Comparative Analysis of Moving Object Detection Algorithms

Habib Mohammed Hussien[1(✉)], Sultan Feisso Meko[2], and Negassa Basha Teshale[1]

[1] School of Electrical and Computer Engineering, Addis Ababa Institute of Technology (AAiT), Addis Ababa University (AAU), Addis Ababa, Ethiopia
habibmohammed2001@gmail.com, negasabasha4@gmail.com
[2] Addis Ababa Science and Technology University (AASTU), Addis Ababa, Ethiopia
sultanfeisso@gmail.com

Abstract. Moving object detection plays a key role in surveillance systems, vehicle and robot guidance, regardless of it is a very troublesome task. Detecting as well as tracking objects in the video so as to distinguish motion features has been rising as a concerning research/study area in image processing/computer vision fields. One of the current demanding study area in computer/machine vision domain are humans and vehicles motion video surveillance system in a dynamic environment. It is considered as a big challenge for researchers to design a good detection technique which is computationally efficient and consuming less time. Moving object detection algorithms must be fast, reliable and vigorous to make video surveillance systems so as to avoid terrorism, crime and etc. This paper presents comparison of different detection schemes for segmenting/detecting moving objects from the background environment. The algorithms are adequate for adapting to dynamic scene condition, removing shadowing, and distinguishing/identifying removed objects both in complex indoor and outdoor. These algorithms are frame/temporal differencing (FD), simple adaptive background subtraction (BS), Mixture of Gaussian Model (MoG) and approximate median filter. These algorithms are appropriate for real time surveillance applications and each of them have their own advantages and drawbacks.

Keywords: Surveillance system · Object detection · Segmentation
Temporal differencing · Gaussian mode · Approximate median filtering

1 Introduction

In computer vision/image processing the focus of research that tries to explore, admire and monitor objects over a succession photographs is Video surveillance system. Object detection and/or monitoring are the most essential and challenging duties in bunches of vision system comparable to surveillance, automobile and self-reliant robot navigation. There have been various studies about motion detection, tracking, classification and activity analysis in the lit. Due to dynamic scene in natural scenes like abrupt illumination and change of climate, motions repetitiveness that cause clutter

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2018
F. Mekuria et al. (Eds.): ICT4DA 2017, LNICST 244, pp. 172–181, 2018.
https://doi.org/10.1007/978-3-319-95153-9_16

(tree leaves moving in blowing wind), motion detection is a troublesome issue to process constantly. Numerous algorithms for detecting objects have been developed in surveillance system. The making/attaining of best surveillance acquires fast, reliable/constant, powerful and versatile algorithm for detecting and tracking moving objects. Identifying/distinguishing object movements from a video sequence is a key and critical task in numerous computer/machine vision applications. This paper exhibits a comparative analysis of 4 types of motion detection algorithms for monitoring outdoor and indoor scenes/environments. These methods are frame/temporal differencing, Background subtraction, Mixture of Gaussian Model and approximate median filtering. At last these four algorithms are compared and assessed.

2 Survey of Moving Object Detection Methods

Identifying the moving pixels (foreground) from the environment (background) is very significant and troublesome. The initial step of a surveillance systems is detecting foreground objects and this step requires efficient algorithm so as to develop reliable, robust and fast system. This paper explores the four key algorithms (background subtraction, frame differencing, approximate median filter and adaptive online Gaussian mixture model) for detecting objects and analyze and test their differences as well as their performances as discussed in the following sections.

2.1 Frame Differencing (FD)

This scheme detects moving regions/pixels by pixel-by-pixel difference of consecutive (2 or 3) frames. FD is very good at adapting the scene changes dynamically but cannot detect whole applicable pixels as well as stopped objects in the scene.

