Motion Recording for Surveillance Camera

Ammar Awni Abbas*

Date of acceptance 19/5 / 2009

Abstract:

In this paper we present an operational computer vision system for real-time motion detection and recording that can be used in surveillance system. The system captures a video of a scene and identifies the frames that contains motion and record them in such a way that only the frames that is important to us is recorded and a report is made in the form of a movie is made and can be displayed. All parts that are captured by the camera are recorded to compare both movies. This serves as both a proof-of-concept and a verification of other existing algorithms for motion detection. Motion frames are detected using frame differencing. The results of the experiments with the system indicate the ability to minimize some of the problems false detection and missed detections (like in a sudden change of light in the scene). The software part is written in Matlab language as an M-file and using the Simulink library, the hardware part we used a Pentium 4 computer with a web camera or a laptop integrated camera.

Key Words: motion detection, surveillance camera, computer vision, frames, real time, Matlab Simulink.

Introduction:

The ability to track moving objects, from a moving platform is a skill required in a number of fields from surveillance to robot-human interaction [1]. The purpose of this project is to first develop a simulation model using Matlab Simulink that will take a live video feed and be able to recognize and track moving objects as they move through the camera's field of view. Once this is achieved the tracking of motion frames only is made by feeding the output Audio-Video file (AVI) to an M-file program that will select the frames that contains motion and submit a report with these frames. The project uses the laptop camera or web camera to capture live data. Possible uses for this project include security surveillance for inside buildings and surrounding areas.

Frame differencing is a technique widely used for the change detection in dynamic images. It compares each incoming frame with a previous frame and classifies those pixels of significant variation into foreground. Therefore, the success of frame differencing depends on the robust extraction of the background image. The background can be modeled with a single adaptive Gaussian, and learnt during an initialization period when the scene is empty. This method is efficient only in less dynamic scenes but has difficulties with vacillating backgrounds (e.g. moving trees, background elements moving, and fast illumination changes flood of sunlight, shadows or lights switched on) [2]. The system is mainly designed to be used indoor and act as a guard that detect and record any motion that happens in an office for example when the building is empty.

It's possible to compare the current frame not with the previous one but with the first frame in the video sequence. So, if there were no objects in the initial frame, comparison of the

*Computer Center / Baghdad University

current frame with the first one will give us the whole moving object independently of its motion speed. But, the approach has a big disadvantage, if there was, for example, a car on the first frame, but then it is gone, we'll always have motion detected on the place, where the car was. Of course, we can renew the initial frame sometimes, but still it will not give us good results in the cases where we can not guarantee that the first frame will contain only static background. But, there can be an inverse situation. If we put a picture on the wall in the room, we will get motion detected until the initial frame will be renewed [3].

The most common approaches are to compare the current frame with the previous one. It's useful in video compression when we need to estimate changes and to write only the changes, not the whole frame. But it is not the best one for motion detection applications. Disadvantage of the approach are: If the object is moving smoothly we'll receive small changes from frame to frame. So, it's impossible to get the whole moving object. Things become worse, when the object is moving so slowly, when the algorithms will not give any result at all [4].

The limiting factor in deployment of these techniques is the frame-rate of currently available authenticating cameras. It is expected, however, that this limitation will be somewhat easier and faster to remove than was the development of the processing technology.

Material and Methods:

Security concerns mandate continuous monitoring of important locations using video cameras. To efficiently record, review, and archive this massive amount of data, we can either reduce the video frame size or reduce the total number of video frames we record. This paper uses the Video and Image Processing Blockset of the Matlab program to demonstrate the latter approach. The suggested model of this project is shown in figure (1).

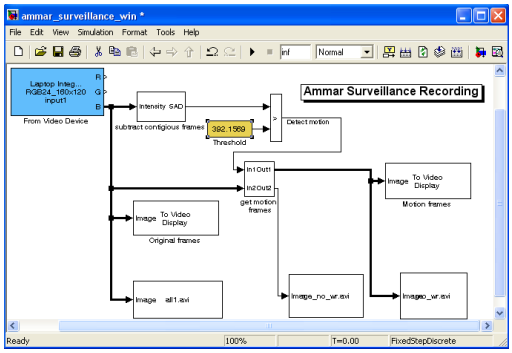


Figure (1): The suggested model

In this model, motion in the camera's field of view triggers the capture of "interesting" video frames. The frames are recorded live from any video device attached to the computer that contains the model. The model uses the two dimensions SUM OF ABSOLUTE DIFFERENCE (2-D SAD) block to detect motion in the video sequence. When the sum of absolute differences (SAD) value of a particular frame exceeds a threshold, the demo records this video frame and displays it in the Motion Frames window [5].

