Journal of Computer Science 3 (2): 62-66, 2007
ISSN 1549-3636

Automated Camera Monitoring System for Selective Areas of Interest

Nael Hirzallah
Applied Science Private University, Shfa Badran, Amman 11931, Jordan

Abstract: Monitoring a number of security cameras by security guards suffers from a number of difficulties especially when the events to be watched for may take months to occur or a large number of cameras exist. Furthermore, finding and hiring people for such tasks is usually expensive. Therefore, there are immediate needs for automated surveillance systems that save money, work in a preventive way and do not miss what must be caught. In this study, we present a new algorithm for detecting motions in selected areas of interest. Although it depends on algorithms to detect scene cuts and camera breaks, but it is tailored to consider motions rather than camera breaks that usually never occurs in such surveillance video feed. The algorithm may be used in real-time by complete systems to continuously monitor surveillance video and alert security officers to suspicious events while there is still time to prevent the crime.

Key words: Surveillance monitoring, motion detection, image processing

INTRODUCTION

"Sur-Veillance" is French for "watching over" according to Merriam-Webster Dictionary. The term is often used for all forms of observation or monitoring from a distance by means of electronic equipment or other technological means like, CCTV Images (closed circuit TV); which can produce images or recordings and can be either video cameras, or stills cameras. Video cameras in turn are either analogue or digital that work on sending analogue or digital signals to a storage device such as a video tape recorder or a PC/laptop.

Traditionally, once a number of cameras are connected to displaying devices, a security guard is set to watch them and note any abnormal behavior. This process unfortunately suffers from a number of drawbacks resulting in missing what these cameras are set for. For instance, when these cameras are set to monitor a beggarly or boarder crossing events, while such events may take months to occur and such video feed are often boring to watch, the person in charge of monitoring these cameras feed may easily miss the events that he is supposed to be waiting for, resulting in some serious damages. Furthermore, mounting video cameras is usually cheap, but finding available human resources to monitor the output is expensive. Moreover, like in most current areas like banks, stores and parking lots, video data is currently used in a reactive mode, that is "after the event" as a replaying tool, thus losing its primary benefit as an active and preventive tool to such unwanted events.

Therefore, there are immediate needs for automated yet real-time surveillance systems to save money, work in preventive way and not miss what it must catch. In this study, we present a new real-time algorithm for detecting motions in selected areas, like windows and doors, or in the whole view when the camera is set, for instance, to an open boarder area. The algorithm may be used by a complete system to continuously, 24 hrs, monitor surveillance video and alert security officers to suspicious events while there is still time to prevent the crime.

Overview: Computer automated image analysis or video surveillance can automatically detect unusual events and trigger alarms, thereby reducing the volume of data presented to security personnel. However, commercially available video motion detection systems implement simplistic approaches to incident detection; thus, causing false triggers to be issued upon moving trees, passing headlights, cloud shadows and more. Operation of these systems is only feasible in highly constrained applications.

Southwest Research Institute (SwRI) has developed unique capabilities to provide heightened security and surveillance for a variety of applications[1]. The cornerstone of this capability is in real-time image processing and machine perception. Video surveillance can provide automatic detection of people and vehicles in video feeds from multiple CCTV cameras. The automated system alerts security personnel and provides data unavailable from conventional systems, such as instant replay. Facilities adopting this technology can significantly increase security because of the untiring vigilance of computer automated video surveillance.

In 1997, the Defense Advanced Research Projects Agency (DARPA) Information Systems Office began a three-year program to develop Video Surveillance and Monitoring (VSAM) technology[2].

Corresponding Author: Nael Hirzallah, Ph.D, Applied Science Private University, Shfa Badran, Amman 11931, Jordan, Tel: +962-6-5609999 x 1299 or 962-6-5519571, Fax: 962-6-5534516, E-mail: Hirzallah@asu.edu.jo

Table 1: Decision map

Short Change	Short-Period Cut	Long-Change	Long-Period Cut	Cut?	Type of Cut
0	X	X	0	0	-
X	X	0	1	0	*Action*
X	0	1	1	1	Gradual
0	1	1	1	0	*Action*
1	X	X	0	0	Flashes
1	1	1	1	1	Sharp
1	1	1	0	0	*Action*

The objective of the VSAM project was to develop automated video understanding technology for use in future urban and battlefield surveillance applications. Technology advances developed under this project enable a single human operator to monitor actives over a broad area using a distributed network of active video sensors. The sensor platforms are mainly autonomous, notifying the operator only of salient information as it occurs and engaging the operator minimally to alter platform operations.

