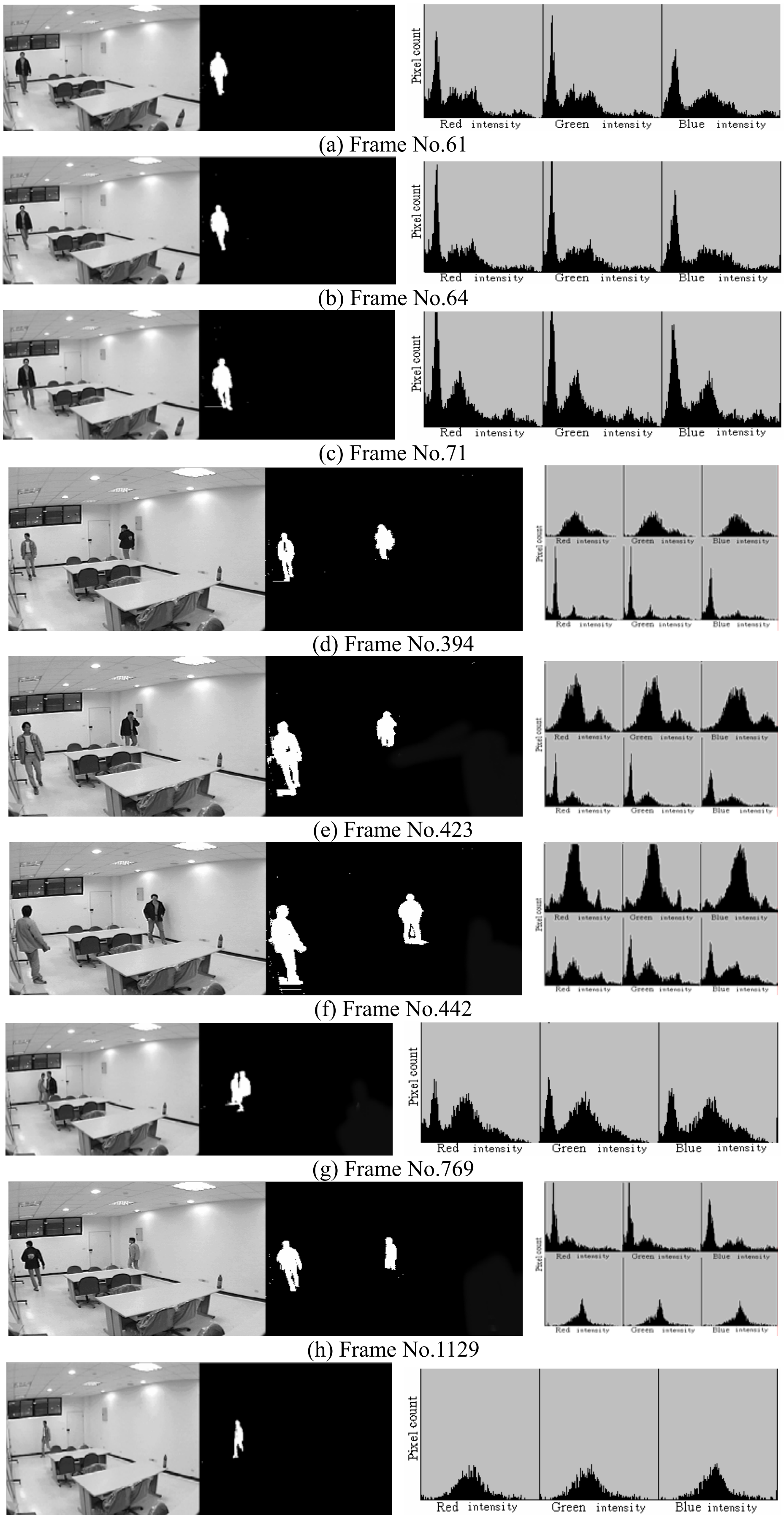
(a) (b) (c)

Figure 9. Two objects occlusion and disocclusion

Histogram comparison

Each image has its corresponding color histograms. Usually, the color histogram represents the *relative frequency* of the occurrence of color intensity of the corresponding image. A similar result applies in dealing with each individual object in the image. However, in order to differentiate the size of an object in the image, the histogram in the present work represents *exact pixel counts* of color intensity of object, i.e., x-axis denotes the color intensity of red, green, or blue and y-axis represents the pixel counts.

