Parameter Analysis for Mixture of Gaussians Model

Qi Zang and Reinhard Klette
Department of Computer Science, Tamaki Campus, The University of Auckland
Auckland, New Zealand
Email: qzan001@ec.auckland.ac.nz

Abstract

Background subtraction is one of the main techniques to extract moving objects from background scenes. A mixture of Gaussians is a common model for background subtraction. There are several parameters involved in such a model. Obviously, the assignment of initial values to these parameters affects the accuracy of background subtraction. In this paper, we analyze in detail the impact of different initial parameter values based on our model implementation. Both indoor and outdoor video sequences have been tested. This parameter value analysis provides suggestions how to choose suitable initial parameter values, assign reasonable thresholds which ensure better results, while using a mixture of Gaussians model in video surveillance applications.

Keywords: mixture of Gaussians model, parameter analysis, video surveillance

1 Introduction

The mixture of Gaussians model (MOGS) became increasingly popular in image sequence analysis due to its robustness and stability [3][4]:

1. *MOGS characterize static scenes.*
   A common example is the paper [9] by Stauffer and Grimson which models each background pixel's distribution using a mixture of Gaussians model; this model allowed (for example) to monitor continuously a university campus. It learns patterns of activities at a given site, then monitors and classifies activities based on these learned patterns. The system provides statistical descriptions of typical activity patterns despite of rainy, snowy, or sunny weather.

2. *MOGS characterize object colors or object trajectories.*
   For examples of applications of mixture of Gaussians model for modelling object colors or tracking of a moving object, see papers [6, 7] by Raja et al. Gaussians mixture models were used to estimate probability densities of the color of human skin, clothing, and background. These models were used to detect, track, and segment people, faces, or hands [8].

Further mixture of Gaussians model applications are to model noise distributions or shaded areas [2]. Paper [2] presents a method for detecting moving object shadows against a static background scene using a Gaussian shadow model. The chosen shadow model is parameterized with several features including the orientation, mean and center position of a shadow region. Using a mixture of Gaussians model to characterize moving objects also allows to deal with partial occlusions (but often in a time-consuming way).

In this paper we use a mixture of Gaussians model for modelling static background scenes. We present our results of implementing a mixture of Gaussians model based on both indoor and outdoor video sequences. A detailed analysis of assigning different values to the parameters in a mixture of Gaussians model is presented. These experimental results provide some guidelines for the selection of different parameter values.

2 Related work

An important property of Gaussian distributions is that they still remain Gaussian distributions after any linear transformation. This property is one of the reasons that the Gaussian models are very commonly used for solving estimation problems [1]. Gaussian models are widely used in adaptive systems. Especially in video surveillance applications, normally a Gaussian distribution is assumed in order to make the system adaptive to uncontrolled changes like in illumination, outdoor weather, color changes, and so on.

A Gaussian mixture is a *pdf* (i.e., point distribution function) consisting of a weighted sum of Gaussian densities [1]. The Gaussian mixture model belongs to a class of density models which combine several functions as additive components.

Let $\mathbf{X}_t$ be the variable which represents the current pixel in frame $\mathbf{I}_t$, K is the number of distributions, and t represents time (i.e., the frame index), $\omega_{i,t}$ is an estimate of the weight of the ith Gaussian in the mixture at time t, η is a Gaussian probability density function, $\mu_{i,t}$

is the mean value of the ith Gaussian in the mixture at time t, $\Sigma_{i,t}$ is the covariance matrix of the ith Gaussian in the mixture at time t. These functions are combined together to provide a combined density function, which can be employed, for example, to model colors of a dynamic scene or object. Probabilities are computed for each color pixel while a model is constructed.

A Gaussian mixture model can be formulated in general as follows:

$$P(\mathbf{X}_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(\mathbf{X}_t; \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

where, obviously,

$$\sum_{i=1}^{K} \omega_{i,t} = 1 \quad (2)$$

The mean of such a mixture equals

$$\mu_t = \sum_{i=1}^{K} \omega_{i,t} \mu_{i,t} \quad (3)$$

that is, the weighted sum of the means of the component densities.

For example, papers [5, 6, 7] are all based on using the Gaussian mixture model. In [7], a number of Gaussian functions are taken as an approximation of a multimodel distribution in color space, and conditional probabilities are computed for all color pixels, probability densities are estimated from the background colors, and peoples' clothing, heads, hands, and so forth. Two assumptions are made, one is that a person of interest in an image will form a spatially contiguous region in the image plane. Another is that the set of colors for either the person or the background are relatively distinct, the pixels belonging to the person may be treated as a statistical distribution in the image plane.

