ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

Design of Networked Visual Monitoring Systems

Chin-Hwa Kuo and *Tay-Shen Wang*

Computer and Networks (CAN) Lab.
Department of Computer Science and Information Engineering
Tamkang University, Tamsui, Taiwan ROC

Abstract

This Paper reports on the design and implementation of a networked visual monitoring system for surveillance. Instead of the usual periodic monitoring, the 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 the auto-tracking feature. The developed tracking scheme is able to segment and track moving objects from the background in real time. The tracked object information is used to guide the motion of the tracking camera to track the intruders and then to take a series of photographs. In multiple object tracking, we have developed a multiple objects tracking scheme, based on object color histogram comparison, to overcome object occlusion and disocclusion issues. To achieve efficient transmission and storage, the captured video is compressed in 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.

I. Introduction

We integrate visual 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's motions at the monitoring site. Furthermore, it takes a series of shots to capture the main characteristics of the intruders.

(2) Web-based remote control and monitoring: Users are able to view the monitoring site from common web browsers on the Internet.

(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.

Research on digital visual monitoring systems has progressed rapidly in recent years. Change detection schemes have been applied in determining the appearance of intruders [6]. As an intruder breaks in, these systems are able to trigger an automatic alarm. Furthermore, content-based retrieval techniques are also used in video databases. The feature points tracking scheme is presented in [7] for particular person-tracking in a crowded environment. In [5], object locations are determined in 3D by multiple cameras. This information is further used in object tracking.

The paper is outlined as follows. In section 2, we present the architecture of the 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, the development of an *object histograms comparison* scheme to detect the motion of objects is described. We detail the implementation of 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 directions in section 5.

II. System Overview

The system architecture of the present networked visual monitoring system is shown in Fig. 1. There are types of two cameras in the monitored site. One is the global camera, which captures the global view of interest. The other are tracking cameras, controlled by VMS server to track the intruders. In order to keep the recorded images safely, the captured images are stored in a VMS server as well as VMS DB server. In the case users request real time image monitoring, the system connects to the VMS Server. On the other hand, the system connects to the VMS DB server if users query stored images.

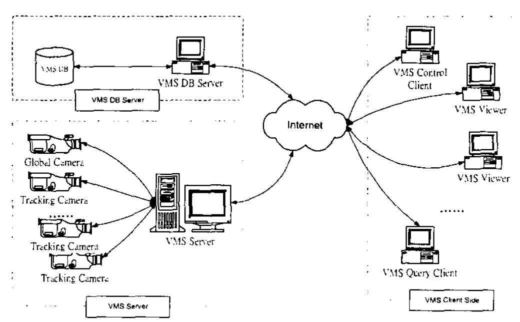

Fig. 1 System architecture of visual monitoring system

0-7803-5482-6/99/$10.00 ©2000 IEEE

Server End

The data flow diagram of the VMS server 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 first is 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.

Client End

The networked visual monitoring system is able to view and control the monitored site from the client ends. Three types of request from the user end are possible: (1) view monitoring site, (2) control camera, and (3) query of stored image. The corresponding units are: (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.

III. Object Segmentation and Tracking

As shown in Fig. 2, the image processing procedure of the designed visual monitoring system is divided into three phases, namely, *pre-processing*, *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 responsible for segmentation of moving objects from the background. Once the moving objects are detected, the centroid and boundary of objects are computed. This information is used to guide the movement of a tracking camera. 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. To facilitate the presentation, we denote F_B as the background image, F_n as the present frame and F_0 as the first frame of the surveillance video.

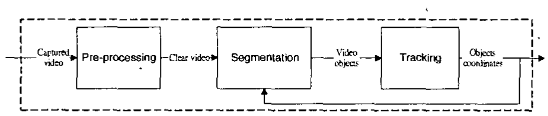

Fig. 2 Image processing procedure of VMS

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. A flow chart of object segmentation in the present paper is depicted in Fig. 3. It consists of *image segmentation* process and *background update* process.

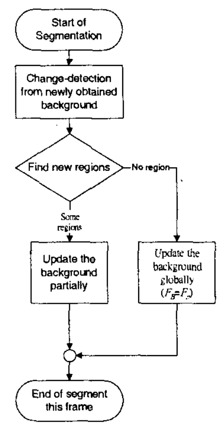

Fig. 3 Visual object segmentation procedure

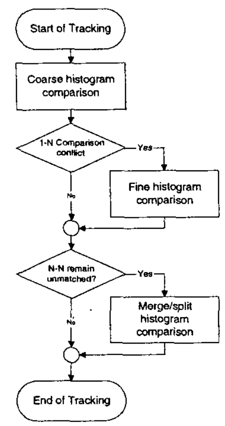

Fig. 4 Objects tracking process

(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 to determine the edges [2]. We label the edge portion as ones and zeros otherwise. Next, we scan this resulting new edge image from left to right and top to bottom. 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. This representation is called a segment diagram.

(2) Image segmentation

By performing the union operation on the segment diagram of the present image F_n and the segment diagram of the background image F_B, we are able to identify the moving objects in the present image.

(3) Region labeling

As the first step in 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.