Based on the optical flow constraints, the color histogram profile of an object has few variants in the case of shift, rotation, and translation in a continuous image sequence. In shift and rotation cases, the color histogram remains the



(a) Frame No.61

(b) Frame No.64

(c) Frame No.71

(d) Frame No.394

(e) Frame No.423

(f) Frame No.442

(g) Frame No.769

(h) Frame No.1129

(i) Frame No.1172

Figure 10. The object histograms of some frames from test video

same. In the case of translation, only a few regions of an object would disappear or appear. Therefore, the corresponding histogram profile will slightly change in some color intensities. As the object comes close to or moves away from a camera, i.e., changes the object *area* in the whole scene, the histogram only shifts in vertical direction. The histogram profile shifts upward when an object comes close to the camera. On the other hand, it shifts downward when object moves away from camera. The similarity of the corresponding profile still exists, however. As multiple objects appear in the same scene, objects occlusion and disocclusion occur. In the beginning phase of occlusion, the merged histogram is similar to the linear addition of the original individual histograms. Based on the above, we can detect and record which objects are merged. As these merged objects split, the recorded object histograms are compared with the new resulting histograms to differentiate the objects. Therefore, object motion correspondence and moving trajectory is obtained.

As shown in Fig. 10, a sequence of continuous images, men walking into a room, is used as a test video to demonstrate the object histogram phenomena. A man appears in a room, see Fig. 10a. The segmentation of object and the corresponding histogram of red, green, and blue are also shown. The feature of the histogram can be used as tracking information is illustrated in Fig. 10b, and Fig. 10c. Note that the histogram shapes are similar in these profiles. When another man appears in Fig. 10d, two objects have different histogram profiles. We can differentiate these two objects by comparing their histogram shapes. Then, tracking of different objects can be done. Comparing Fig. 10d, Fig. 10e, and Fig. 10f also indicates the above feature. As shown in Fig. 10g, the two men crossover, i.e., two objects merge, which results in a single object. Then, they split in Fig. 10h. Comparing these figures, it is clear that the histogram has special features that can be used to differentiate different objects.

Tracking process

The designed tracking process of multiple objects case is shown in Fig. 11. It includes the following steps: (1) Coarse histogram comparison, (2) Fine histogram comparison, and (3) Merge/split histogram comparison.

(1) Coarse histogram comparison

The developed coarse histogram comparison algorithm is shown in Fig. 12. Suppose a general

