Motion Detection in Videos of Visual surveillance Using Robust Temporal Averaging Method

Sivanagireddy kalli, Bhanu Murthy Bhaskara

Abstract— The video surveillance system are used for monitoring security in sensitive areas like banks, shopping complex, traffic monitoring on highway, public places which are crowded. For the making of video surveillance systems smart, it requires fast and reliable algorithm for detection of the person or object motion. Most of the motion detection algorithms focus on decreasing the rate of false alarms for obtaining reliable and fast. In this paper we presented an efficient motion detection algorithm based on background subtraction using robust temporal Averaging Method by updating speed of the background model.

Index Terms— TAM, Motion Detection, video surveillance, Background, Kernel Density.

1 INTRODUCTION

The moving object detection is an important task in the image sequences of the area which is under surveillance system. Detection of moving object is necessary for surveillance application, for guidance of autonomous vehicles, person movements in sensitive areas like banks and shopping-malls. The initial step of object recognition is the moving object detection and it is an extracting moving object from video sequences with back-ground which can be static or dynamic. In the proposed system the CMOS camera is used for vehicles to detect their motion. The digital camera has become a popular sensor for monitoring and surveillance systems. By improvement of the semiconductor technology, modern digital cameras with high pixel number can provide greater image detail in various applications. A digital image sensor usually utilizes an array consists of the charged-couple device (CCD) or complementary metal oxide semiconductor (CMOS). By passing light through the lens and color filter array (CFA), the real image is transformed into the mosaic-like RGB image projected on the digital image sensor array. The raw RGB image then is reconstructed as a meaningful image for human perception by de-mosaic and color correction. For sensing applications, the additional image processing to extract object is required for every image pixel. Moreover, it should be noted that imaging processing consumes a lot of memory space and most of computation resources in a computer system. The algorithm mostly uses either temporal or spatial information in the image sequences to perform moving object detection, and the most commonly used approach is pixel intensity. Basically there are three techniques which are used for the motion detection in the sequence of the images such as temporal difference, optical flow and background subtraction algorithm. Out of these background subtraction algorithm is used.

Background subtraction is based on the comparison between input video frame and the background to detect the moving objects, or foreground [1,2]. The major methods single Gaussian background model, mixed Gaussian background model [3,4,5,6] kernel density estimation, etc [7]. These methods are easy and fast. But there is still some problems. The major external condition is the low detection accuracy due to the change of light and the noise. The distinctive limitations of the method are hollow space, streak phenomena, stretched moving objects, etc.

Every year number of papers publishing about the algorithm of motion detection and extraction of moving objects. Still there is no general segmentation theory now. Existing segmentation methods are limited for special problems. For practical application, background subtraction is widely used which improves the general background subtraction by using dynamic threshold method.

In this paper we proposed Robust Temporal Averaging Method (RTAM), there we modified two variants of the general Temporal Averaging Method” (TAM) [8,9].

The rest of this paper is organized as follows: In section 2 we describe Temporal Averaging Method. In section 3 we present our proposed method. In section 4 we provide performance Evaluation and finally in section 5 our conclusion.

2 Temporal Averaging Method (TAM)

The first step of the original TAM method is to create the background model. It represents relatively unmoving part of the scene. For each frame a new background model $B_t(x, y)$ is estimated from [10] as follows as,

$$B_{t+1}(x,y) = \lambda C_{t+1}(x,y) + (1-\lambda)B_{t+1}(x,y) \tag{1}$$

where $C_t(x, y)$ is the current pixel value, $t$ is the frame number, $(x, y)$ is the pixel location in the image and $\lambda$ is learning rate. Then the difference $D(t, x, y)$ between the current frame and the background is given by,

...
\[ D_{t+1}(x,y) = \left| C_{t+1}(x,y) - B_{t+1}(x,y) \right| \]  

(2)

The pixels whose difference value is higher than a given threshold \( T \) are classified as a foreground. The algorithm estimates the pixels as foreground \( F_t(x,y) \) from [10],

\[ F_{t+1}(x,y) = \begin{cases} 0, & D_{t+1}(x,y) \leq T \\ 1, & D_{t+1}(x,y) > T \end{cases} \]  

(3)

### 3 The Proposed Algorithm

When lighting of the scene is constant original TAM method gives relatively accurate in foreground-background estimation results. When sudden illumination changes or repeating backgrounds such as waving trees, results in high number of false positive pixels and lower levels of quality parameters. The main problem with the original method is in the speed of updating the background model determined by \( \lambda \). The TAM method assumes that \( \lambda \) is equal for all pixels in the current frame. This is not the best option when the background is changing very fast. So, if the speed of updating could be robust to each pixel difference \( D_t(x, y) \) there would be faster algorithm reaction to sudden light changes and repeating backgrounds. Furthermore in very noisy and dynamic background scenes, the above mentioned approach wouldn’t be enough. To reduce the number of the false positives pixels in the estimated scene an robust threshold will be applied. In the original algorithm the threshold \( T \) is constant (3). Like in the first approach, we make the threshold \( T \) robust to each pixel difference \( D_{t(x,y)} \).