However, unlike the existing one, this study presents an algorithm that is based on the scene detection algorithm named SmartFrame[3,4] with minor modification. SmartFrame is featured by its proven performance in detecting both gradual and sharp cuts yet filtering out false cuts. However, because our concern is not scene cuts especially that the cameras used do operate 24 hrs and the video feed is considered all as one scene, the algorithm will be customized to detect motions rather than scene cuts and filtering out false motions. If compared to the other algorithms, the proposed one is featured by its simplicity, ability to work well on moving surveillance cameras and finally its flexibility to work on either the whole frame or selected areas of interest set during camera setup.

SmartFrame: Video segmentation is a difficult process when considering various types of camera breaks and operations. A typical simple algorithm may result in detecting false cuts or missing true cuts. False cuts may result from certain camera operations, object movements or flashes within a video clip; while missed ones may result from gradual scene changes.

Video parameters include intensity, red-green-blue (RGB), hue-value-chroma (HVC), or motion vector. A basic approach to detect cuts is to compare one or more of these parameters, such as intensity, of corresponding pixels in a pair of consecutive frames. In simple words, if the number of pixels whose intensity values have changed from one frame to the next exceeds a certain threshold, a cut is detected. Although, the solution is quite simple, but it does not usually result in high detection rates.

However, SmartFrame, SF[4] was invented to achieve a higher video cut detection rate, satisfy real-time requirements and use low resources.
As seen in Fig. 1, SF carries out four comparisons:

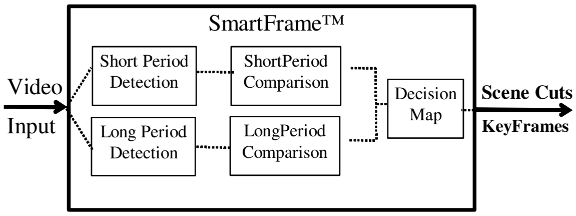

Fig. 1: SmartFrame, SF, block diagram

* Short period (say between frame 4 and 5) interframe difference with a (lower) threshold
* Long-period (say between frame 1 and 8) interframe difference with a (higher) threshold
* Comparing the short period difference with previous short periods and
* Comparing long period difference with previous long periods.

Finally, the decision map (Table 1) takes into consideration the above four results as well as the number of frames from the previous sure cut, or camera break, before deciding on a cut.

The first two comparisons allow SmartFrame to avoid detecting false cuts that may result from camera operations, objects movements or flashes. The second two comparisons allow gradual cuts to be detected and distinguished from sharp cuts.

The proposed Automated Camera Monitoring System, ACMS, algorithm takes the SmartFrame process further and focuses on detecting unwanted behaviors in the complete area of coverage by the camera or just selected areas of interest where it is believed that such events may occur.

Motion detection: Our interest in this study is to monitor the video feed from surveillance cameras in order to detect abnormal motions or intrusions and not to detect scene cuts as discussed in the previous section with the SmartFrame algorithm. This is because we know that such surveillance cameras are set to operate 24 hrs; that is, no scene cuts should exist. Besides, our interest is Motion rather than camera breaks. However, our proposal in this paper still takes advantage of such algorithms through segmenting the frame into a number of miniframes before applying the algorithm to each miniframe individually.

Figure 2 depicts the complete algorithm in question. EOM refers to "End of Miniframe" and EOF refers to "End of Frame sequence".

Segmenting the frame: Each frame of the video feed from the camera is segmented into N x M miniframes (Fig. 3). The number of miniframes chosen to segment a frame depends on the camera view and the object sizes to be monitored. If the objects to be monitored are expected to be small in size, that is, when the camera is mounted at a distance from the action; small miniframes would be a better choice.

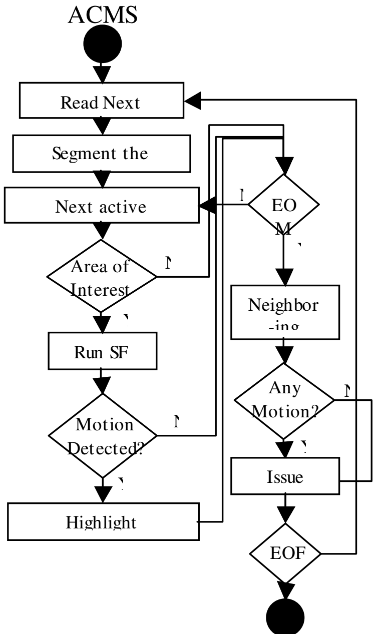

Fig. 2: ACMS algorithm flow diagram

On the other hand, if one expects that the complete camera view will be changed as a result of a motion expected and then one could leave the frame as is, that is, one miniframe. Besides, the smaller the miniframe is, the larger the processing and the more the accurate the detection is. The number of miniframe should be set manually during setup process. Figure 3 and 4 shows two views from the same camera split into 12 x 9 and 26 x 20 miniframes, respectively.

Going back to Fig 2, the SF algorithm will search for scene cuts or camera breaks in each miniframe considering them as individual frames. Although, we don't have an actually camera break, but a small miniframe may change completely within the same scene, thus, considering such event as a camera break for this particular miniframe.