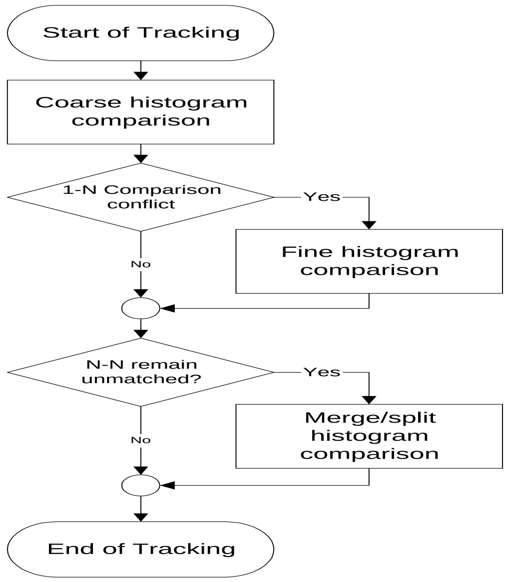


Figure 11. Objects tracking process

case, we have *m* objects in F_{n-1} and *n* regions in F_n. Coarse histogram comparison works as follows. First, the algorithm enumerates every object in F_{n-1}. The distance of each object centriod and every region centriod in F_n is computed, line#3 to line#8. Then the enumerated object is compared with the closest *n/l* neighboring regions in F_n. *l* is a predetermined integer. We select *l*=3 in our implementation. The above process makes use of spatial relationship to identify the possible regions in F_n to be the corresponding object in F_{n-1}. Next, line#10 to line#19, we perform the histogram comparison to these objects. Both mean and variance of the histogram difference between the *j*th object in F_{n-1} and the *k*th region in F_n are computed. If the computed mean or variance is less than the threshold, τ_{avg} and τ_{var}, respectively, R_i is identified to be the matched object in F_{n-1}, the algorithm adds this *k*th region into the set S_j, i.e., line#21 to line#23. This loop continues for all *n/l* neighboring regions. Therefore, the set S_j may consists of elements from zero to *n/l.* If S_j only has an element, this region will be viewed as the corresponding object in F_n from F_{n-1}. This region becomes an *object* and is marked as a *matched* object. Then, this object histogram is eliminated in the later computation process, line#24. If S_j consists of zero element, the *j*th object fails to identify the corresponding region in F_n. So, this object histogram will be further processed in the Merge/split step. If S_j consists of more than one elements, this object histogram becomes indistinguishable by only coarse histogram comparison. It will be handled in fine histogram comparison step. This process continues until the last object is compared.

Coarse Histogram Comparison Procedure.

```
for j=1 to n
   Match[j]=FLASE
for j = 1 to m
   remove all the elements in S[j]
   for i = 1 to n
      dis[i].label = i
      dis[i].distance = (O[j].x-R[i].x)^2 +
                        (O[j].y-R[i].y)^2
      sort dis array and take the first n/L
         regions.

    for k = 1 to n/l
       if Match[j] then continue
       sum = 0
       for h= 1 to 128
          sum = sum + abs( H[k][h] -
                H'[j][h] )
       Avg[k] = sum/128
       sum = 0
       for h = 1 to 128
         sum = sum + ( abs(H[k][h]-H'[j][h])
        - Avg[k] )^2
      Var[k] = sqrt(sum/127)

   for k = 1 to n/l
      if AVG[dis[k].label] <= τavg or
         VAR[dis[k].label] <= τvar
         then add dis[k].label to S[j] list
    if elements in S[j] = 1 then Match[S[j]] =
         TRUE
```

Figure.12 Coarse Histogram Comparison Procedure.

The comparison of two histograms in the above pseudo-code is as follows.

$$\mu_{i,j}=\frac{\sum_{k=1}^{128}\left|H_i(k)-H'_j(k)\right|}{128} \tag{3}$$

$$\sigma^2{}_{i,j}=\frac{\sum_{k=1}^{128}\left(\left|H_i(k)-H'_j(k)\right|-\mu_{i,j}\right)^2}{127} \tag{4}$$

Where i =1,2,…,n is the regions in F_n ; j=1,2,…,m is the objects in F_{n-1}, $H_i(k)$ is the color histogram of region i in F_n, $H'_j(k)$ is the color (depends on R_i's color) histogram of object j in F_{n-1}. And $\mu_{i,j}$ is the average of the difference between R_i and O_j, $\sigma^2_{i,j}$ is the variance of R_i and O_j's color histogram. To eliminate the camera white balance effect, the RGB color identity value is chosen to be 128 instead of 256. Not all three RGB color histograms are computed in the histogram comparison. Instead, one of these three histograms with the largest variance will be selected to represent this object's color information.

(2) Fine histogram comparison

In some cases, the above coarse histogram comparison scheme may fail to allocate the corresponding object due to existing similar histograms. A fine histogram comparison is followed to overcome the above problem. It is a multi-resolution like approach. We divide the similar pair into two parts, then compare the corresponding histograms. If the above procedure fails to identify the corresponding object in F_n, these two parts will be separated into four parts and the comparison procedure will be repeated. The process continues until the corresponding object is identified. If the area of the separated part is too small, the process is terminated. Then, the spatial relationship is used to identify the corresponding object.