An adaptive technique based on the Gaussian mixture model is discussed in [9] for the tracker module of a video surveillance system. This technique is to model each background pixel as a mixture of Gaussians. The Gaussians are evaluated using a simple heuristics to hypothesize which are most likely to be part of the "background process". Each pixel is modelled by a mixture of K Gaussians as stated in Equation (1), where K is the number of distributions. Normally, K equals 3, 4 or 5 in practice. Every new pixel value $\mathbf{X}_t$ is checked against the existing K Gaussian distributions until a match is found. Based on the matching results, the background is updated as follows:

$\mathbf{X}_t$ *matches* component i if $\mathbf{X}_t$ is within 2.5 standard deviation of this distribution (multiple matches are possible); in case of such a match, the parameters of the ith component are updated as follows:

$$\begin{aligned} \omega_{i,t} &= (1-\alpha)\omega_{i,t-1} + \alpha & (4)\\ \mu_{i,t} &= (1-\rho)\mu_{i,t-1} + \rho \mathbf{X}_t & (5)\\ \sigma_{i,t}^2 &= (1-\rho)\sigma_{i,t-1}^2 & (6)\\ &\quad +\rho(\mathbf{X}_t - \mu_{i,t})^{\mathsf{T}}(\mathbf{X}_t - \mu_{i,t}) \end{aligned}$$

where $\rho = \alpha \mathrm{P}(\mathbf{X}_t | \mu_{i,t-1}, \Sigma_{i,t-1})$. α is the predefined learning parameter, $\sigma_{i,t}^2$ is the variance of the ith Gaussian in the mixture at time t, μ_t is the mean of the pixel at time t, $\mathbf{X}_t$ is (as above) the recent pixel at time t.

The parameters for all the unmatched distributions remain unchanged, what means that

$$\begin{aligned} \mu_{i,t} &= \mu_{i,t-1} \quad \text{and} & (7)\\ \sigma_{i,t}^2 &= \sigma_{i,t-1}^2 & (8) \end{aligned}$$

But the corresponding weights $\omega_{i,t}$ need to be adjusted using the formula:

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} \quad (9)$$

If $\mathbf{X}_t$ matches none of the K distributions, then the least probable distribution (i.e., the distribution with the lowest weight) is replaced by a distribution where the current value acts as its mean value, the variance is chosen to be "high" and the a-priori weight is "low" [9].

The background estimation problem is solved by specifying the Gaussian distributions, which have the most supporting evidence and the least variance. Because the moving object has larger variance than a background pixel, so in order to represent background processes, first the Gaussians are ordered by the value of $\omega_{i,t}/\|\Sigma_{i,t}\|$ in decreasing order. The background distribution stays on top with the lowest variance by applying a threshold T, where

$$B = \mathrm{argmin}_b \left(\frac{\sum_{i=1}^{b} \omega_{i,t}}{\sum_{i=1}^{K} \omega_{i,t}} > T \right) \quad (10)$$

(Note that the denominator is supposed to be equal to 1 in case of proper normalization.) All pixels $\mathbf{X}_t$ which do not match any of these components will be marked as foreground.

3 Analysis of parameter values

Threshold T is to define the fraction between background distribution and foreground distribution. This value is based on the background scene and the number of components in the Gaussian mixture model. We can obtain it from a testing procedure before starting the real application system. A small value of T (say, $T = 0.1$) will lead to a situation, in which not all background distribution is covered; a large T value (say, $T = 0.9$) will lead to a situation in which the foreground distribution is "merging" with the background distribution. The T value we used in our program equals 0.79. We will analyze other parameter values in the following.

3.1 Number of components

K denotes the number of components in a Gaussian mixture model. For simple indoor scenes, a small value of K is sufficient, perhaps $K = 2$; for outdoor complex scenes, a larger K is needed, usually 3, 4, or 5.

Figure 1 presents our indoor testing results without removing noise. The values we assigned to K are from 1 to 5. Figure 1 illustrates our general experience that adding more components in a Gaussian mixture model does not help in improving the quality of the extracted foreground region. On the contrary, the quality of the extracted foreground region even decreased for $K > 1$. This is because although more components can model more distributions, indoor simple scenes are often not characterized by complex changes, and updating components of the model causes more noise. Figure 1 illustrates that $K = 1$ or $K = 2$ appears here to be the best choice.