Areas of Interest: In some cases where portions of the camera view may only be of interest to the security guard rather than watching the whole frame and searching for motions in areas of no interest, the algorithm offers the flexibility to select groups of miniframes and mark them as areas of interest. Only in these areas, motion needs to be monitored; hence, running the SF algorithm on these miniframes only. For instance, if one needs to monitor the windows and doors at the ground level of the building in Fig. 4, rather than detecting any motion in the whole frame including those that may happen on the front street, one can select the areas of interest as in Fig. 5 during the setup process. A miniframe within areas of interest is called Active Miniframe, while the unselected miniframe is called Inactive.

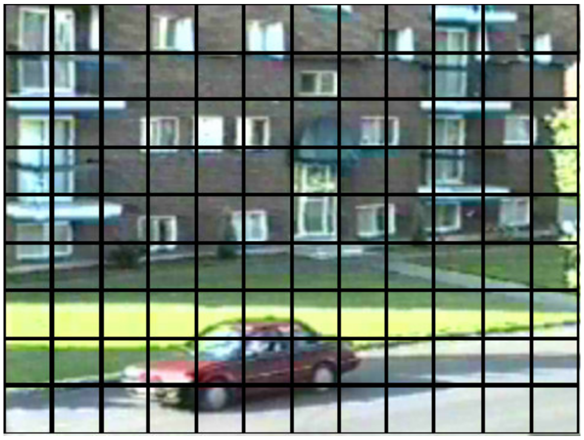

Fig. 3: Segmentation to 12 x 9 minframes

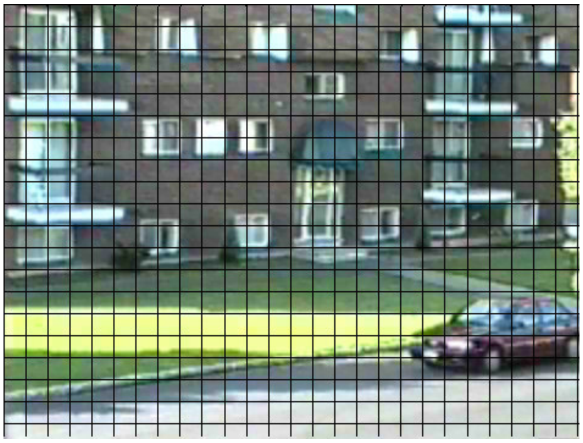

Fig. 4: Segmentation to 26 x 20 miniframes

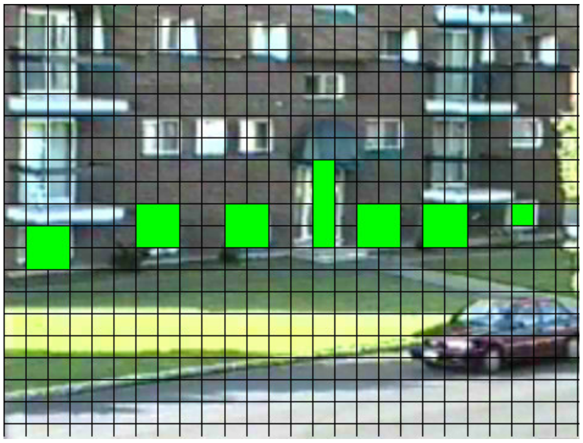

Fig. 5: Areas of Interest

By selecting the areas of interest, we not only instruct the system to alert the security guards when motion is detected in these areas, but also speed up the

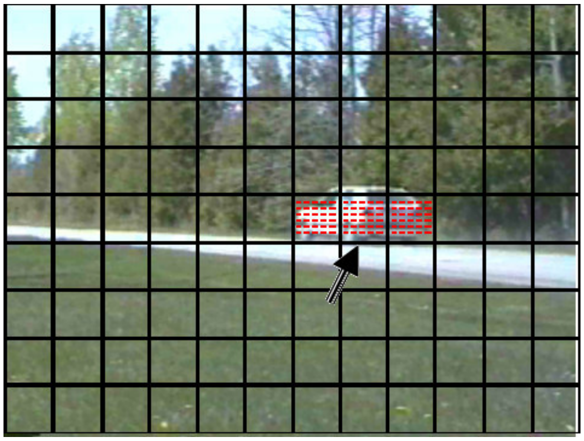

Fig. 6: Highlighting miniframes

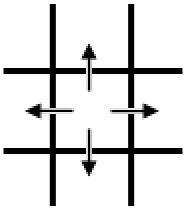

Fig. 7: Four-neighboring

motion detection process. Thus, a car could pass in front of the building in Fig. 5, while the system alerts not the guards.

