Video Detection using Mixture of Gaussian Model

Ravdeep Singh Johar
Rochester Institute of Technology
One Lomb Memorial Drive, Rochester, NY 14623-5603
rsj7209@rit.edu

ABSTRACT

Detecting moving objects in a video is a fundamental and critical task required in the field of computer vision applications. A commonly used technique is to use background subtraction, which is a method typically used to segment moving regions in image sequences taken from a static camera by comparing each new frame to a model of the scene background. I present a non-parametric background model and a background subtraction approach. The model can handle situations where the background of the scene is cluttered and not completely static but contains small motions such as tree branches and bushes. The model estimates the probability of observing pixel intensity values based on a sample of intensity values for each pixel. The model adapts quickly to changes in the scene which enables very sensitive detection of moving targets.

Keywords
Video Detection, Image Processing, Masking, Morphological operations, Gaussian Models, Mixture Models.

1. INTRODUCTION

Today, security is of much importance and lots of electronic equipments are being used in security applications which monitor continuously the movements of persons or vehicles and report when a non predefined event takes place. A human observation based system for implementing the same has several disadvantages, where a person has to observe the camera's output on a TV and detect when any unexpected events occur. This project deals with detection of moving object in a video which is taken by a stationary camera in dynamic situations where the background can change frequently, the algorithm proposed needs to adapt to the changes in the background quickly and should not have false detection of background as foreground.

In this paper I present a parametric model for background subtraction using the mixture of Gaussian model. This can be used to detect any moving object in a video. An AVI file is read and it reduced to frames, these frames are converted to grayscale, then the Gaussian model is applied to the frames, the background and foreground is separated, morphological operations are applied to improve the results of the foreground. Finally the frames are then used to form a video which displays only the moving objects in the video. The paper is organized

as follows: the survey of background subtraction algorithms can be found in Section 2. The principles of mixture models and Gaussian models are presented in Section 3. Section 4 has the actual algorithm implemented. The actual results are showed in section 5. Finally, I conclude our paper and discuss future work in Section 6 and 7 respectively. Section 8 contains the acknowledgement, Section 9 states the references followed by Section 10, Author Biography.

2. Detecting Motion in Video:

Motion detection in a video in simple words is to compare the consequent frames in the video and detect if there are any changes that occur in the subsequent frames. This type of detection can enable a surveillance camera to start recording the video if there is any motion in the video; this is also called as activity detection, which can trigger an alarm if required.

The most basic technique which can achieve motion detection is background subtraction, wherein, the foreground is separated from the background, the background is nothing but the a frame which consists of only normal situation if anything new is added to this frame, it has to be detected. Thus continuous subtraction of the frame from the background will give us the abnormal motion detection but this method is very inefficient, it can't adapt to dynamic situations, for example if the lighting of the room is changed, this would be an abnormal condition.

Another similar technique is frame difference, where the current frame is simply subtracted from the previous frame, and 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. The biggest flaw with this system is that if there is anything in the frame for more than two frames it would not be

detected anymore because it becomes the part of the background and for moving objects with uniformly distributed intensity values; it will also be a part of the background and wouldn't be detected. Using this method does have its advantages, it is a very light weight in terms of computational load and the background is very adaptive since it changes every frame but fixing a threshold value for real time video detection is challenging, it is something which can be fixed empirically.

The idea of frame difference can be further improved by using a median filter, in which the previous N frames that are already buffered, are being used to calculate mean value, the subsequent frame is then subtracted from this mean value and is thresholded to determine the moving objects. The median filter technique is very robust and is effective than the much more complex algorithms but it requires a lot of computations as every frame is stored and mean is calculated.

Researchers from United Kingdom, N.J.B. McFarlane and C.P. Schofield developed a much more efficient recursive approximation of the median filter, the approximate median works as such, if a pixel in the current frame has a value larger than the corresponding background pixel, the

