Thi_Qar University Journal for Engineering Sciences, Vol. 3, No. 1 2012

Real-time Alert of Motion Detection Using Microcontroller

Haithm Abdul Raheem Taha

Al-Mostanserya University

Abstract

This paper addresses the problem of motion detection and moving objects of variable appearance in challenging scenes rich with features and texture. It is made particularly difficult by the nature of objects encountered in such scenes. The research included a method which uses fast motion detection and segmentation as constraint of differences between frames of video, and when changes in each pixel of an image, will alert start and message displayed on LCD that plugged to microcontroller.

Keywords: Motion Detection; Microcontroller; LCD; Alarm; Interface.

المستخلص

الغرض من البحث هو التنبيه إثناء عملية الكشف عن الحركة وإثناء تحريك الأجسام الموجودة في المشهد الذي يحتوي على الكثير من التفاصيل والأجسام الأخرى. والعمل المقترح هو باستخدام طريقة سريعة للكشف عن الحركة وذلك بالفرق بين الصور المصورة عبر التصوير الفيديوي، وعندما يكون التغيير في اي عنصر من عناصر الصورة (البكسل)، يتم التنبيه بإرسال رسالة إلى دائرة تحتوي على معالج دقيق وشاشة عرض صغيرة لعرض الرسالة التنبيهيه وتشغيل منبه صوتي إثناء عملية الكشف عن الحركة.

1. Introduction

A video sequence is a much richer source of visual information than a still image. This is primarily due to the capture of motion; while single image provides a snapshot of a scene, a sequence of images registers the dynamics in it. The registered motion is very strong cue for human vision; that can easily recognize objects as soon as they move even if they are inconspicuous when still. In this paper, an application designed require image points be identified as moving or not, or that it be measured how they move. The first task is referred to as motion detection, whereas the latter is referred to alert using microcontroller which considered so important task in security especially in government offices and special sectors that need kind of security systems found in place and do work which considered difficult for human to do, and here come the benefit of automation devices self-controlled[1].

2. Study of proposed work

The objective of this work was to develop a surveillance system which would detect motion in a live video feed and if motion is detected, then to activate a warning system and display warning message on LCD.

The activation of an alarm would help in nullifying a threat of security. The proposed system is shown in the Figure (1).

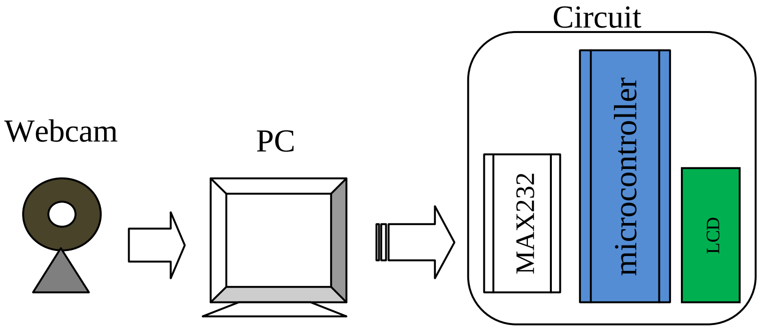


Figure (1). System proposed of motion detection.

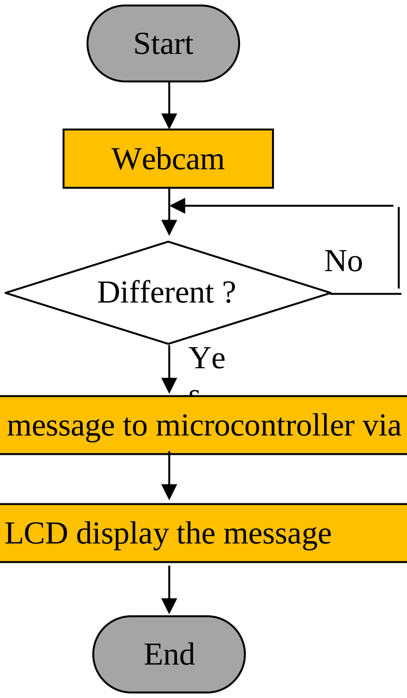


Figure (2). Flow chart of proposed work.

3. Motion detection

Motion detection is, arguably, the simplest of the motion related tasks, i.e., detection, estimation, and segmentation. Its goal, is to identify which image points, or more generally which regions, of the image have moved between two time instants. As such, motion detection applies only to images acquired with a static camera. However, if camera motion can be counteracted, by global motion estimation and compensation, then the method equally applies to images acquired with a moving camera[2].

