



Real Time Motion Detection in Surveillance Camera Using MATLAB

Furat N. Tawfeeq

Iraqi National Cancer Research Center/
Baghdad University, Iraq

Abstract— Surveillance cameras are video cameras used for the purpose of observing an area. They are often connected to a recording device or IP network, and may be watched by a security guard or law enforcement officer. In case of location have less percentage of movement (like home courtyard during night); then we need to check whole recorded video to show where and when that motion occur which are wasting in time. So this paper aims at processing the real time video captured by a Webcam to detect motion in the Scene using MATLAB 2012a, with keeping in mind that camera still recorded which means real time detection. The results show accuracy and efficiency in detecting motion.

Keywords— motion detection, real time video, surveillance camera, comparing frames, MATLAB.

I. Introduction

Surveillance cameras can be an effective technique to protect public safety and detect or deter criminal activity. Surveillance cameras are increasingly being installed inside and outside of public buildings (in elevators, hallways, entrances, etc.), on streets, home, highways, in parks and public transportation vehicles. [1] Now a day, it is seen that surveillance cameras are already prevalent in commercial establishments, with camera output being recorded to tapes that are either rewritten periodically or stored in video archives. To extract the maximum benefit from this recorded digital data, detect any moving object from the scene is needed without engaging any human eye to monitor things all the time. Real-time detection of moving regions in image sequences is a fundamental step and critical task in many vision applications. [2][3]

Motion detection is usually a software-based monitoring algorithm which will signal the surveillance camera to begin capturing the event when it detects motions. [4] Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. The video captured by the camera is being processed by the MATLAB program that helps in motion detection. [2] Frame difference used in this paper, which calculates the differences between 2 frames at every pixel position and store the absolute difference. It is used to visualize the moving objects in a sequence of frames. [5] This done according to threshold value while the video recorder still work.

II. System Methodologies

The propose system uses MATLAB as environment to detect motion in real time video as shown in flow diagram below:

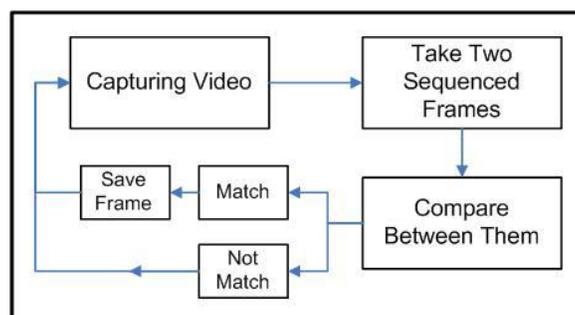


Fig. 1 System Flow Diagram

A. System Steps

The system starts by clicking button “start preview” which will start to preview video captured from webcam attached to the system, if any motion occurred; then the system will start to capture images of that motion and save it in a previously selected place. The process of detection occurs by taking two sequenced frames (for example frame1 and frame2 after converting them to gray) and compares to find the mismatching between them according to a specific threshold which will be described later. The Graphical user interface will be like figure (2) which also contains two buttons to pause and resume the video capturing.



Fig. 2 System GUI

The comparison starts between frame 1 and 2 then frame3 and 4 and so on, thus by using this method there is no reference image or background of detection which is compared sequentially with all video frames; so this proposal system can be applied in any place.

B. Noise Removal

The main idea behind the motion detection algorithm is by filtering the frames based on algorithms. Most cameras produce images with a lot of noise, so motion is detected while there is no motion. Erosion filter is used to remove random noisy pixels in order to get only the regions where the actual motion occurs. [6]

C. The Outputs

The output images that indicate the motion have names to the date and time when mismatch occurs. This option will give efficiency to the system because, by using it the user will not be forced to check the whole recorded video to find any suspicious motion instead of going to the specific date and time. Additional to this, error that indicates the motion will be added to the saved image in the upper-left corner. After completing all these processes; the system returns to the capturing video to find another mismatch between frames.