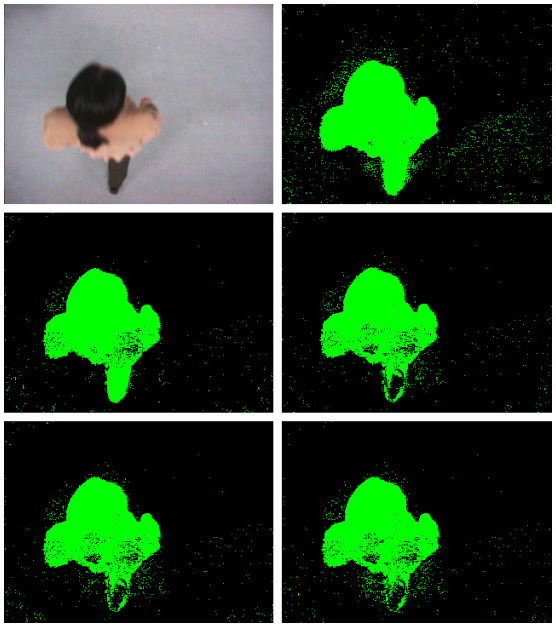

Figure 1: Top left: an original image of a captured sequence. Top right: result for $K = 1$. Middle left: $K = 2$. Middle right: $K = 3$. Bottom left: $K = 4$. Bottom right: $K = 5$.

In complex outdoor scenes, assigning $K = 1$ or $K = 2$ is typically insufficient. For example, we also tested on a winter traffic sequence (uncommon to Auckland) which involves bad weather, snow, and wind. In order to control the movement of snow, waving leaves, and so forth, we defined pixels with values within 4 times standard deviation to be background. K is set to 3. Figure 2 illustrates that, although most small movement of tree leaves and snow are controlled, foreground regions of walking people are missing. The extracted foreground regions are not clear, because vehicles are not running as fast as they normally would on a highway without

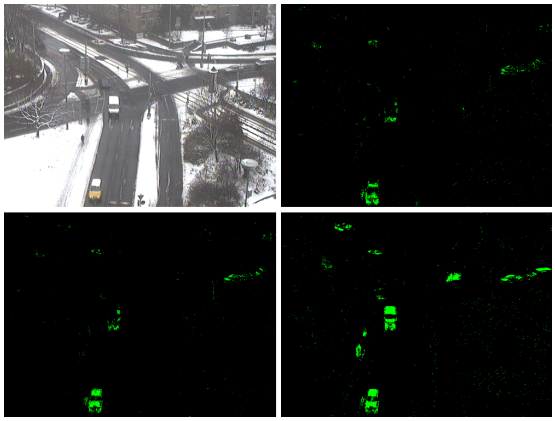

Figure 2: Top left: an original image of the sequence. Top right: result for $K = 3$. Bottom left: $K = 4$. Bottom right: $K = 5$.

snow. We increased the value of K to 4 and 5. The quality of the extracted regions improved.

3.2 Learning rate α

There are two learning rates defined in [9]: one is the predefined learning rate α, the other is the calculated learning rate ρ. ρ is used as a second filter in [9]. As we already summarized in [10], using ρ as a second learning rate is not helpful. We tried using ρ with a very small value, say, less than 10^{-5}. The increase in computation time is costly. In general, assume that the computation time of using one learning rate α is m seconds; then the computation time of using two learning rates α and ρ was greater than $2m$ seconds.

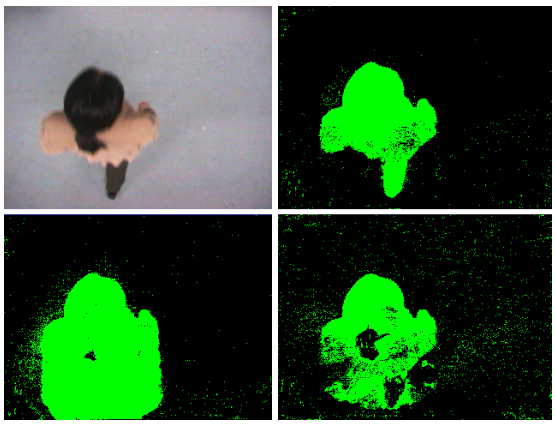

Figure 3: Top left: an original image of the sequence. Top right: result for $\alpha = 0.1$. Bottom left: $\alpha = 0.01$. Bottom right: $\alpha = 0.5$.

In conclusion, we used one learning rate α only. How to assign a reasonable value to α will depend on the given background scenery. A slowly changing background scene needs a small learning rate, a fast chang-

ing background scene needs a larger learning rate. The value α can be obtained from a testing sequence. Here we present an example of using different α values for indoor testing data, see Figure 3. The results in Figure 3 are background estimation before removing noise. Figure 3 illustrates that using value $\alpha = 0.1$ is the best choice for the illustrated cases.