According to Lipton et al. [1], a two-frame differencing scheme can estimate the foreground pixels if the Eq. 1.1 satisfies as follows. A pixel $I_t(x, y)$ can be classified as foreground if the difference between $I_t(x, y)$ and $I_{t-1}(x, y)$ is larger than I_{th}

$$|I_t(x,y) - I_{t-1}(x,y)| > I_{th} \tag{1.1}$$

On the other hand the shortcoming of two frame differencing solved using three frame differencing [2] as illustrated by Collins et al. [3]. Let $I_t(x,y)$ denotes the intensity/gray-level value at position (x) and at time instance n of video image sequence $I(x,y)$ in the range [0, 255]. The two- temporal differencing pixel is moving if it meets the following rule:

$$|I_t(x,y) - I_{t-1}(x,y)| > T_h \tag{1.2}$$

Equation 1.2 cannot detect some of the pixels inside the object even if the object moves due to the uniform color regions. The threshold, T_h, is initially set to a predetermined value and later can be updated as follows:

$$Th_{t+1} = \begin{cases} \alpha * Th_t(x,y) + (1-\alpha)(\gamma * |I_t(x,y) - I_{t-1}(x,y)|), (x,y) \in B_g \\ Th_t(x,y), |(x,y) \in F_g \end{cases} \tag{1.3}$$

If α (updating parameter) set to 0, the background holds the image I_{t-1} and this value becomes similar to two-frame temporal differencing. FD algorithm can detect only the exterior pixels and left the interior pixels which results in holes. The FD methods, initially subtract the current pixel from the past/previous one. Then, the value has compared with a particular threshold. Finally, if the result is larger than the assigned value, then the pixel pertains/belongs to the foreground/detected, otherwise, it pertains to the background/not detected. The FD detection technique is described below.

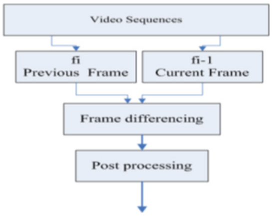

Fig. 1. Block diagram for frame differencing algorithm

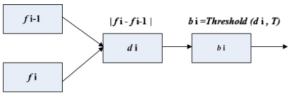

Fig. 2. Temporal differencing algorithm

2.2 Adaptive Mixture of Gaussian Model (MoG)

A dynamic model that can genuinely deal with change of lights, motions repetition, clutter, adding or avoiding objects from the environment and slow motion objects are proposed by Stauffer and Grimson [4]. Since a unimodal model could not manage noise of image acquisition, change of light and etc. for a specific pixel at a time, they used a MoG to denote each pixel in the model [4, 7]. In this model, the values of an individual pixel over time is considered as a "pixel process" and the present history of individual pixel {X1, …, Xt} is modeled by Gaussian mixture model or K. The probability of looking present pixel value then becomes:

$$P(X_t) = \sum_{i.t}^{k} \omega_{i,t} * \eta(X_i, \mu_{i,t}, \sum_{i,t}) \tag{1.4}$$

Where $w_{i.t}$ is an estimated weight of the i^{th} Gaussian $G_{i,t}$ at time t, $\mu_{i,t}$ is mean value of $G_{i.t}$ and $\sum_{i.t}$ is the covariance matrix of $G_{i,t}$.

The Gaussian probability density function can be:

$$\eta(X_t, \mu, \sum) = \left\{ \frac{1}{(2\prod)^{\frac{n}{2}} |\sum|^{\frac{1}{2}}} e^{\frac{-1}{2}(X_t-\mu_t)^T \sum^{-1}(x_t-\mu_t)} \right\} \tag{1.5}$$

The value of K is decided by using the existed memory as well as power calculation/computation. Moreover, the covariance matrix for efficient computation, is described in Eq. 1.6 below, which assumes a unique (red, green and blue) color components which have similar variance.

$$\sum\nolimits_{k,t} = \alpha_k^2 I \tag{1.6}$$

There are certain procedures for detecting foreground pixels. Initially, K distributions are initialized with defined mean, large variance and minimum prior weight. Then sequence, type and RGB vector of the image will be estimated/determined against K till match is available. Matching is defined as a pixel value in the range $\gamma = 2.5$ standard deviation. Next, the existing weights of K distributions will be updated as follows:

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{1.7}$$

Where

α is the learning rate,

$M_{k,t}$ is 1 for the matching Gaussian and 0 for the remaining distributions.