The first device needed for motion tracking is the camera for acquiring the real-time video stream. A video capture device will digitize the output of the camera and write it to the hard-drive to be manipulated later. A laptop computer can be used equipped with a fast enough processor and enough RAM to run the computations. The heart of the design project is the software program to analyze the incoming video stream, determine and select moving frame. The camera chosen for this project is a Laptop Integrated Webcam. The algorithm chosen to detect motion is the simplest type to speed up calculation, so that we can work on the real time. Once the new image has been balanced in the

memory the program begins to compare it with the last acquired image. For unchanged scene the result is almost zero. Any change in the scene when exceeding certain pre-defined threshold will be detected.

Figure (2) displays the detection of motion frames using the Matlab Simulink library.

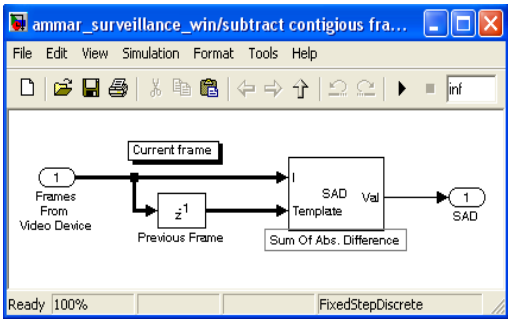


Figure (2): Blocks for detection of motion frames Subsystem.

The suggested model contains the following blocks:

1-From Video Device Block: Acquire live image data from image acquisition device.

2-Motion Energy Subsystem: Represent system within another system, this subsystem is shown in figure (2) it contains two blocks:

A- Sum of Absolute Difference (SAD): Perform 2-D sum of absolute differences.

B-Delay Block: Delay discrete-time input by specified number of samples or frames.

3-Threshold block: The DSP Constant block generates a signal whose value remains constant throughout the simulation.

4-Detect (Relational Operator block): Compares the output of motion energy subsystem with the threshold block if the result is first is greater than the second then a motion is detected and the next block is triggered.

5- Enabled Subsystem: Represent a subsystem whose execution is enabled by external input (in this case the motion).

6-Frame Counter block: Subsystem that counts all the frames that is captured by the video capture device.

7-Combine Count Block: Concatenate two input signals of same data type to create contiguous output, first signal is the motion frames, and the second is all the frames.

8-Add Frame Labels: Insert Text Block, Draw text on image or video stream, ['No of All Frames: and 'Motion Frames'].

9-Motion Frames: To Video Display Block, Send video data of the motion frames to display devices.

10-Write AVI files: Write video frames to uncompressed AVI file, there are three of these blocks:

A-All Frames: write all the frames in a file.

B-Motion Frames: Write all the motion frames in a file.

C-motion Frames No Label: Write all the motion frames in a file with no frame labels.

Today's motion detecting cameras all use variations of the same method for detecting scene changes. Conceptually, each successive picture is compared to a reference image pixel by pixel. If the number of pixels that are different exceeds some threshold, the picture is declared to be different from the reference image.

An alarm is declared when $\sum D(x, y) > d$ where d is the alarm threshold and $D(x, y)$ is the resulting image from subtracting the current frame from the previous one (pixel by pixel) [6].

Results and Discussions:

The system is placed in one of the angles of a (5M*6M) room and in a hallway so that all objects even small ones (like a copybook) have a significant size in the image and this size is larger than the threshold. Most surveillance cameras are designed to be used for human motion detection, and the human body (even babies) has

a very large size in the image and it exceeds the threshold by many times; if the system is placed in the proposed sized room or hall. The lighting in the room must be constant, this project is not designed to reduce the effects of severe false alarm (like a sudden change of light), we can reduce false alarm by increasing the threshold but this will reduce the accuracy. The threshold used in this research is obtained from other researches [5,6], and it is tested for this program and proved to be very efficient.

Testing of the system will first begin with loading artificial or acquired images to the software with a certain result expected. Testing will be as simple as a person walking in front of the camera and his or her motion viewed on the screen.

Using image-processing techniques, we will analyze a stream of images to detect motion frames.

During the testing, 40 different videos are fed to the system (27.5 hours of video containing in total of 110,918 frames) have been captured. The program labeled 40,929 of the frames (41.4%) as containing camera motion.

The Graphical User Interface (GUI) of the Program Shown in Figure (4). The interface contains three buttons:

1-Start Surveillance: When this button is pushed the suggested model will appear Figure [1], and we can start the surveillance camera action to record the motion frames by pressing the start simulation button in the standard tool bar, choose simulation from the format menu Or, simply press ctrl+T (short cut) in the keyboard, the resulted image is shown in figure (5).