It is essential to realize that the motion of image points is not perceived directly but rather through intensity changes. However, such intensity changes over time may be also induced by camera noise or illumination changes. Moreover, object motion itself may induce small intensity variation or even none at all[2].

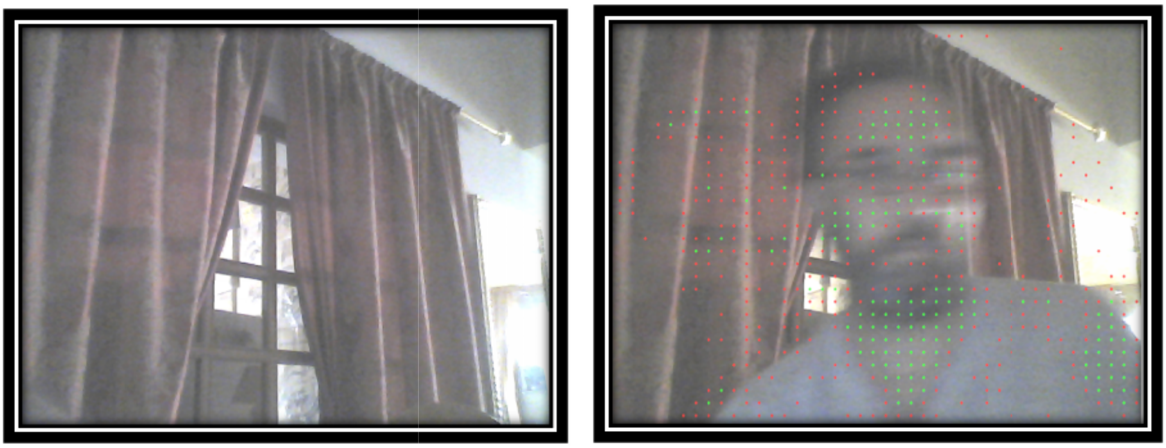
4. Image differencing for change detection

Often it is of interest to detect changes that occur in images taken of the same scene but at different times. If the time instants are closely placed, e.g., adjacent frames in a video sequence, then the goal of change detection amounts to image motion detection. There are many applications of motion detection and analysis. Detected motion is also useful for tracking targets and for recognizing objects by their motion[3].

If the time separation between frames is not small, then change detection involves the discovery of gross scene changes. This can be used for security or surveillance cameras, or in automated visual inspection systems. Suppose that f_1 and f_2 are images to be compared.

$$g = |f_1 - f_2| \tag{1}$$

The absolute difference image will embody those changes or differences that have occurred between the images. At coordinates *n* where there has been little change, g(n) will be small. Where change has occurred, g(n) can be quite large. Figure (3) depicts image differencing. In the difference image, large changes are displayed as points appear over the place of change in the image. Since significant change has occurred, there are many pointed intensity. This difference image could be processed by an automatic change detection algorithm. A Simple series of steps that might be taken would be to binaries the difference image, thus separating change from nonchange, counting the number of high change pixels, and finally, deciding whether the change is significant enough to take some action. Sophisticated variations of this theme are currently in practical use[3].



(a) Original placing scene; **(b) motion detect with pointing non-matched area.**

Figure (3). Image differencing example.

Both continuous and discrete representations of motion and images will be used, with bold characters denoting vectors. May be $x=(x_1,y_1)^t$ be a spatial position of a pixel in continuous coordinates may be expression as x □ $R2$ within image limits, and let I_t denotes image intensity at time t. Then, $I_t(x)$ □ R is limited by dynamic range of the sensing device (webcam in our study). Before images can be manipulated digitally, they have to be sampled and quantized. Let $n=(n_1,n_2)^t$ be a discretized spatial position in the image that corresponds to x. Similarly, let k be a discretized position in time, also denoted t_k. It is assumed here that images are either continuous or discrete simultaneously in position in amplitude. Consequently, the same symbol I will be used for continuous and quantized intensities; the nature of I can be inferred from its argument (continuous valued for x and quantized for (n)[4].