The background image of the monitoring site may change in some cases. 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 the new F_B, i.e., updating globally. This process takes care of gradual changes in 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 partially background image.

3.2 Object tracking

There are cases in which multiple objects may appear in the monitoring site at the same time. 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. 5. As shown in Fig. 5, two objects are separate at the beginning. At some instant they meet, overlap, and then separate again. Some object features may be temporarily hidden when overlap occurs. In reality, there are cases which are much more complex than shown here. 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 motion.

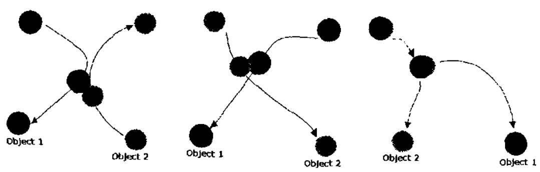

Fig. 5 Occlusion and disocclusion of two objects

Each image has its corresponding color histogram. Usually, the color histogram represents the *relative frequency* of the occurrence of color intensity of the corresponding image [1,3,4]. 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 an 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 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 moves toward 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, occlusion and disocclusion of objects 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.

The designed tracking process in case of multiple objects is shown in Fig. 4. It includes the following steps: (1) Coarse histogram comparison, (2) Fine histogram comparison, and (3) Merge/split histogram comparison.

(1) Coarse histogram comparison: In the coarse histogram comparison step, the algorithm enumerates every object in F_{n-1} and compares with every enumerated object in F_n. 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 a threshold, this region is identified to be the matched object in F_{n-1}. This process continues until the last object is compared.

(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.

(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. The process is reversed to obtain case (2). In case (3) and case (4), if we perform comparison directly, this creates a heavy computation load. Assume, for instance, that we have three remaining objects in F_{n-1} and five regions in F_n. It takes $(C_2^3 + C_3^3) \times (C_2^5 + C_3^5 + C_4^5 + C_5^5)$ 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 to reasoning the object motion. The same reasoning applies in dealing with case 4.

IV. Implementation and Results

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 to download the designed viewing window through MS IE.

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.

A test video is shown Fig. 6 to illustrate the above tracking process. The corresponding frame numbers are indicated.

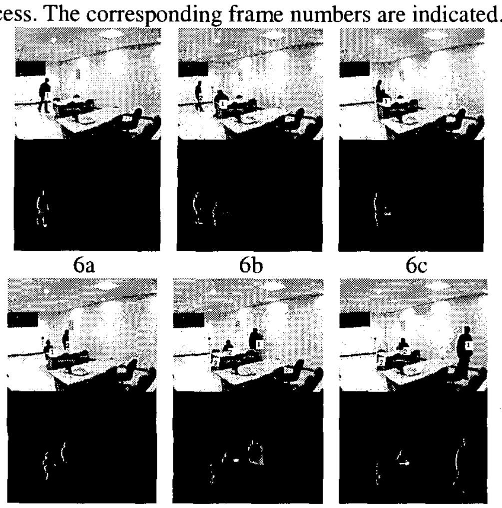

Fig.6 Test video and result of proposed tracking scheme

The graphical user interface is depicted in Fig. 7. There are two windows. One shows the view of the tracking camera and the other is from the global camera, which shows the global view of interest. Clicking on the window will switch between these two views.

Fig.7 User interface in Client side

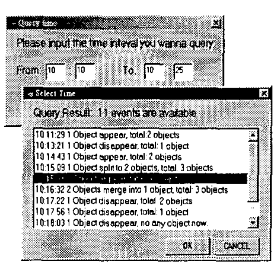

Fig.8 Query by events

Two query modes are provided to speedup the search process. One is based on time. The other is based on event. In the query based on event, a list of events appears after the user inputs the selected time interval. As shown in Fig. 8, time and event descriptions are shown in the dialog box, and the user can further view the recorded video by double clicking the corresponding event.

V. Conclusions

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 distinguishing moving objects from the background scene. Then, an object histogram comparison is used in tracking moving objects. The proposed system is thereby 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 over the Internet, and (3) digital storage and query of images. These features are useful to applications such as surveillance as well as many other areas.

References

[1] Constantinos Boukouvalas, Josef Kittler, Radek Marik and Maria Petrou, "Color Grading of Randomly Textured Ceramic Tiles Using Color Histograms", IEEE Transactions on Industrial Electronics, Vol.46, No.1, Feb. 1999, pp.219-226

[2] J. Canny, "A Ccomputional approach to edge detection", IEEE transactions on Pattern Analysis and machine Intelligence, vol.PAMI-8, no.6, Nov 1986, pp.679-698

[3] Jing-Huang, S. Ravi Kumar, Manda Mitra and Wei-Jing Zhu, "Spatial Color Indexing and Application", Proceedings of ICCV'98, pp.602-607, 1998

[4] Jing-Huang, S. Ravi Kumar, Manda Mitra, Wei-Jing Zhu and Ramin Zabih, "Image Indexing Using Color Correlograms", Proceedings of CVPR'97, pp.762-768, 1997

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

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

[7] A. Tesei, G.L. Foresti and C.S. Regazzoni, "Human body modeling for people localization and tracking from real image sequences", Image Processing and its Applications, 4-6 July 1995, pp.806-809