(3) Merge/split comparison

After the above process, unmatched objects or regions may still exist. These include the cases of (1) new objects appearing in frame F_n, (2) objects disappearing, (3) more than one object merged into one region, and (4) one object splitting to more than one region. After the coarse and fine histogram comparison steps, it is clear that if no residual object is in F_{n-1} and there are regions existing in F_n, i.e., case (1), these regions are viewed as new objects. Oppositely, we obtain case (2). In case (3) and case (4), if we perform comparison directly, this causes a heavy computation load. For instance, we have three remaining objects in F_{n-1} and five regions in F_n. It takes $\left(C_2^3+C_3^3\right)\times\left(C_2^5+C_3^5+C_4^5+C_5^5\right)$ additions of each histogram. In resolving this computational burden, we establish a Distance Table (DT) to record the least distance between object contours and region contours, respectively. The DT is used to select the objects in F_{n-1} that may merge or split in F_n. Based on the information in DT, we add the histogram pixel count from the closest object. Then comparing the result with sequentially. If perform addition of histogram from the closest histogram. The same reasoning procedure applies in case (4).

A test video is shown Fig. 13 to illustrate the above tracking process. The corresponding frame number is indicated.

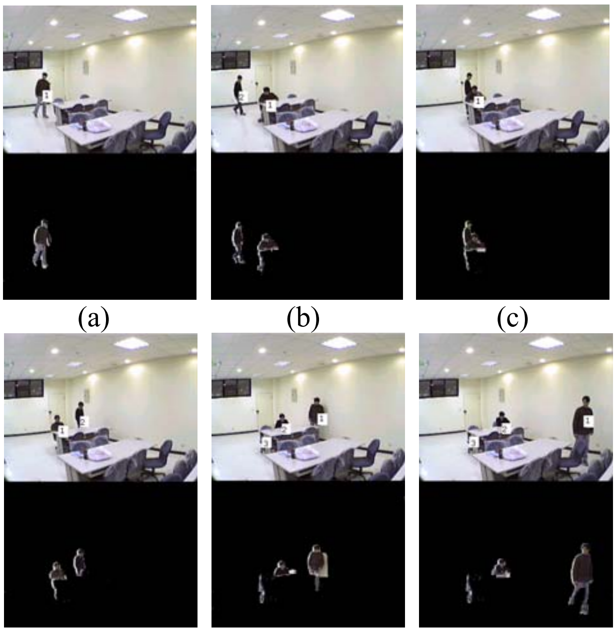


Figure 13. Test video and result of proposed tracking scheme

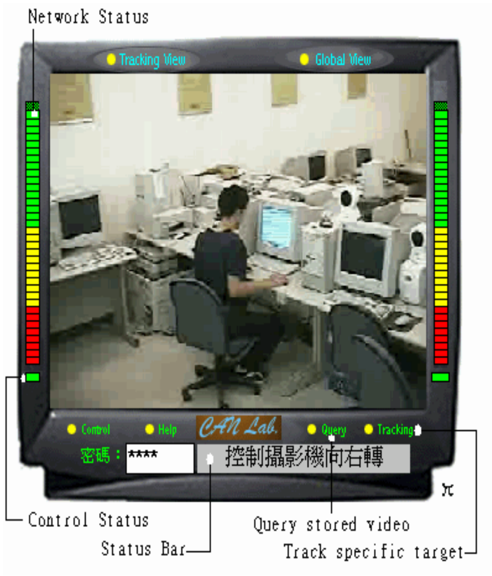


Figure 15. User interface in Client side

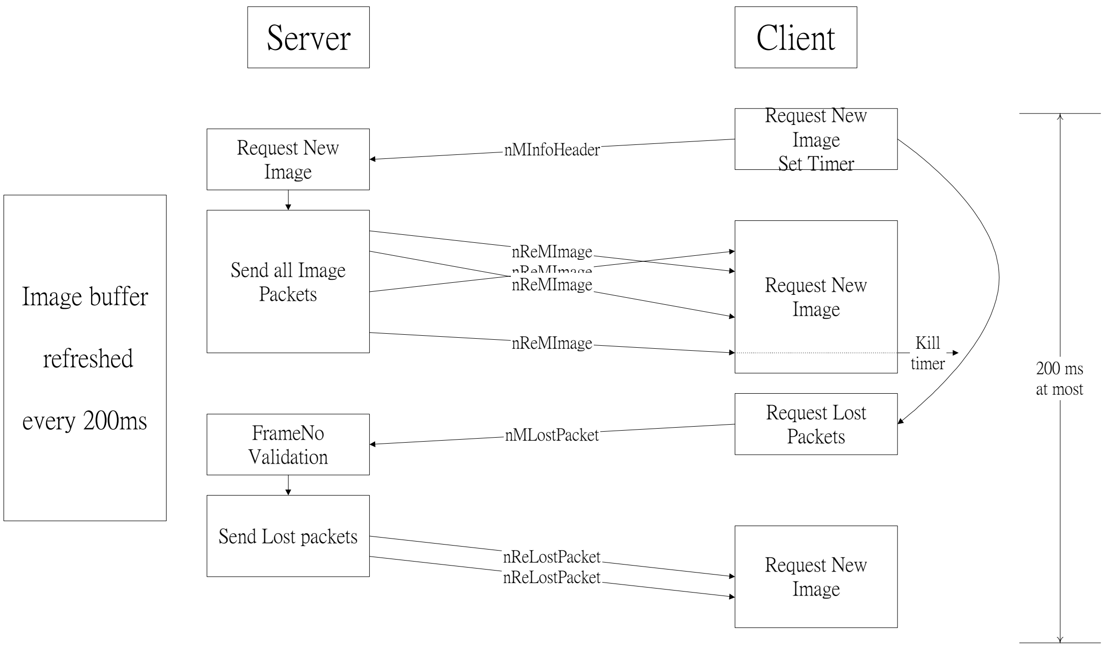


Figure 14. Network packets transmission process for real-time monitoring

4. Implementation and Results

In this section, we describe the implementation and experimental results of networked visual monitoring system.