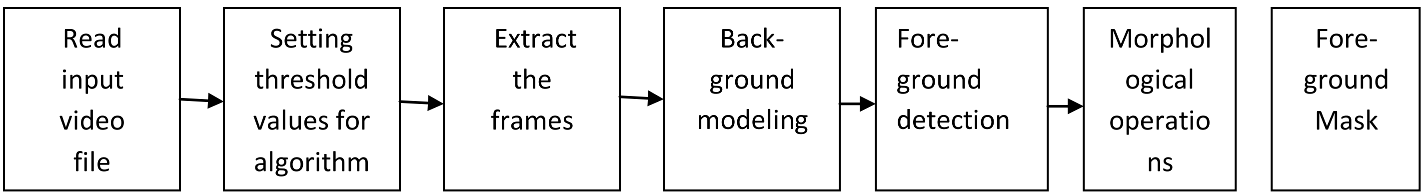

Fig 1: Flowchart for detecting objects

Background model pixel is incremented by 1. Likewise, if the current pixel is less than the background pixel, the background model is decremented by one. In this way, the background model eventually converges to an estimate where half the input pixels are greater than the background, and half are less than the background. This approach does a very good job of detecting objects but if the moving object is at very high speed it will leave behind a trail because the background is slowly adapting and incorporates a longer history of the visual scene.

In this paper, I implement background modeling using mixture of Gaussian in which the background model is parametric and not an actual background. Each pixel location is represented by a mixture of Gaussian functions that sum together to form a probability distribution function *F*.

$$f(x) = \frac{1}{\sigma\sqrt{2\Pi}} e^{-(x-\mu)/(2\sigma^2)} \quad (1)$$

The mean *u* of each Gaussian function is a guess of the pixel value in the next frame—we assume here that pixels are usually background. The weight and standard deviations of each component are measures of our confidence in that guess (higher weight & lower σ = higher confidence). There are typically 3-5 Gaussian components per pixel—the number typically depending on memory limitations.

To determine if a pixel is part of the background, we compare it to the Gaussian components tracking it. If the pixel value is within a scaling factor of a background component's standard deviation σ, it is considered part of the background. Otherwise, it's foreground.

3. BACKGROUND MODELING USING MIXTURE OF GAUSSIANS

3.1 General Principle of Mixture Model

A mixture model is a probabilistic model for representing the presence of sub-populations within an overall population, without requiring that an observed data-set should identify the sub-population to which an individual observation belongs. Formally a mixture model corresponds to the mixture distribution that represents the probability distribution of observations in the overall population. However, while problems associated with "mixture distributions" relate to deriving the properties of the overall population from those of the sub-populations, "mixture models" are used to make statistical inferences about the properties of the sub-populations given only observations on the pooled population, without sub-population-identity information.

3.2 Principal of Mixture of Gaussians

First, each pixel is characterized by its intensity in the RGB color space. Then, the probability of observing the current pixel value is considered given by the following formula in the multidimensional case:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} . \eta(X_t, \mu_{i,t}, \sum\nolimits_{i,t}) \quad (2)$$

Where the parameter K is the number of distributions, $\omega_{i,t}$ is a weight associated to the i th Gaussian at time *t* with mean $m_{i,t}$ and standard deviation $\mu_{i,t}$. η is a Gaussian probability density function:

$$\eta(X_{t,}\mu,\Sigma) = \frac{1}{(2\Pi)^{n/2}\left|\Sigma\right|^{1/2}} e^{-\frac{1}{2}(X_t-\mu)\Sigma^{-1}(X_t-\mu)} \quad (3)$$

For computational reasons we assume that the RGB color components are independent and have the same variances. So, the covariance matrix is of the form:

$$\Sigma_{i,t} = \sigma_{i,t}^2 I \quad (4)$$

So, each pixel is characterized by a mixture of K Gaussians. Once the background model is defined, the different parameters of the mixture of Gaussians are initialized. The parameters of the MOG's model are the number of Gaussians K, the weight $\omega_{i,t}$ associated to the i th Gaussian at time t, the mean $m_{i,t}$ and the covariance matrix.

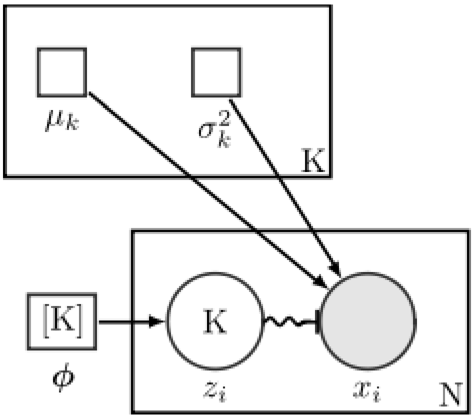

Fig 3: Gaussian mixture model using plate notation. Smaller squares indicate fixed parameters; larger circles indicate random variables. Filled-in shapes indicate known values. The indication [K] means a vector of size *K*.

