BACKGROUND FOREGROUND SEPARATION FOR SURVEILLANCE SYSTEM

[1]Neha Chaudhary
M.E, Electronics and Communication Engineering,
Sarvajanik college of Engineering and Technology, Surat,
Email:nehapc.chaudhary@gmail.com

Abstract— Background subtraction is one of the important and useful step for moving object detection, especially in the domain of video surveillance. There are various methods have been developed over the recent years. This paper gives survey of the recent approaches which concern to statistical background modeling techniques. These background subtraction techniques have benefits and limitations in terms of noise, illumination change. To overcome this problem, this paper provides a review of Background subtraction methods and comparison mainly based on three factors speed, memory requirements and accuracy.

Index Terms— Background modeling, Mixture of Gaussians, Running Gaussian Average.

I. INTRODUCTION

Background and foreground separation method is one of the key technique for automatic video surveillance analysis. Background subtraction in surveillance and security applications is very important, because this is the first fundamental step and critical task in detecting and identifying objects. Background subtraction also reduces the search space in the video frame for the object detection unit by filtering out the uninteresting background. In this paper, we compare various background sub- traction methods for detecting moving object in video sequences. There are many challenges in developing a good Background Subtraction algorithm. such as quick lighting variations, heavy occlusion, foreground fragments, slow moving or stopped object etc.While complicated techniques often produce superior performance, our experiments show that simple techniques such as adaptive median filtering can produce good results with much lower computational complexity.

II. STEPS OF BACKGROUND SUBTRACTION

Shown in fig 1 dataflow diagram of background subtraction algorithm. Which consists of four major processing blocks: pre-processing, background modeling, foreground detection, data validation.

Background modeling

Background modeling is at the heart of any background subtraction algorithm. In this section, describe the different background modeling techniques considered in our comparative study.

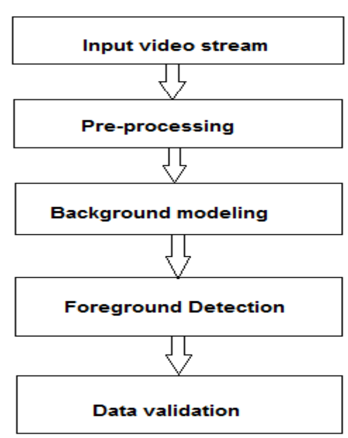

Fig 1: Fig:flow diagram of background subtraction algoritham

III Non recursive technique

Frame difference

Frame difference is the simplest form of background subtraction. The current frame is simply subtracted from the previous frame, if the difference in pixel values for a given pixel is greater than a threshold T_s, the pixel is considered part of the foreground otherwise background

$$frame_t - frame_{t-1} = T_s \quad (1)$$

Where: $(x,y)T$ is the current frame, $I(x,y)T-1$ is the previous frame and Th is the predefined threshold.

Median filter

Median Filtering is one of the most commonly-used background modeling techniques proposed by [5]. The background model is defined to be the median at each pixel location of the frames in the buffer. Assuming that the background is more likely to appear in a scene, the median can be used as the a prefect tool for modeling the background. The background is then represented by the group of the median values in each pixel location. The formula below describes the method:

$$I(x,y)_t - B(x,\ y) > Th \quad (2)$$

$I(x,y)t$ is the current frame, $B(x,y)$ is the median value in (x, y) location from the first buffered frame to the previous frame Th is the predefined threshold.

IV RECURSIVE technique

Approximated Median filter

In approximated median filtering ,the previous N frames of video are buffered, and the background is calculated by the median of this buffered frames. So the background is subtracted from the current frame And threshold to determine the foreground pixels. In median Median filtering has been shown to be very robust and to have performance comparable to higher complexity methods. In this method storing and processing many frames of video requires large amount of memory.

The approximate median method works as such: if a pixel in the current frame has a value larger than the corresponding background pixel, the background pixel is incremented by 1. Likewise, if the current pixel is less than the background pixel, the background is decremented by one. In this way, the background eventually converges to an estimate where half the input pixels are greater than the background, and half are less than the background—approximately the median (convergence time will vary based on frame rate and amount movement in the scene.)