3.3 Assigning initial values

There is an initialization procedure when starting the surveillance system. Assigning different initial values in this procedure will affect the extraction of foreground regions. There are two values that need initial consideration: mean and standard deviation. We will discuss them separately.

Regarding the mean value, from our testing sequences we conclude that assigning either a very large value or a very small value can be considered to be of benefit. Figure 4 shows test results without removing noise. Increasing the mean value from zero to 50 does not impact the extraction of the walking person (as foreground region) very much, and this was experienced for various scenes. In the shown example, the result improved for value 100, but this is not standard for complex backgrounds, and results often were less satisfactory for mean around 100, compared to means below 50. (There are possibilities that the foreground region will be misclassified as the background region.) Large mean values, such as 355 or -999, also proved to be more robust.

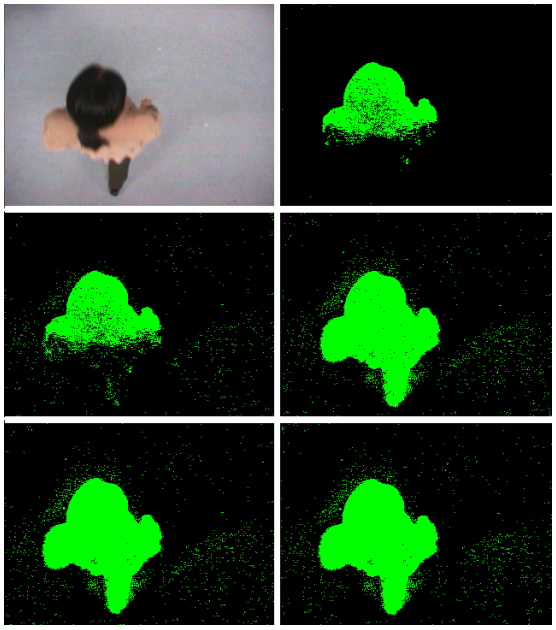

Figure 4: Top left: an original image of the sequence. Top right: result for mean $= 0$. Middle left: mean $= 50$. Middle right: mean $= 100$. Bottom left: mean $= 355$. Bottom right: mean $= -999$.

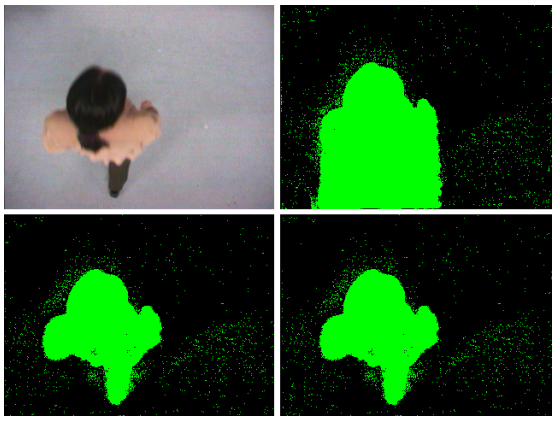

Figure 5: Top left : an original image of the sequence. Top right: result for standard deviation $= 0$. Bottom left: standard deviation $= 100$. Bottom right: standard deviation $= 350$.

In the initialization procedure, we assign in general a very large value to the standard deviation based on our experiments. Figure 5 shows testing results again for the standard sequence used in this paper (without removing noise): for standard deviation equals zero (as an extreme value), many background pixels are misclassified as foreground region even for this simple background. Standard deviation values between 100 and 350 are recommended. In general, using a small value of the standard deviation causes that background pixels are too often classified as foreground distribution.

There are other options to assign a value to the standard deviation. The least probable distribution will be replaced if the current pixel does not match with any of the existing distributions. The mean value will be replaced using the current pixel value. The standard deviation value needs to be large. Figure 6 shows test results without removing noise. If assigning the standard deviation value to 2, then almost the whole scene is classified as being foreground. This is because pixels with lower values of the standard deviation will be easily classified into the foreground distribution. The middle row of Figure 6 are results of assigning standard deviation values to 12 and 42, respectively. The extracted foreground regions improve in these cases. If assigning standard deviation values between 112 and 212, then part of the foreground region pixels are misclassified as background. This is because the newly appearing pixels will be misclassified in the distribution which has a high variance, taking too long to update the variance value to its real value. Distributions with high weighting values tend to be classified as background.