In the proposed algorithm we are going to change two modifications.

**Modification 1:**

Step 1: The background \( B_t(x, y) \) is estimated by,

\[ B_{t+1}(x,y) = \lambda_{t(x,y)} C_{t(x,y)} + (1 - \lambda_{t(x,y)}) B_{t(x,y)} \]  

(4)

where \( \lambda(x,y) \) is robust learning rate.

Step 2: Estimate the absolute difference between the current frame and the background. It is given by (2). To modify speed of updating the background to be robust, the parameter \( \lambda_{t(x,y)} \) of (4), should be robust for the activity in each pixel of current frame. The simplest way to determine \( \lambda_{t(x,y)} \) is to use the value of the difference \( D_{t(x,y)} \) (2). High value of \( D_{t(x,y)} \) corresponds to significant variation in the pixel value and \( \lambda_{t(x,y)} \) should be increased. The rule of updating \( \lambda_{t(x,y)} \) is similar to (1):

\[ \lambda_{t+1}(x,y) = \mu \cdot \left( \frac{S_{t+1}(x,y)}{N} \right) + (1 - \mu) \lambda_{t(x,y)} \]  

(5)

where \( \mu \) is learning rate of the speed of updating \( \lambda_{t(x,y)} \) , \( S \) is an user set parameter that determines the range of \( \lambda_{t(x,y)} \) and \( N \) is the dynamic range of the processed signal in levels (which can be intensity or color signal). The goal is to increase the speed of updating the background when sudden changes occur in the background. For example these could be illumination changes, new unmoving objects, noise in the image or repeating backgrounds such as waving trees.

Step 3: Estimate the foreground mask, \( F_{t(x,y)} \) as same as (3).

**Modification 2:**

In this we make threshold \( T \) of the foreground mask \( F_{t(x,y)} \) (3), robust for each pixel of the current frame. Pixels that represent dynamic backgrounds assume frequently high levels of difference \( D_{t(x,y)} \). To prevent occurring false positive alarm the threshold \( T_{t(x,y)} \) should be greater than \( D_{t(x,y)} \). Updating the threshold \( T \) for a current frame \( t \) at a current pixel location \((x,y)\) is given by the rule:

\[ T_{t+1}(x,y) = \left( \frac{2(D_{t(x,y)} + \Delta T)}{c} \right) + \left( 1 - \frac{\lambda_{t(x,y)}}{c} \right) T_{t(x,y)} \]  

(6)

where \( D_{t(x,y)} \) is the current level of the difference, \( \Delta T \) is constant threshold summed over the current value of \( D_{t(x,y)} \). The function of \( \Delta T \) is increasing the threshold over the difference and reducing the number of false positives pixels. The other part of the algorithm is same as the proposed Variant A. The background \( B_{t(x,y)} \) is updated according to (4) The absolute difference \( D_{t(x,y)} \) and the learning rate \( \lambda_{t(x,y)} \) are given by (2) and (5), respectively. The foreground mask is estimated by the modified equation (3),

\[ F_{t(x,y)} = \begin{cases} 0, & D_{t(x,y)} \leq T_{t(x,y)} \\ 1, & D_{t(x,y)} > T_{t(x,y)} \end{cases} \]  

(7)

where \( T_{t(x,y)} \) is the robust threshold from (6).

### 4 Performance Evaluation

The proposed system can be experimented with different settings of adjustable parameters which can be used for performance evaluation.

For measuring accuracy we adopted different metrics, namely Precision, Recall, F1-matrix and similarity[11].

1) **Recall vs Precision :**

\[ \text{Recall} = \frac{tp}{tp + fn} \]  

(8)

Where \( tp \) is the total number of true positives, \( fn \) is the total number of false negatives, and \( (tp + fn) \) indicates the total number of items present in the ground truth.

2) **Precision:** Recall alone is not enough to compare different methods, and is generally used in conjunction with Precision,
that gives the percentage of detected true positives as compared to the total number of items detected by the method.

\[
\text{Precision} = \frac{tp}{tp + fp} \tag{9}
\]

Where \(fp\) is total number of false positive and \((tp + fp)\) indicates the total number of detected items.

b) **Figure of merit**- We considered the F1 metric, also known as Figure of Merit or M-measure that is the weighted harmonic mean of Precision and Recall.

\[
\text{Figure of merit} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \tag{10}
\]

Greatest value of similarity shows accurate detection of moving object.

c) **Similarity**- We considered the pixel-based Similarity measure

\[
\text{Similarity} = \frac{tp}{tp + fp + fn} \tag{11}
\]
d) Final Result:-

![Original video frames and Foreground frames](Fig. 6. Original video frames and Foreground frames)

### 4 CONCLUSION

In this work, Background subtraction algorithm based motion detection using a Matlab coding. Improving segmentation results as well as being able to extract additional information such as frame deference, background subtraction allows for improved object detection. The subtraction of the two image is gives the good results of the moving object in the surveillance area. The resultant subtracted frame contains the information or data from both the input frames. It provides an effective way of detecting moving object.

### REFERENCES