4.1 System Implementation

MS Visual C++ 6.0 MFC is used as a system development tool. Windows NT and Windows 98 are chosen as platforms for the server end and the client ends, respectively. In the client end, we make use of embedded software technique of ActiveX to allow viewers download the designed viewing window through MS IE.

We choose Chiper CPT-8800 and Cannon VC-C1 NK II as the global monitoring camera and tracking camera, respectively. The captured images from the tracking camera will be compressed and transmitted. In this encoding portion, we integrate

MS Windows VCM (Video Compression Manager) and Intel H.263 Video codec (VIDC.I263 V.2.55.016). This is only software codec. It does not require any extra hardware. The sampling rate of tracking camera is 5 fps. The key frame rate is 6, which consists of one I-frame and five P-frames. The compressed image sizes are 4 to 6 KB for I-frame and below 1.5 KB for P-frame. Roughly speaking, network bandwidth over 160 Kbps is preferred to view the system with good quality. Meanwhile, the system equipped with Pentium-100 CPU or above is recommended in the client ends.

Users may access the VMS server in two different cases, namely, real-time monitoring and query of stored images. In the case of real-time monitoring, as shown in Fig. 16, a request message is sent to VMS server. The VMS server network transmission unit divides the current frame into several packets. These packet sizes are less than the maximum transmission unit (1,500 Bytes on the present Internet). If all packets do not arrive in time, retransmission request for those lost packets will be sent. The VMS server refreshes its buffer in 200 ms. Therefore, the retransmission process needs to be completed within 200 ms. In the case of query of stored images, the transmission process is similar.