As you can see, the approximate median method does a much better job at separating the entire object from the background. This is because the more slowly adapting background incorporates a longer history of the visual scene, achieving about the same result as if we had buffered and processed *N* frames.

To get a feel for how the background model works, sometimes it's useful to visualize it. Below is a video of the background model. Rather ghostlike if you ask me.

This method is a very good compromise. It offers performance near what you can achieve with higher-complexity methods (according to my research and the academic literature), and it costs not much more in computation and storage than frame differencing.

Running Gaussian average method

In have proposed to model the background independently at each (i,j) pixel location. The model is based on ideally fitting a Gaussian probability density function (pdf) do on the last n pixel's values. In order to avoid fitting the pdf from scratch at each new frame time, t, a running (or on-line cumulative) average is computed instead as:

$$\mu = \alpha I_t + (1 - \alpha)\mu_{t-1} \quad (3)$$

Where I, is the pixel's current value and , μ_t the previous average; α as an empirical weight often chosen as a tradeoff between stability and quick update. Although not stated explicitly in the other parameter of the Gaussian pdf, the standard deviation σ_t ,can be computed similarly. In addition to speed, the advantage of the running average is given by the low memory requirement: for each pixel, this consists of the two parameters (μ_t,σ_t) instead of the buffer with the last n pixel values. At each f frame time, the I, pixel's value can then be classified as a foreground pixel if the inequality holds; otherwise, I, will be classified as background. The name background subtraction used to commonly indicate this set of techniques actually derives from equation:

$$|I_t - \mu_t| = K\sigma_t \quad (4)$$

The model in is updated also at the occurrence of such foreground values. For this reason, they propose to modify the model update as:

$$\mu_t = M\mu_{t-1} + (1 - M)(\alpha I_t + (1 - \alpha))\mu_{t-1} \quad (5)$$

where the binary value M is 1 in correspondence of a foreground value, and 0 otherwise. This approach is also known as selective background update.

As the model in was proposed for intensity images, extensions can be made for multiple-component colour spaces such as (R,G,B), (Y,U,V), and others. Moreover, if real-time requirements constrain the computational load, the update rate of either, μ or σ can be set to less than that of the sample (frame) rate. However, the lower the update rate of the background model, the less a system will be able to quickly respond to the actual background dynamic.

Gaussian mixture model

Consider a particular pixel (x; y). For notational convenience we define C_i to be the value of the pixel in frame i, i.e.

where I is the image sequence and Ii is the current frame. Note that Ci is assumed to be a 3-element vector containing, for example, the red, green and blue components of the pixel. The history of pixel (x, y), at any given time t, is therefore given by the ordered set An aspect of variation occurs if moving objects are present in the scene. A moving object will normally produce more variance than a stationary object.

$$\{C_1 + C_2 \ldots\ldots\ldots\ldots C_t\} \quad (6)$$

We model this history by a mixture of K Gaussian distributions, so that the probability of observing the current pixel value is

$$P(C_t) = \sum_{i=1}^{k} \omega_{i,t}\eta\ (C_t\ ;\ \mu_{i,t}, \sum i, t) \quad (7)$$

where K is the number of distributions, $\omega_{i,t}$ is an estimate of the weight (what portion of the data is accounted for by this Gaussian), $\mu_{i,t}$ is the mean value, $\Sigma_{i,t}$ is the covariance matrix of the ith Gaussian in the mixture at time t, and η is a Gaussian probability density function.

$$\eta(c, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}\,|\Sigma|^{1/2}} e^{-\frac{1}{2}(c-\mu)^T\Sigma^{-1}(c-\mu)} \quad (8)$$

Here d is the number of dimensions of c, in this case d = 3.

K is determined by the available memory and the computational power, normally between 3 and 5 is used. For computational reasons the covariance matrix is assumed to be of the form:

$$\Sigma_{k,t} = \sigma^2 k, t\, I \quad (9)$$

where I is the 3 *3 identity matrix.