D. Error Threshold Value

By choosing the appropriate value for threshold, the detection of motion will be more accurate and neglects all suspicious motions, ones we select a value (approximately between 0.5 and 2), then a computation will take place as shown in figure (3).

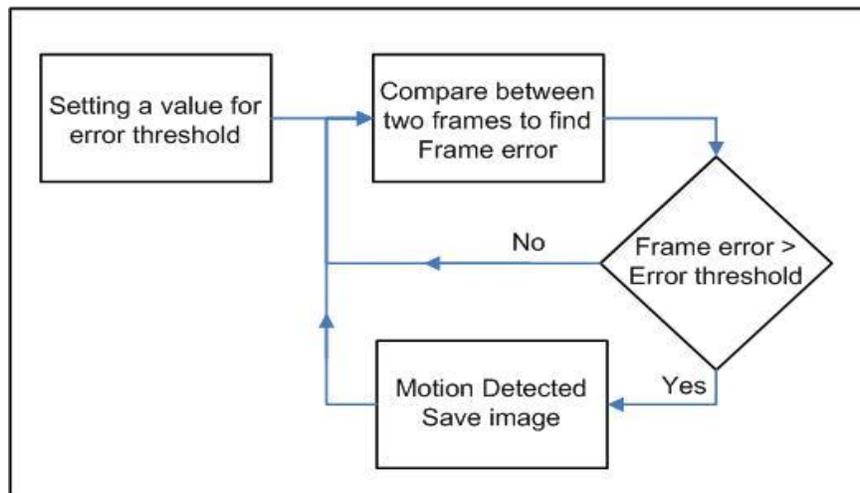


Fig. 3 Error Threshold Process

When two frames are added to the system, if the frame error (which is the ratio of mismatching) is less than error threshold, this means that there is no motion or there is but too small to detect. On the other hand if frame error is greater than error threshold, then there is a motion needs to be detected.

III. Experimental Results

The proposed system was tested in a home courtyard on person movements in two cases: afternoon and night with four different values of error frame thresholds (1, 1.3, 1.5 and 2) and by using 2MP webcam already exists in DELL laptop. The results showed during afternoon, threshold 1 detects all the movements of person in the scene with some errors (additional images with no motion), while 1.3 and 1.5 generate few number of sequenced motions without errors, threshold of 2 finds only one image from the same sequenced motion. The results are shown in figure (4).