4.2 Results

We implemented a prototype of the proposed system at Computer and Networks (CAN) in Tamkang University. Readers may access and view the results through the Internet at the web site, http://www.can.tku.edu.tw/VMS/default.html. To manipulate the camera, MS IE 4.0 browser or above are recommended. The graphical user interface is depicted in Fig. 15. There are two windows. One shows the view of tracking camera and the other is from the global camera, which shows the global view of interest. Clicking on the window will switch the view between these two windows.

Once accessing the above web page, Fig. 15 will appear. A networking condition indicator is shown on the monitor both sides to illustrate the network current traffic conditions. The authorized people may choose tracking or query function.

(1) Tracking

The system consists of two tracking modes, automatic tracking and manual control modes. In the case of automatic tracking, the tracking camera follows the target automatically. This target can be selected by VMS server or user assigns. In the case of manual control mode, the user can remotely manipulate the tracking camera to any location in the monitoring site from the web browser.

(2) query

Two query modes are provided to speedup the search process. One is based on time. The other is based on event. Query based on time is usual. The user chooses the starting time to play the record video. In query based on event, a list of events appears after the user inputs the selected time interval. As shown in Fig. 16, time and event descriptions are shown in the dialog box, and the user can further view the record video by double clicking the corresponding event.

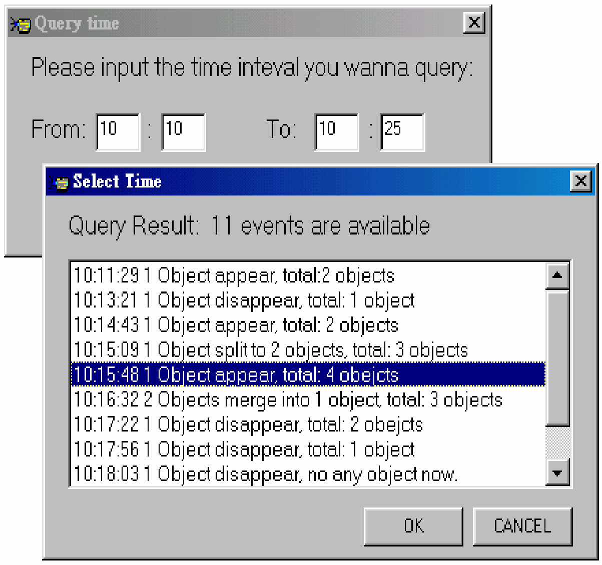


Figure 16. Query by events

5. Conclusions and Future Works

In the present paper, we make use of advanced image processing, networking, and computer technologies in developing a networked visual monitoring system. A visual object segmentation scheme is used in separating moving objects and the background scene. Then, object histogram comparison is followed in tracking moving objects. Therefore, the proposed system is able to capture further the intruders' characteristics effectively. In summary, the designed and implemented system consists of many useful features, such as (1) auto-tracking of intruders, (2) remote control and access on the Internet, and (3) digital storage and query of images. These features are useful to application such as surveillance as well as many other areas.

We will further exploited two research directions . One is multiple tracking cameras case and the other is content-based retrieval. As the environment of a monitoring site is complex, multiple cameras work cooperative is an important way in capturing the features of intruders. Thus, the cooperation of these cameras becomes a crucial issue.

Reference

[1] Boukouvalas C., Kittler J., Marik R. and Petrou M., "Color Grading of Randomly Textured Ceramic Tiles Using Color Histograms," *IEEE Transactions on Industrial Electronics*, Vol. 46, No. 1, Feb., pp. 219-226 (1999).

[2] Bremond F. and Thonnat M., "Tracking multiple nonrigid objects in video sequences," *IEEE transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 5, Sept, pp. 585-591 (1998).

[3] CADDX Controls, Inc., Motion Detectors and Accessories, "http://www.caddx.com/."

[4] Canny J., "A Ccomputional approach to edge detection," *IEEE transactions on Pattern Analysis and machine Intelligence*, Vol. PAMI-8, No. 6, Nov, pp. 679-698 (1986).