4. Proposed Algorithm

The following steps are followed by the algorithm proposed in this paper.

4.1 Threshold Values

Proper threshold values have to be chosen for background, standard deviation and area of the moving objects. The statistical parameter standard deviation is used in the processing of removing the shadow of the moving object. In this algorithm threshold value of background chosen as 250 pixels, standard deviation is 0.25 and area of the moving object is 8 pixels.8*8 pixel is taken as one block in this algorithm. The other parameters are set as mentioned in table.

Table 1: Parameters and their values used for this paper.

Parameters	Test Values
Number of Gaussian Components (K)	3
Initial Variance (σ)	36
Initial weight (ω)	0.1
Adaptation rate (α)	0.01
Weight Threshold (T_w)	0.25
Deviation Threshold (T_d)	2.5

4.2 Input Video

The input video is in avi format, where avi stands for audio video interleave. The AVI files stores audio and video under the RIFF(Resource Interchange File Format) container format

The input video format is avi. avi stands for audio video interleave. An AVI file actually stores audio and video data under the RIFF (Resource Interchange File Format) container format. In AVI files, audio data and video data are stored next to each other to allow synchronous audio with- video playback. Audio data is usually stored in AVI files in uncompressed PCM (Pulse-Code Modulation) format with various parameters. Video Data is usually stored in AVI files in compressed format with various codecs and parameters. The aviread, aviinfo functions are used to read the input video avi format.

4.3 Extraction

After reading the input video file, the individual frames are extracted using the cdata from the aviread object. The frame is then converted to grayscale using the rgb2gray for further processing.

4.4 Applying Mixture of Gaussian Model:

1. Compare the input pixels to the means u_i of their associated components. If a pixel value is close enough to a given component's mean, that component is considered a matched component. Specifically, to be a matched component, the absolute difference between the pixels and mean must be less than the component's standard deviation scaled by a factor D.
2. Then the component variables are updated (w, u, σ) to reflect the changes in the pixel values, for every matched component, a set of equation increase our confidence in the component (w increases, σ decreases, and *u* is nudged towards the pixel value). For non-matched components, the weights decrease exponentially (u and σ stay the same). How fast these variables change is dependent on a learning factor *p* present in all the equations.

$$\omega_{i,t-1} = (1-\alpha)\omega_{i,t-1} + \alpha \quad (5)$$

$$\mu_{i,t-1} = (1-\rho)\mu_{i,t-1} + \rho I_t \quad (6)$$

$$\sigma^2_{i,t-1} = (1-\rho)\sigma^2_{i,t-1} + \rho(I_t - \mu_{i,t-1})^2 \quad (7)$$

Updated values for matched components

$$\omega_{i,t-1} = (1-\alpha)\omega_{i,t-1} \quad (8)$$

Updated values for non-matched components

$$p \approx \frac{\alpha}{\omega_{i,t}} \quad (9)$$

Learning Factor *p*

3. Determine which components are part of the background model
   - 3.1 Order the components according to a confidence metric *w/σ*, which rewards high *w*'s and low *σ*. We do this because we want to keep only the M most confident guesses.
   - 3.2 Apply a threshold to the component weights w.

$$\sum_{i=1}^{K} \omega_{k,t<=\tau} \quad (9)$$

Threshold Value

   - 3.3 The background model is then the first M components (in order of highest to lowest *w/σ*), whose weight w is above the threshold. M is the maximum number of components in the background model, and reflects the number of modes we expect in the background probability distribution function *f* (or it may reflect our computational resource limitations).
   - 3.4 Determine foreground pixels. Foreground pixels are those that don't match any components determined to be in the background model.

4.5 Re-combining the frames to form the video s:

After applying the mixture of Gaussian algorithm, we convert the resultant images to frames and after the entire video is processed, movietoavi command is used to convert the collection of resultant frames to an avi file.

4.6 Applying Morphological Operations:

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size.

To improve the quality of the detected objects and remove the false detection, we can apply morphological operations.

The basic operations are dilation and erosion, dilation adds pixels to the boundary of an object based on the structure element and erosion Removes pixels from the boundary of an object. Often these operations are applied in succession, with same structure element or different element.