After this step the existing weights of the distributions are normalized and the matching Gaussian are updated as follows for the new observation:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho(X_t) \tag{1.8}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t-\mu_t)^T(X_t-\mu_t) \tag{1.9}$$

Where

$$\rho = \sigma\eta(X_t|\mu_k - \sigma_k) \tag{1.10}$$

If no match is found, the Gaussian distribution with the least probability is replaced with a new distribution. Then the first B distributions are chosen as the background model, where B = arg $\min_b$ and T is the minimum portion of the pixel data that should be accounted.

$$B = \arg\ \min_b(\sum \sum_{k=1}^{b} \omega_k > T) \tag{1.11}$$

If T is very small, the background is unimodal. Accumulated pixels define the background Gaussian distribution whereas scattered pixels are classified as foreground.

When the new frame incomes at times t + 1, a match test is made for each pixel if the Mahalanobis distance is

$$sqrt((X_{t+1} - \mu_{i,t})^T \sum\nolimits_{i,t}^{-1} (X_{t+1}. - \mu_{i,t})) < k\sigma_{i,t} \tag{1.12}$$

Where k is a constant threshold equal to 2.5.
Then, two cases can occur:

$$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t} + \alpha \tag{1.13}$$

Where α is a certain constant learning rate

$$\mu_{i,t+1} = (1 - \rho)\mu_{i,t} + \rho.X_{t+1} \tag{1.14}$$

$$\sigma^2_{i,t+1} = (1 - \rho)\sigma^2_{i,t} + \rho(X_{t+1} - \mu_{i,t+1}).(X_{t+1} - \mu_{i,t-1})^T \tag{1.15}$$

Where

$$\rho = \alpha.\eta(X_{t+1}, \mu_t, \sum\nolimits_t) \tag{1.16}$$

For the unmatched component, μ and $\sum$ are unchanged, only the weight is replaced by:

$$\omega_{j,t+1} = (1 - \alpha)\omega_{j,t} \tag{1.17}$$

Once the parameters maintenance is made, foreground detection can be made and so on [5, 6].

2.3 Background Subtraction (BS)

A Scheme used for object detection in motionless scenes/environment is BS [6]. BS is performed by subtraction of current/present and reference/background frame to detect moving parts/regions. The pixel difference larger than the threshold is assumed as foreground/detected. After creating a detection pixels map, some morphological post processing (dilation/expanding, erosion/shrinking, and closing) are performed so as to minimize noise and enhancing the detected/foreground regions. The background is updated with new images over time to adapt a changing environment.

The simple type of BS method was presented by Heikkila and Silven as shown in Eq. 1.18 below. A pixel at location (x, y) in the present/current image I_t is marked as foreground if

$$|I_t(x, y) - B_t(x, y)| > T \tag{1.18}$$

is fulfilled where T is a threshold. To update B_T, we use an Infinite Impulse Response (IIR) filter as below:

$$B_{t+1} = \alpha I_t + (1 - \alpha) B_t \tag{1.19}$$

BS scheme is too much sensitive to change of dynamic scenes but performs well at extracting moving parts regardless of detecting stooped objects. BS algorithm is partially motivated by the research exhibited in [3] as follows.

Let $I_t(x, y)$ denotes intensity value at pixel location (x, y) and image sequence $I(x, y)$ in the range [0, 255]. Let $B_g(x, y)$ be background intensity value for pixel location (x, y) determined over time from video images $I_0(x, y)$ through $I_{t-1}(x, y)$. As BS method indicates, a pixel at location (x, y) in the present/current video image belongs to foreground/detected if it fulfills

$$|I_t(x, y) - B_g(x, y) > T_h| \tag{1.20}$$

BS provides the most complete features of data. This scheme is successful for lots of surveillance scenarios where objects moving endlessly and the background is visible. The block diagram of BS is shown below.

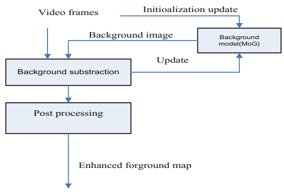

Fig. 3. Block diagram for background subtraction

BS detection algorithm can be described as below.