[5] Christopoulos C. A., Philips W., Skodras A. N., Cornelis J., "Segmented image coding: Techniques and experimental results," Signal Processing: Image Communication 11, pp. 63-80 (1997).

[6] Deklerck R., Corneils J. and Bister M., "Segmentation of medical images," *Image and Vision Computing*, Vol. 11, No. 8, October, pp. 486-503 (1993).

[7] Dufaux F., Moscheni F., *Segmentation besed motion estimation for second generation video coding techniques* in: Torres L., Kunt M. (Eds.), Video Coding, Kluwer Acdemic Publishers, Boston, pp. 219-263 (1996).

[8] Etoh M., "Promotion of block matching: parametric representation for motion estimation," *Proceedings of 4th ICPR*, Vol. 1, pp. 282-285 (1998).

[9] Foresti G. L., Regazzoni C. S., " A change-detection method for multiple object localization in real scenes," *Proceeding of International Conference on Industrial Electronic*, Control and Instrumentation, pp. 984-987 (1994).

[10] "http://www.komoto.com.tw/html/idx-info.html"

[11] Jing H., Kumar S. R., Mitra M. and Zhu W. J., "Spatial Color Indexing and Application, "*Proceedings of ICCV'98*, pp. 602-607 (1998).

[12] Jing H., Kumar S. R., Mitra M., Zhu W. J. and Zabih R., "Image Indexing Using Color Correlograms," *Proceedings of CVPR'97*, pp. 762-768 (1997).

[13] Ma, W. Y., Manjunath, B. S., "Edge flow: A framework of boundary detection and image segmentation," *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 744-749 (1997).

[14] Maeda J.; Novianto S., Miyashita A., Saga S., Suzuki Y., "Fuzzy region-growing segmentation of natural images using local fractaldimension," *Proceedings of Fourteenth International Conference on Pattern Recognition*, Vol. 2, pp. 991-993 (1998).

[15] Meier T. and Ngan K. N. "Automatic Segmentation of Moving Object for Video Object Plane Generation," *IEEE Transactions on Circuits and System for Video Technology*, Vol. 8, No. 5, pp. 525-538, Sep. (1998).

[16] Meier T. and Ngan K. N., "Automatic segmentation of moving objects for video object plane generation," *IEEE Transaction on Circuit and System for Video Technology*, Vol.8, No.5, pp. 525-538 (1998).

[17] Meyer, M., Hotter, M., Ohmacht, T. "New options in video surveillance applications using multiple views of a scene, "*32nd Annual International Carnahan Conference on Security Technology*, pp. 216-219 (1998).

[18] Moscheni F., Dufaux F., Kunt M., "Object tracking based on temporal and spatial information," *Proceedings of ICASSP-96*, Vol. 4, pp. 1914 -1917 (1996).

[19] Neri A., Colonnese S., Russo G., Talone P., "Automatic moving object and background separation, "*Signal Processing*, Vol. 66, No. 2, pp. 219-232 (1998).

[20] Sarkar S. and Boyer K. L., "Optimal, efficient, recursive edge detection filters," *Proceedings of 10th International Conference on Pattern Recognition*, Vol. 1, pp. 931 -936 (1990).

[21] Stringa, E., Regazzoni, C. S., "Content-based retrieval and real time detection from video sequences acquired by surveillance systems," *International Conference on Image Processing*, ICIP 98, Vol. 3, pp. 138 -142 (1998).

[22] Tang C. Y., Chen C. S., Hung Y. P., Shih S. W., and Chen Z., "Model-Based Tracking Via Verification of Hypotheses using Robust Estimation Techniques," *Proceedings of 4th RAMS' 98*, pp. 77-84 (1998).
[23] Tesei, A., Foresti G. L. and Regazzoni C. S., "Human body modeling for people localization and tracking from real image sequences," *Image Processing and its Applications*, 4-6 July, pp. 806-809 (1995).

Accepted: Dec. 30, 1999