2-Show Motion Frames: This button show a compact report for the motions only in the scene that have taken place for the period of the detection that the model have been working.

3-Show all Frames: displays all the frames that is captured by the video device whether it contains motion or not.

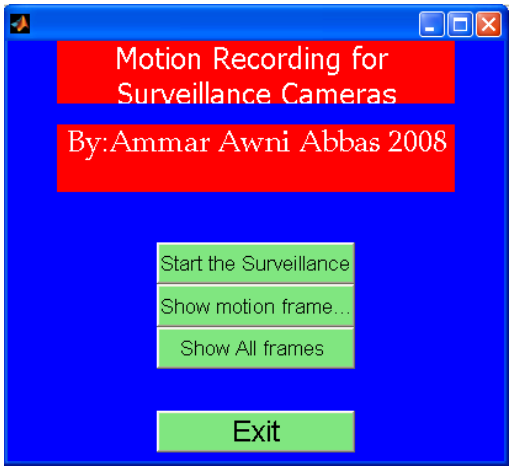


Figure (4): Graphical User Interface

The frame rate must be high enough to allow each frame to capture the tracked object's motion in small increments to allow for the motion tracking to work effectively. Also, the frame rate must be low enough to allow for the computation of each step in the motion detection process.

Too many frames will slow down the processing machine and the detection will fall behind without little chance of catching up. Ideally the camera should capture grey frames [7]. If the camera captures in colors, it will just have to be converted to grayscale at the processing level. It is estimated that a frame rate of 30 frame/second should perform well in this project.

In this project all the frames that is captured from the device are given a sequential number and all the frames that is considered motion frames are also given a different sequential number, so that we can visually compute the results and asses the work of the program during running time as shown in figure(6). When the execution of the program starts two windows appears:

The first window titled (Original): is the frames that is captured from the attached camera with no processing is

made on the frames, the frames appears gray.

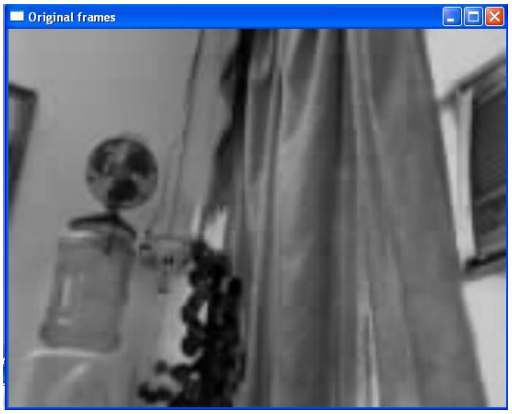


Figure (5): The captured frames from the surveillance camera.

The second window titled (Motion Frames): This window will capture the frames that contain motion in them.

Only displaying the total amount of frames captured, and the total amount of motion frames.

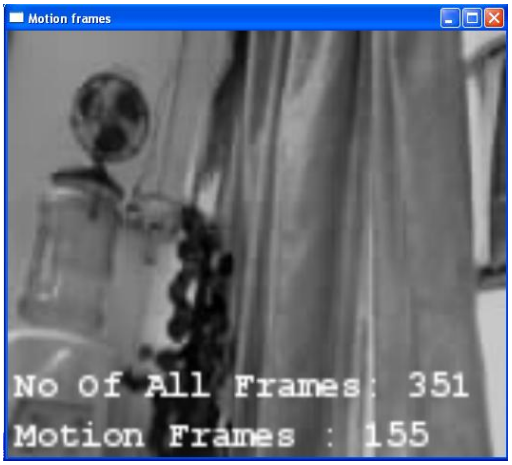


Figure (6): The program displays a window of the motion images that is being captured.

This window also appears when we start the program. It will contain frames for the motion only and it will stand still displaying the last frame that is moved if no motion take place in the current time although, the number of all frames changes, As the number of all frames is always increasing constantly by one due to continuous capturing, the number of motion frames is increased by one only when motion is detected in a frame, this will give us a visual way to evaluate the work of the program as it is being executed.

While the above two windows are made for human surveillance the M-file program is designed to make a brief report for the motions that took place for all the frames that is captured by the camera. This report is displayed with the same structure of the motion frames but with a very significant difference, that is only the motion frames are shown with no stand still images. The window will appear as if it contains one continuous movie because all the redundant frames (frames with no motion) are discarded.

Conclusions:

Significant progress has been made toward the goal of automatically detecting motion in a surveillance scenario. A significant amount of work remains to be completed to achieve this goal, and Smart Video is expected to emerge as a mature technology in the next few years. Nonetheless, the results to date already offer potential benefits in today's remote monitoring.