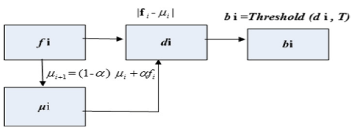

Fig. 4. Background subtraction algorithm

Foreground (region of motion) will be attained, if the difference is larger than thresh.

$$|I(x, y) - Bg(x, y)| > thresh \tag{1.21}$$

Initially, Bg is the first frame and threshold can be initialized automatically using global techniques. For foreground pixels, FG,

$$Fg(x,y) = \begin{cases} 1, & \textit{for fr\_diff}(x,y) > \textit{thresh} \\ 0, & \textit{otherwise} \end{cases} \quad (1.22)$$

Where

$$Bg_{t+1}(x,y) = \begin{cases} \alpha * Bg(x,y) + (I - \alpha) * I_t(x,y) \in BG \\ Bg(x,y), (x,y) \in FG \end{cases} \quad (1.23)$$

$$\textit{thresh} = \begin{cases} \textit{thresh} \quad \textit{for} \quad (x,y) \in FG \\ \alpha * \textit{thresh} + (1-\alpha)(\gamma * (\textit{fr\_diff}(x,y))) \quad \textit{for} \quad (x,y) \in BG \end{cases} \quad (1.24)$$

From the above equations background and threshold are selectively update at each new frame for non-moving pixels α = rate of adaptation, γ = local temporal average.

2.4 Approximated Median Filter Method

A recursive filter for evaluating an image pixels median is proposed by McFarlane and Schofield [8]. In this scheme the running estimate of median is augmented by 1 if the input pixel is larger than the estimate and decremented by 1 if it is lesser than the estimate. This estimate ultimately converges to median and the median filtering buffers the leading N frames of the video input. then the background frame is calculated from buffered frame and the foreground/detected pixel can be obtained by subtracting the background from the current frame as indicated in Eqs. 1.25 and 1.26 below.

$$F_r > B_g \rightarrow \sum\nolimits_{n-1.1}^{l,m} B_g(l,m) + 1 \quad (1.25)$$

$$F_r < B_g \rightarrow \sum\nolimits_{n-1.1}^{l,m} B_g(l,m) - 1 \quad (1.26)$$

Side effects: It did not provide same results in all conditions. But, it needs minimum memory.

2.5 Thresholding

Gray scale image can be obtained from binary image by applying thresholding. A binary image composed of 2 colors, black (zero) or white (one). A careful selection of threshold value is required so as to separate the object from the background. Mathematically thresholding can be expressed as follows.

$$f(x,y) = \begin{cases} 1 & ('255') & f(x,y) > T_h \\ 0 & ('0') & f(x,y) < T_h \end{cases} \quad (1.27)$$

Where T = Threshold value

Any point of (x,y) for which $fr(x,y) \geq$ thresh is called an object point: otherwise, it is a background point. In other words, the thresholded image $ga(x,y)$ is defined a

$$g_a(x, y) = \begin{cases} 1 & if \quad fr(x, y) \geq thresh \\ 0 & if \quad fr(x, y) \leq thresh \end{cases} \tag{1.28}$$

2.6 Post Processing

The outputs of from above algorithms for foreground detection contains noise and therefore are not suitable for extra processing unless post processing is applied, to directly enhance the quality of the segmentation mask.

2.6.1 Morphological Operations (MO)

MO can be erosion/removing, dilation/adding, and hit/miss or cascaded form of them [10]. MO is applied on images with either 0 or 1 pixel values. Erosion scheme is used to shrink extra white noise pixels as well as diminish, the edges using a mask of the same size as shown in Fig. 5. Dilation is used to expand/enlarge the binary objects as shown Fig. 5 below.

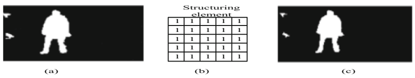

Fig. 5. Results of erosion, (a) detected object, (b) structuring element, (c) erosion output.

From the above results we can see that post processing (morphological operations) can remove noises.