Motion in continuous images can be described by a velocity vector $v=(v_1,v_2)T$. Whereas $v(x)$ is a velocity at spatial position x,v_t will denote a velocity field or motion field, i.e., the set of all velocity vectors within the image, at time t. Often the computation of this dense representation is replaced by the computation of a small number of motion parameters b_t with the benefit of reduced computational complexity. The, v_t is approximated by b_t by means of a known transformation. For discrete images, the notion of velocity is replaced by displacement (d) [4].

5. Hypothesis testing with adaptive threshold

The motion detection methods presented thus far were based solely on image intensities and made no a priori assumptions about the nature of moving areas. however, moving object usually creates compact, closed boundaries in the image plane, i.e., if an image point is declared moving[5], it is likely that its neighbor is moving as well (unless the point is on a boundary) and the boundary is smooth rather than rough. To take advantage of this a priori information, hypothesis testing can be combined with Markov random field models.

Let E_k be a Markov Random Field (MRF) of all labels assigned at time t_k, and let e_k be its realization. Let us assume for the time being that $e_k(n)$ is known for all n except I. Since the estimation process is iterative, this assumption is not unreasonable; previous estimates are known at $n \neq I$. Thus, the estimation process is reduced to a decision between $e_k(I)=M$ and $e_k(I)=S$. Let the label field resulting from $e_k(I)=M$ be denoted by e_k^M and that produced by $e_k(I)=S$ be e_k^S. Then, the decision rule for $e_k(I)$ can be written as follows[3]:

$$\frac{P(\rho_k|e_k^M)}{P(\rho_k|e_k^S)} \underset{S}{\overset{M}{\square}} \theta \frac{P(E_k = e_k^S)}{P(E_k = e_k^M)} \tag{2}$$

where P is a probability distribution governing the MRF E_k. By making the simplifying assumption that the temporal differences ρ_k (n) are conditionally independent, i.e., $P(\rho_k|e_k) = \prod_n P(\rho_k(n)|e_k(n))$, then further rewrite eq.(2) as[3]

$$\frac{P(\rho_k(I)H_M)}{P(\rho_k(I)H_S)} \underset{S}{\overset{M}{\square}} \theta \frac{P(E_k = e_k^S)}{P(E_k = e_k^M)} \tag{3}$$

The hypothesis H_M means that $e_k(I)=M$ and H_S means that $e_k(I)=S$. All constituent probability densities from the left-hand side of Eq.(3), except at I, cancel out since $e_k M$ and $e_k S$ differ reasonable in stationary areas (temporal differences are mostly due to camera noise), it is less so in the moving areas. However, a convincing argument based on experimental results can be made in favor of such independence.

To increase the detection robustness to noise, the temporal differences should be pooled together, for example within a spatial window W_1 centered at I. This leads to the evaluation of the likelihood for all ρ_k within W_1 given the hypothesis H_M or H_S at I. Under the assumption of zero-mean Gaussian density P with variances σ_M^2 and σ_S^2 for H_M and H_S, respectively, and assuming that $\sigma_M^2 \ \square \ \sigma_S^2$ the final hypothesis becomes[3]:

$$\sum_{n \square W_1} \rho_k^2(n) \underset{S}{\overset{M}{\square}} 2\sigma_S^2 \left(-ln\theta + N \ln \frac{\sigma_M}{\sigma_S} + \ln \frac{P\,(E_k = e_k^S)}{P\,(E_k = e_k^M)} \right) \tag{4}$$

where N is the number of pixels in W_1. In case the a priori probabilities are identical or fixed (independent of the realization e_k), the overall threshold depends only on model variances. Then, for increasing σ_S^2, the overall threshold rises as well, thus discouraging M labels (as camera noise increases only large temporal differences should induce moving labels). Conversely, for decreasing σ_S^2, the threshold falls, thus biasing the decision toward moving labels. In the limit, as $\sigma_S^2 \to 0$ the threshold becomes 0; for a noiseless camera even the slightest temporal difference will induce a moving label[1].