The demonstrated ability to determine that a scene change was not due to lighting fluctuations, but was indeed due to the movement of a discrete object in the field of view has the potential to dramatically reduce the false alarm rate. The following conclusion can be drawn from testing the suggested model:

1-frame differencing techniques are suitable for indoor surveillance not for outdoor surveillance.

2-Gray color images is very suitable for surveillance cameras due to their small size, while colored images will increase the calculation unnecessarily.

3- If the frame rate is too high the changes from scene to scene are so small that we cannot reliably extracted a frame from noise.

4- The common annotation of camera motion has not only resulted in the availability of ground truth, but has also provided insights in the properties of this video feature and the way it is perceived by humans.

5- In some cases it has been difficult to draw the line between instabilities of the camera and real camera motion. Special case of this problem is hand-held camera sequences. In such a case there can be intended camera motion, which is often difficult to separate from the instabilities that are always present.

Reference:

1.Renno, J., Lazarevic, N., D., McManus and Jones, G.A., 2006. "Evaluating Motion Detection Algorithms: Issues and Results**".** Penrhyn Road, Kingston upon Thames,Surrey,UK KT1 2EE.6(8):1.

2.Ming, X. and Ellis, T., 2001. "Illumination-Invariant Motion Detection Using Colour Mixture Models", IEEE 29(8):163.

3.Bobruk, J., Austin, D., 2003, "Laser Motion Detection and Hypothesis Tracking from a Mobile Platform", Pattern Recognition. 6(2):3.

4.Choi, H., Gregory, M., Brozyna, R., Sgherza, J., Trumper, K., 2002. **"**Motion Track / Object Recognition". Pattern Recognition. 7(2):7-14.

5. Gonzalez, R., Woods, R. Eddins, S., 2004. "Digital image processing with Matlab".Adison-Wisely, Ed.4,USA. PP:35-45 .

6.Ondrik, M., Kadner, S., Kraus, S. , Thompson, V., 2005." Motion Detection Technologies" .Pattern RecognitionLetters,2(2):2-5.

7.Laptev, I., Belongie, S.J., Perez, P., Wills, J., 2006. "Periodic Motion Detection and Segmentation via Approximate Sequence Alignment". Pattern Recognition Letters.9(5):5-7.

8.Bailer, W., Schallauer, P., Thallinger, G.,2005. "Camera Motion Detection" .Sumitmo Electronic technical Review. 2(25):5.

9.Olson, C.F., Huttenlocher, D.P., 1997."Automatic Target Recognition by Matching Oriented Edge Pixels". IEEE TRANSACTIONS ON IMAGE PROCESSING.3(4):2.

10. Bhat, K.S., Saptharishi, M., Khosla, P.K., 2001. "Motion Detection and Segmentation Using Image Mosaics". Pattern Recognition Letters. 2(2):6.

تسجيل التحرك بواسطة كاميرات المراقبة

عمار عوني عباس محمد*

*مركز الحاسبة /جامعة بغداد.

الخلاصة:

يقدم هذا البحث نظاما عملياً لرؤية الحاسبة لمراقبة الحركة وتسجيلها يعمل في الزمن الحقيقي ويمكن استخدامة في نظام مراقبة. يقوم النظام بتسجيل المقاطع الفلمية لموقع محدد ويتعرف على الصور التي تحتوي على حركة ويقوم بتسجيلها بطريقة بحيث انه يسجل فقط المناظر المهمة لنا والتي حصلت بها حركة ويقوم بعمل تقرير على شكل مقطع فلمي مختصر عن الحركة فقط يمكن عرض هذا الفلم. مع العلم ان جميع اجزاء المقطع الفلمي الذي تلتقطة الكاميرا يتم تسجيلها لكي تتم المقارنة بين المقطعين. واثبتت النتائج المستخلصة صحة النظريات الموجودة لكشف الحركة. تم تحديد المناظر التي تحتوي على حركة من خلال استخدام خوارزمية طرح المنظر الحالي من المنظر الذي سبقة. وقد اثبتت النتائج ان النظام يمكنة ان يتجاوز بعض من مشاكل المراقبة مثل التعرف الخاطئ على الحركة(وقد تم انجاز الجزء البرامجي من العمل باستخدام برنامج برنامج الماتلاب مع استخدام مكتبة السميولنك في برنامج الماتلاب, اما المكونات المادية للعمل فتم استخدام حاسبة بنتيوم 4 تتصل بها كاميرا ويب او استخدام كاميرا مظمنة في الحاسبة لابتوب).