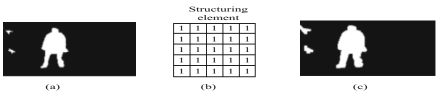

Fig. 6. Results of dilation, (a) output after erosion, (b) structuring element, (c) output after dilation

3 Experimental Results and Discussion

3.1 Graphical User Interface (GUI) Design

The GUI enabled us to start, stop and show the program and its results. During the moment the starting button is clicked the system will be running and the selected program will be called to carry out the computations till the stopping button is clicked and the output can be performed as detection.

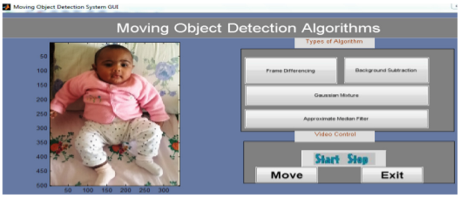

Fig. 7. GUI layout design

3.2 Comparison of Different Detection Algorithms Result

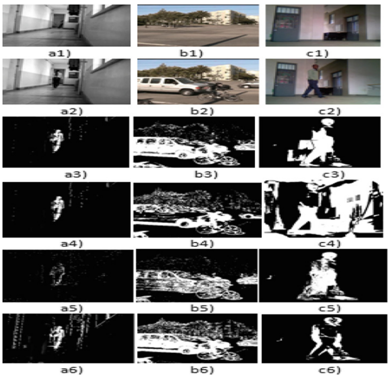

Fig. 8. Outputs of different algorithms, ((a1), (b1), and (c1)) are background images, ((a2), (b2), and (c2)) are video inputs, (a3), (b3), and (c3) are approximate median filter outputs, ((a4), (b4) and (c4)) are background subtraction outputs, ((a5), (b5) and (c5)) are mixture of Gaussian outputs, ((a6), (b6) and (c6)) are frame differencing outputs

3.3 Morphological Operations Detection Results

As shown in Fig. 9 below results are obtained after applying MO. MO take its input by combining the structuring element together with binary images by using a set operators.

Fig. 9. Morphological operations detection results

4 Conclusions

This paper presents a widespread review of visual surveillance systems describing its phases of object detection. Various approaches of detection and their representation have been explained and compared. Object detection techniques like background subtraction, frame differencing, mixture of Gaussian and approximate median filter are briefly described and a comparative study is also presented. The above four moving object detection methods have compared based on their basic principles, computational time, accuracy and drawbacks are also described.

References

1. Lipton, A.J., Fujiyoshi, H., Patil, R.S.: Moving target classification and tracking from real-time video. In: Proceedings of Workshop Applications of Computer Vision, pp. 129–136 (1998)
2. Wang, L., Hu, W., Tan, T.: Recent developments in human motion analysis. Pattern Recogn. **36**(3), 585–601 (2003)
3. Collins, R.T., et al.: A system for video surveillance and monitoring: VSAM final report. Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May 2000
4. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 246–252 (1999)
5. Horprasert, T., Haritaoglu, I., Wren, C., Harwood, D., Davis, L., Entland, A.: Real-time 3D motion capture. In: Workshop on Perceptual User Interfaces (PUI 1998), San Francisco, California, November 1998, pp. 87–90 (1998)
6. Atev, S., Masoud, O., Papanikolopoulos, N.: Practical mixtures of Gaussians with brightness monitoring. In: IEEE Conference on Intelligent Transportation Systems, Proceedings (ITS 2004), 423–428 (2004)
7. Zang, Q., Klette, R.: Parameter analysis for mixture of Gaussians. CITR Technical Report 188, Auckland University (2006)
8. McFarlane, N., Schofield, C.: Segmentation and tracking of piglets in images. Mach. Vis. Appl. **8**(3), 187–193 (1995)
9. Haritaoglu, I., Harwood, D., Davis, L.S.: W4: who? When? Where? What? A real-time system for detecting and tracking people. In: Proceedings of 3rd Face and Gesture Recognition Conference, pp. 222–227 (1998)
10. Heijden, F.: Image Based Measurement Systems: Object Recognition and Parameter Estimation. Wiley, Hoboken (1996)