For example, Opening of an image is erosion of an image by the structure element followed by dilation of the image by the same structure element. The vice versa of opening i.e dilation followed by erosion is called Closing.

$$A \circ B = (A \ominus B) \oplus B \qquad (10)$$

Opening of A by B

$$A \bullet B = (A \oplus B) \ominus B \qquad (11)$$

Closing of A by B

In practical image-processing applications, dilation and erosion are used most often in various combinations. An image will undergo a series of dilations and/or erosions using the same, or sometimes different, structuring elements. Imdilate, imerode, imopen and imclose matlab functions are used to implements these operations in this part.

5. Comparison of Results:

(a) (b)

Fig 4: a) Sample Image. (b) Result after Detecting moving object

I applied the proposed algorithm to videos collected from the internet as well videos taken by myself. The results were satisfying and the moving objects were always detected. There are false detections, for example if we see fig. , the algorithm shouldn't have detected the tree but since the leaves of the trees are moving continuously the algorithm fails and would have required some second stage of evaluation.

Here are a few more sample results from the video.

(a) (b)

(c) (d)

Fig 5: (a) & (c) Sample Image. (b) & (d) Result after Detecting moving object.

The above fig 5 (a) and (b) shows the detected car (b), this video is taken from the internet. (c) and (d) are taken from a video shot by myself, the object in (d) is me moving in the lab where the background is glass wall and has lot of background motion which is mostly not detected.

6. CONCLUSION

In this paper, I've compared the existing techniques and presented an approach which is robust and has non-parametric background model and background subtraction mechanism that works with color imagery. Each pixel is compared to a set of functions to determine if the pixel is background or foreground. This technique has proven to be very robust; it does an excellent job of separating out objects and reducing background noise but it does have its own quirks, for example if there is sudden change in illumination in the video, it lets the entire background to become the foreground but this disappears based on the learning rate, the faster the learning rate, the less errors there are but it increases the number of computations. Since we are not using any color information it can be extended to any color space i.e RGB or YCrCb or MYK or any other color space, this also has a disadvantage; that is we cannot use this algorithm to detect something of a particular intensity, for example we won't be able to separate out the shadow of an object with this method.

7. Future work

This algorithm can be extended to work with color images by applying the Gaussian model for the different RGB components individually and combining the results in to a single video. This color information will enable us to detect shadows and any small change in illumination. We can use this information to prevent the algorithm to detect the shadows and give better results.

8. ACKNOWLEDGMENTS

This work was done under the guidance of Professor Roger S. Gaborski for the course Introduction to Computer Vision.

9. REFERENCES

[1] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in Proceedings of IEEE ICCV'99 Frame-rate workshop, Sept 1999.

[2] IA. Mittal and D. Huttenlocher, "Scene modeling for wide area surveillancd and image synthesis," in Proceedings IEEE conference on computer vision and pattern recognition, 2, pp. 160{167, (Hilton Head Isand, SC), June 2000.

[3] T. Bouwmans, F. El Baf, B. Vachon, "Background Modeling using Mixture of Gaussians for Foreground Detection", "Recent Patents on Computer Science 1, 3 (2008) 219-237".

[4] Isaac Cohen, G´erard Medioni, "Detecting and Tracking Moving Objects for Video Surveillance", IEEE Proc. Computer Vision and Pattern Recognition, Jun. 23-25, 1999. Fort Collins CO.

[5] Prithviraj Banerjee and Somnath Sengupta, "Human Motion Detection and Tracking for Video Surveillance", IEEE.

[6] FR. Cucchiara, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," IEEE Transactions on Pattern Analysis and Machine Intelligence 25, pp. 1337{1342, Oct 2003.röhlich, B. and Plate, J. 2000.

[7] TK.-P. Karmann, A. V. Brandt, and R. Gerl, \Moving object segmentation based on adabtive reference images," in Signal Processing V: Theories and Application, Elsevier Science Publishers B.V., 1990.avel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

10. Author Biography:

Ravdeep Johar was born in Bokaro (J.H.) in India on July 16, 1991. He completed his B.Tech.in Computer Science Engineering from SRM University in 2012. Previously, he was a visiting student at the University of Wisconsin-Milwaukee, USA in 2011.He is currently pursuing Master of Science in Computer Science at Rochester Institute of Technology, and his fields of interest include Computer Graphics, Computer Vision and Programming Languages.