Highlighting: Now after segmenting the frame into a number of miniframes and selecting the areas of interest that would result in issuing alerts, should motions be detected in them, the system passes through the entire Active miniframes within the areas of interest. Active miniframe that motions are detected in them will be highlighted for easy spotting by the guard, as shown in Fig. 6.

Neighboring check: Once SF processes all active miniframes, a Neighboring check is performed. The reason for this check is to reevaluate the miniframes that are tagged by the SF algorithm as false cuts, that is "Flashes" (Table 1), to make sure it is not a result of a fast moving real object. Consider the camera view in Fig. 6which is set to monitor a country road. If a car is moving on the road at a terribly fast speed, then the change in the miniframe (pointed at by the arrow in the figure) would be for a very short period of time, hence, detected by SF as a false cut; similar to a flash light, thunderstorm, or a bird in front of the camera. Neighboring check will check the 4-neighboring active miniframes, if any, as shown in Fig. 7. If SF as "Action" tagged any, then it would change the tag of the current miniframe from "Flashes" to "Sharp" and then highlight it.

To understand this more, a false cut means that the result of the long-Period Cut in SF is negative while Short-Change is positive; in other words, a change is occurring but will disappear momentarily. On the other hand, an "Action" means Long-Period Cut is positive, but short change is negative, in other words, a quick change that is occurring has just disappeared. This scenario illustrates that the previous active miniframe (one of the 4-neighbors) has experienced the same quick change in the past, giving an indication that an object has moved fast from the previous miniframe to the current one.

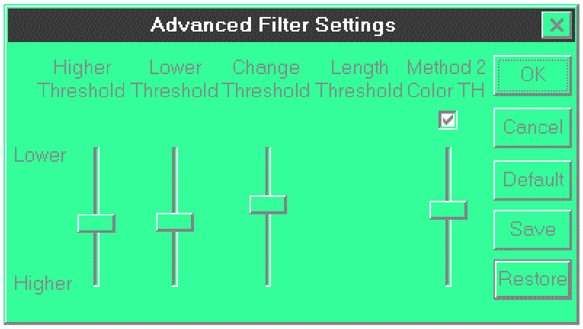

Fig. 8: Parameters settings user interface

Alerts: Once any motion within an active miniframe is detected, the system invokes the preset alerts. Although this is beyond the scope of this study, but it could be a simple module that may accept at most one alert every 15 min or so. Such alerts could be in the form of recording the video feed throughout the duration of the motion, setting an alarm sound, sending SMS messages or any type of alert the system is capable of doing, in addition to highlighting the miniframe where motion is detected in them.

RESULTS AND ANALYSIS

The system was tested on various types of cameras, including home cameras. Two different setups will be illustrated here. In the first setup, the camera was mounted away from a building and the areas of interest were selected as in Fig. 4. The results were very satisfactory; however, on some occasions when a passing car on the road was slow and its reflection falls on the building main door glass, causing a motion to be detected at the door. This may be an acceptable behavior in some occasions and not in others, depending on the setup and the guard's interests.

In the second setup, the camera was mounted monitoring the road as in Fig. 6. Regardless of the speed of the car, the car was always detected and the highlighted miniframes were following the motion.

The speed of the processing was set to a minimum of 20 frames per second for a small resolution of 180 x 240 pixels and 26 x 20 miniframes for the first setup and 12 x 9 miniframes for the second setup.

Depending on the camera used and the view monitored, whether the view is usually still like inside a building and in corridors, or it includes normal motion like moving trees due to some wind, the parameters of the algorithm could be changed through a user interface given in Fig. 8, to suite each case.

CONCLUSION

The algorithm uses a cut detection algorithm to detect motion by segmenting each frame to a number of preset miniframes. Segmentation is introduced for two folds; One to speed up processing time and two to avoid detecting motion and issuing alarms in areas of no interest to security guards. False motions, like flashes and birds in front of the cameras, are although dropped but doubled checked for possible important moving objects. If compared to the other algorithms, the proposed one is featured by its simplicity, ability to work well on moving surveillance cameras and finally its flexibility to work on either the whole frame or selected areas of interest set by the operator during camera setup.

REFERENCES

1. Magee, M.J. and M.P. Rigney, 2004. Image processing based automated monitoring of critical transportation infrastructure. Proc. ITS Safety and Security Conf., Miami Beach, Florida.
2. Collins, L., Kanade, Fujiyoshi, Duggins, Tsin, Tolliver, Enomoto and Hasegawa, 2000. A system for video surveillance and monitoring: VSAM final report. Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University.
3. Hirzalla, N. and A. Karmouch, 1995. Detecting scene boundaries for video indexing. Advanced Digital Library Forum'95, May 15-18, Washington D.C.
4. United States Patent #6,415,000. Method Of Processing A Video Stream. By Hirzalla, Nael; Streatch, Paul; Maclean, Roger; and Menard, Rob.