This assumes that the red, green and blue pixel values are independent and have the same variances. It is most probably not true, but this assumption allows us to avoid a costly matrix inversion at the expense of some model accuracy.

The distribution of recently observed values of each pixel in the scene is characterized by a mixture of Gaussians. A new pixel value will normally be represented by one of the components of the mixture model and can be used to update the model.

Every new pixel value, Ct, is checked against the existing K Gaussian distributions until a match is found. It is a match if the pixel value is within 2.5 standard deviations of the mean of that distribution. This threshold can be changed slightly with little effect on performance.

If none of the K distributions match the current pixel value, the least probable distribution is replaced that has a distribution with the current value as its mean. The distribution initially has high variance and low prior weight. The prior weights of the K distributions at time t, $\omega_{k,t}$ are adjusted as follows:

$$\omega_{k,t} = \alpha M_{k,t} + (1 - \alpha)\omega_{k,t-1} \quad (10)$$

where α is the learning rate and $M_{k,t}$ is 1 for the model that matched and 0 for the rest. After this approximation, the weights are normalized. The time constant 1/α determines the speed at which the distribution parameters change.

The μ and σ parameters for the unmatched distributions remain the same, but the parameters of the distribution that matches the new observation are updated as follows:

$$\mu_{k,t} = \rho c_t + (1-\alpha)\mu_{k,t-1},$$
$$\sigma 2_{k,t} = \rho(c_{k,t} - \mu_{k,t})^{\wedge}T(c_{k,t} - \mu_{k,t}) + (1-\alpha\sigma 2_{k,t-1},$$

(11)

Eq.(11) are effectively a type of low-pass filter, except that only the data which matches the model is included in the estimation.

Stauffer and Grimson found very useful case for multi-valued background model for multiple background objects. In the context of a traffic surveillance system, proposed to model each background pixel using a mixture of three Gaussians corresponding to road, vehicle and shadows.

Then, the Gaussians are manually initialized: the darkest component is labeled as shadow, in the remaining two components; the one with the largest variance is labeled as vehicle and the other one as road.

COMPARISION OF BACKGROUND SUBTACTION TECHNIQUES

Type of approaches	Methodology	Speed	Memory Requirements	Accuracy
Recursive technique	Running Gaussian average	Fast	Low	Acceptable
	Gaussian mixture model(GMM)	Intermediate	Intermediate	Good
	Approximated median filter	Intermediate	High	Acceptable
Non-Recursive technique	Frame differencing	Fast	Low	Better
	Median filtering	Fast	High	Acceptable

III. CONCLUSION

In this paper, we have presented statistical background techniques. This allows comparing the statistical BS methods complexity in terms of speed, memory requirements and accuracy. Simple methods such as the running Gaussian average and median filter offer acceptable accuracy while achieving a high frame rate and both method requires less memory

REFERENCES

[1] Komagal, E., A. Vinodhini, A. Srinivasan, and B. Ekava. "Real time Background Subtraction techniques for detection of moving objects in video surveillance system." *In Computing, Communication and Applications (ICCCA), 2012 International Conference on, pp. 1-5. IEEE, 2012*J.

[2] Cheung, Sen-Ching S., and Chandrika Kamath. "Robust techniques for background subtraction in urban traffic video." *In Proceedings of SPIE, vol. 5308,IEEE 2004.*

[3] Piccardi, Massimo. "Background subtraction techniques: a review." *In Systems, man and cybernetics, International conference on, vol. 4, IEEE, 2004.*

[4] Chen, Yu-Ting, Chu-Song Chen, Chun-Rong Huang, and Yi-Ping Hung. "Efficient hierarchical method for background subtraction." *Pattern Recognition, Elsevier 2007*

[5] M. Hedayati, Wan Mimi Diyana Wan Zaki_, Aini Hussain, "A Qualitative and Quantitative Comparison of Real-time Background Subtraction Algorithms for Video Surveillance Applications",*Journal of Computational Information Systems 2012*