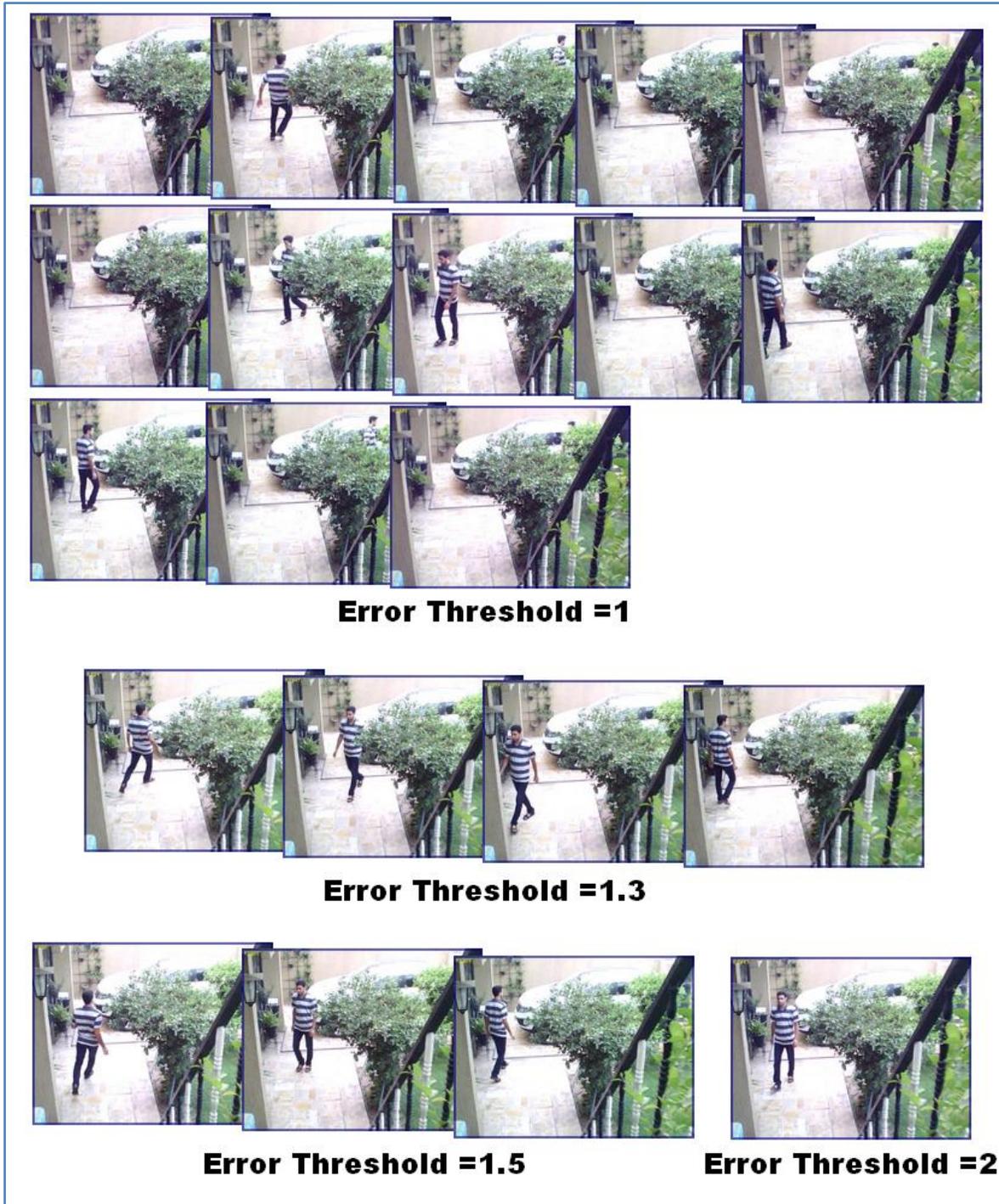


Fig. 4 Testing during afternoon with different values of threshold

In the second case where the experiment occurred during night with a good lightening condition, the same happened; but with more number of images compared with the previous case in all threshold cases due to light effects as shown in figure (5).