4 Conclusions

The Gaussian mixture models are a type of density models which are composed of a number of

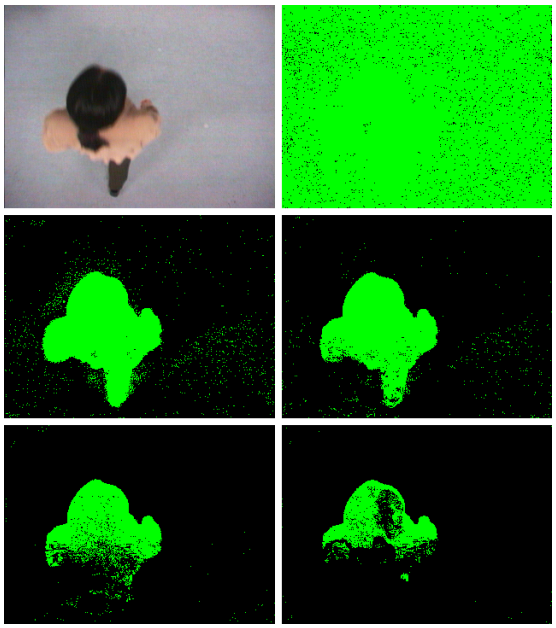

Figure 6: Top left: an original image of the sequence. Top right: result for standard deviation $= 2$. Middle left: standard deviation $= 12$. Middle right: standard deviation $= 42$. Bottom left: standard deviation $= 112$. Bottom right: standard deviation $= 212$.

components (functions). These functions can be used to model the colors of objects or backgrounds in a scene. This allows color-based object tracking and background segmentation. Adaptive Gaussian distributions are applicable for modelling changes, especially when related to fast moving objects such as vehicles on a highway.

The usage of Gaussian distributions has to be based on the application context. It can provide analysis results for long duration scenes (e.g., a surveillance system that monitors a car park or a campus day and night). It is also quite suitable for complex scenes or multi-colored objects. For outdoor scenes, different weather is taken into account. The Gaussian mixture model allows us to adapt to weather changes, such as from rain to snow, from cloudy to sunny, and so forth. Small movements in scenes like waving trees can also be handled. For simple indoor scenes or objects which appear to be monocolored, a small number of components in a Gaussian mixture model is suggested, say one or two components. For outdoor complex scenes, a larger number of components in a Gaussian mixture model is suggested, say starting with 3, but not extending 5 (very much). The maximum number is important if care has to be taken about computation time and system efficiency. In general, more components do have the potential for further improvement.

Of course, how to assign suitable values to parameters during an initialization period will also depend on specific applications. Values of parameters and other suitable initial values can be obtained during a pre-testing procedure. The higher the number of components of a mixture model, the better the results for a complex scene, but the computation time increases. Assigning a very small value to the learning rate will avoid that a slowly moving and large object melts into the background, but will affect the system's adaptation. One needs to balance out all these conditions according to different applications and environments.

References

[1] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Boston, 1993.

[2] C. J. Chang, W. F. Hu, J. W. Hsieh, and Y. S. Chen. Shadow elimination for effective moving object detection with Gaussian models. In Proc. *Int. Conf. Pattern Recognition*, **2**: 540–543, 2002.

[3] S. S. Cheung and C. Kamath: Robust techniques for background subtraction in urban traffic video. In Proc. *Electronic Imaging: Visual Comm. Image Proc.*, 881–892, 2004.

[4] D. S. Lee: Effective Gaussian mixture learning for video background subtraction. *IEEE Trans. Pattern Analysis Machine Intelligence*, **27(5)**: 827–832, 2005.

[5] S. J. McKenna, Y. Raja, and S. Gong. Object tracking using adaptive color mixture models. In Proc. *Asian Conf. Computer Vision*, 615–622, 1998.

[6] Y. Raja, S. J. McKenna, and S. Gong. Tracking color objects using adaptive mixture models. In Proc. *Image Vision Computing*, **17**: 225–231, 1999.

[7] Y. Raja, S. J. McKenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using color. In Proc. *IEEE Int. Conf. Automatic Face Gesture Recognition*, 228–233, 1998.

[8] K. She, G. Bebis, H. Gu, and R. Miller: Vehicle tracking using on-line fusion of color and shape features. In Proc. *IEEE Int. Conf. Intelligent Transportation Systems*, **16**: 731–736, 2004.

[9] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In Proc. *Computer Vision and Pattern Recognition*, **2**: 246–252, 1999.

[10] Q. Zang and R. Klette. Evaluation of an adaptive composite Gaussian model in video surveillance. In Proc. *Image Vision Computing New Zealand*, 243–248, 2002.