By suitably defining the a priori probabilities one can adapt the threshold in response to the properties of e_k. Since the required properties are object compactness and smoothness of its boundaries, a simple MRF model supported on a first-order neighborhood with two-element cliques C{n,*I*} and the following potential function[3]

$$V_{nl} = \begin{cases} 0 & if \;\; e_k(n) = e_k(I) \\ \beta & if \;\; e_k(n) \neq e_k(I) \end{cases} \tag{5}$$

The eq.(5) is appropriate. Whenever a neighbor of n has a different label than $e_k(n)$, a penalty $\beta > 0$ is incurred; summed over the whole field it is proportional to the length of the moving mask boundary.

$$p(E_k = e_k) = \frac{1}{Z}\exp\left(-\frac{1}{T}\sum_{\{n,I\}} V_{nl}\right) \tag{6}$$

Thus, the resulting probability will increase for configurations with smooth boundaries and will reduce for those with rough boundaries. More advanced models, for example, based on second-order neighborhood systems with diagonal cliques, can be used similarly[3].

6. Interfacing and microcontroller

Figure (4) illustrates , the schematic of the controller that receives a message from software of motion detection. This design uses a microcontroller 8051 family (AT89S52) with LCD (16*2) to display the message that is sent via RS232 port to microcontroller[6].

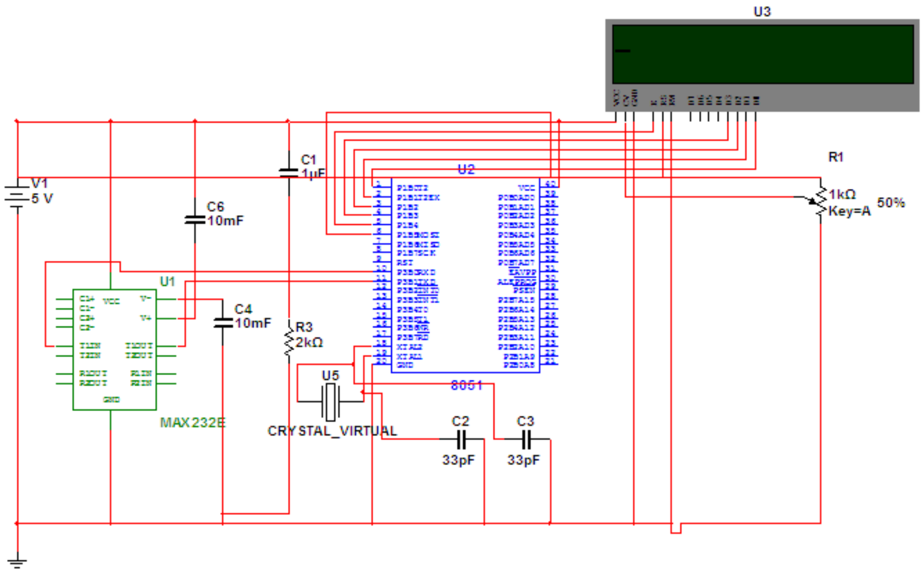


Figure (4). Schematic of the proposed work.

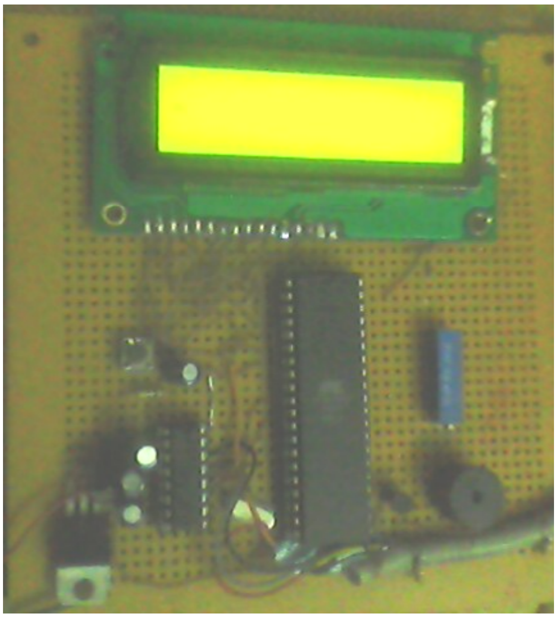


Figure (5). Practiced design circuit of proposed work.

7. Results

A webcam starts after specify the port number of the RS232, which allowed later control via this port. No actions will take while there is no motion happened in front of the webcam that can pass the differences part and send a message. When motion detected then a statement used to send a message to microcontroller via Rs232 port, is descript below:

Mscomm1.Output* = *"Motion Detected"

As shown in Figure (6), shows when a microcontroller start or when restart.

Figure (6). Starting/Restarting a microcontroller.

Below the snapshots take when an motion detected. As noted in Figure (7) the points of green color drawn around the object or any differences happening in an image.