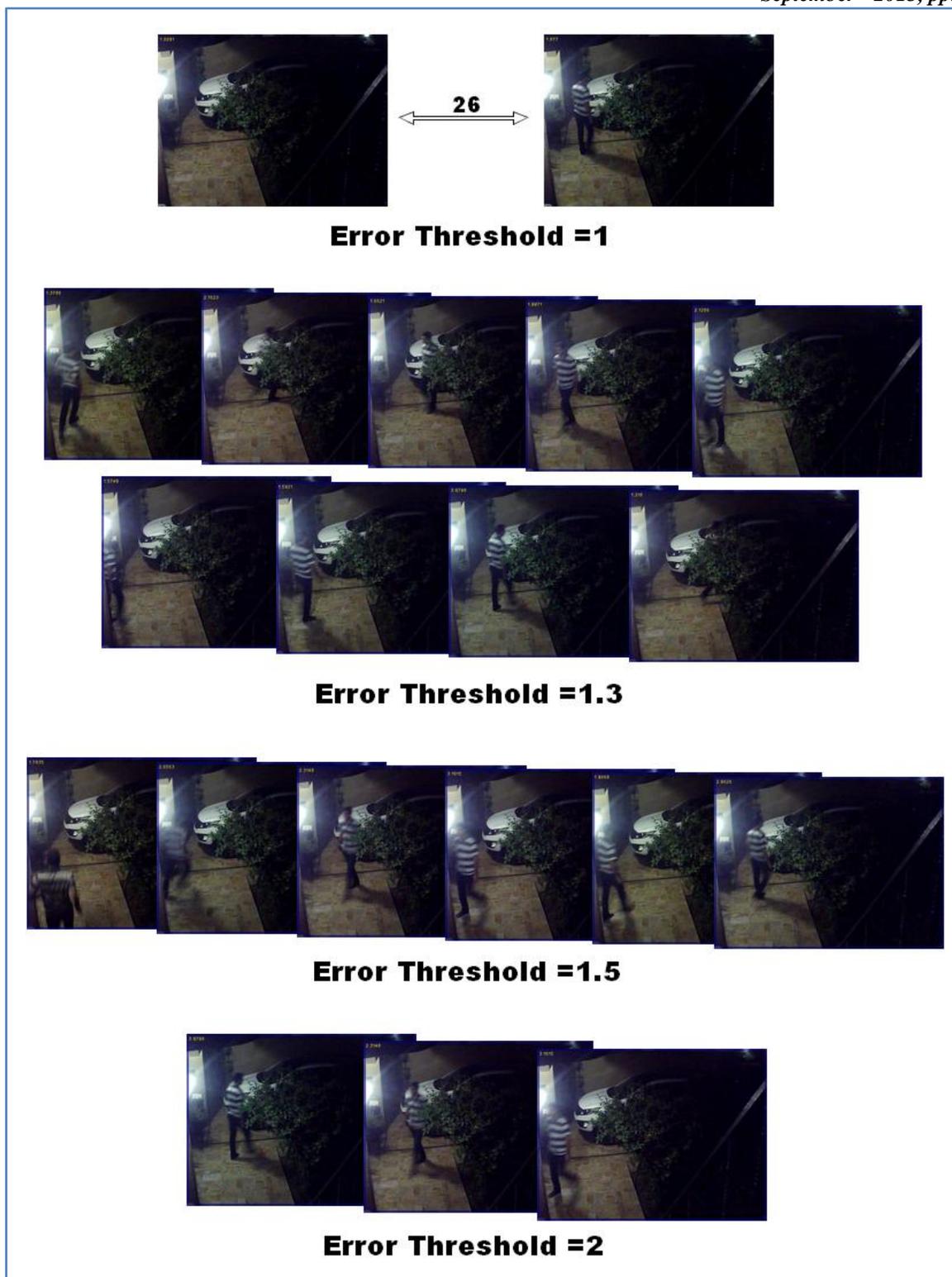


Fig. 5 Testing during night with different values of threshold

IV. Conclusions and Future Work

After trying human motion (slow and fast) with different error threshold values for the same motion; the results show that the appropriate value is between 1.3 and 1.5 especially if several motions (medium changes) occur in the scene; this will give less number of saved images that contain false or wrong detections. If the case and the threshold are not standard; we can specify a threshold according to our location that we want to monitor.

Future work can be directed toward following issues:

- Select or mention the object that moves during the scene.
- Focusing on person detection and neglecting some motions detection like wind effect on objects.
- Generating and saving a video that contains a motion and stops with the end of it.

References

- [1] Access and Privacy Service Alberta, “Guide to Using Surveillance Cameras in Public Areas”, Freedom of Information and Protection of Privacy June 2004, ISBN 0-7785-3125-2
- [2] Rao, I.S. ; Nandy, S. ; Murthy, P.H.S.T. ; Rao, V.M. ; Dept. Of ECE, GITAM Univ., Visakhapatnam, India “A real world system for detection and tracking” International Conference on Advances in Recent Technologies in Communication and Computing, 2009 „Artcom 09”.
- [3] JAIN, R. AND TERZOPOULOS, D. 1997: Color-based tracking of heads and other mobile object at video frame rates. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21-27
- [4] Gottipati. Srinivas Babu, “Moving Object Detection Using Matlab”, International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 6, August – 2012, ISSN: 2278-0181
- [5] “People tracking in surveillance applications”, 2nd IEEE International Workshop on Performance Evaluation on Tracking and Surveillance, PETS 2001, Kauai (Hawaii-USA),14/12/2001
- [6] Andrew Kirillov, Motion Detection Algorithms, Mar 2007.