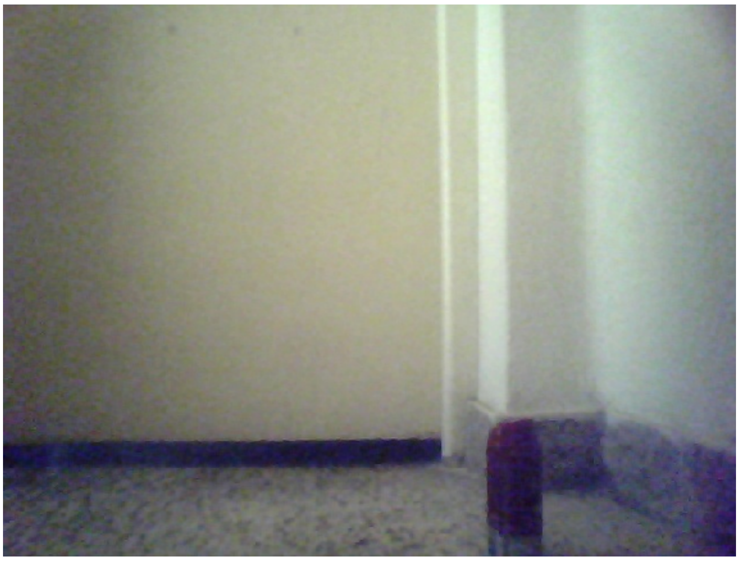


Figure (7). Snapshot of object when no motion detected.

Figure (8). Snapshot of object when motion detected.

Figure (9). Snapshot of object when no motion detected.

Figure (10). Snapshot of object when motion detected.

And when a motion/movement detected, microcontroller start display that a movement detect, and the alarm will start after 10 seconds as shown in Figure (11).

movement detect
10 seconds start

Figure (11). Movement detection display.

After detection happened and microcontroller start a notification for the movement found, an alarm will started, as shown in Figure (12).

Alarm Started
duration 5 sec_

Figure (12). Alarm started display.

8. Conclusion

In this paper, camera motion detection methods using a differences of images generated by the video (webcam) are described. The proposed method is evaluated by using a circuit implemented a microcontroller and LCD (16*2) to enhance the security of sensitive places and sectors need to watch automatically from any kind of movement of objects or get through inside the scene that observed.

The advantage of proposed system is good solution for victim detection motion detection and good visibility during daytime and relatively low initial cost, low power, built-in electronic package, small size, and ease of interface with its digital output. And the disadvantage can fail if the sensor mat is not placed on a hard board.

The proposed system has present an algorithm based on MRF modeling for motion detection in image sequences, spatial and temporal interactions between pixels are modeled by MRF. This system communicates with a modem or PC to electronic design circuit sending and receiving data, and then direct the program to understand its operation. And the operation here is based on image processing where image been changed or any part of the image, and then calculation between two images (the original and last image). PC's serial data acquisition interfaces require the sending and receiving of data to operate. To communicate with the serial port using Visual Basic, the MsComm control must be utilized to allow serial data transfer via a serial port (Com1-Com4).

When detection of motion happens, the signal send to microcontroller which interfaced to same system has been detect a motion. A microcontroller received this data via MAX232 which convert the data of serial to TTL/CMOS, when receive a data, the microcontroller will find out what kind of data, is there a detection or not, if there is are a motion detection happened, then will send a message to serial LCD to be displayed, and in same time will set the alert on for 5 seconds.

9. Reference

[1] Al Bovik; 2002; "Handbook of Image and Video processing"; Elsevier Academic Press.

[2] Anthony C. Caputo; 2010; "Digital Video Surveillance and Security"; Butterworth-Heinemann Press.

[3] Janusz Konard; 2002; "Motion Detection and Estimation"; Elsevier Academic Press.

[4] Masaki Naito, Kazunori Matsumoto; 2006; "Camera Motion Detection Using Video Mosaicing"; Multimedia and Expo, IEEE International Conference.

[5] Jonathan Glumac; 2009;"Motion Detection and Object Tracking with Interactive Surfaces"; Redfish Group, Worcester Polytechnic Institute (WPI).

[6] Chris Braithwaite, Fred Cowan, Hassan Parchizadeh; 2004; "8051 Microcontroller, An Applications Based Introduction"; Elsevier Academic Press.