**Foreground Modeling by Modeling Color Distribution**

**Santosh Hiremath**
**S0787216**
**University of Edinburgh**
**AV Practical 1**

## Introduction

Automatic video surveillance of a scene mainly consists of two parts, object detection and object tracking. This is done with generally two methods, one is with the assumption that the background remains more or less constant, subsequent frames from a video camera are compared for changes in the scene. If an object has moved during that time period then it will be noticeable in the series of frames. This technique is called background modeling where an adaptive model of the background is used to compare subsequent image frames. This forms the first step in surveillance. Details about different background modeling algorithms can be found in [1, 2, 3]. Once the object is detected it is tracked using a technique called foreground modeling. Color distribution is one of the more popular techniques used for foreground modeling and segmentation as it is robust to occlusion and deformation of objects. This is because the object's color does not tend to change irrespective of the outside condition.

Both parametric and non parametric methods are used to represent the color distribution. Some of the parametric technique includes representing the pixel intensity as a single Gaussian [4] or as a mixture of Gaussian [5].

This tutorial mainly focuses on two methods

**1. Foreground segmentation using Non Parametric Kernel Density Estimation (KDE).**

**2. Color modeling using Gaussian Mixture Models.**

A short tutorial on KDE can be found in [6, 7]. Similarly, you can read in brief about Gaussian Mixture Models in [12]. A detailed account of both the concepts can be found in [8, 13].

## Non Parametric Kernel Density Estimation (KDE)

(Details of the algorithm can be found in [3])

### 1.2. Color Density Estimation

Consider a sample $S = \{x_i\}_{i=1,\dots,N}$ of an image region where $x_i$ is a $d$ dimensional vector representing the color, the density of an arbitrary point $y$ in the color space can be given by

$$P(y) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d} K_{\sigma j}\left(y_j - x_{ij}\right) \qquad (1)$$

Where $K_{\sigma j}$ is a Gaussian kernel function of $j^{th}$ dimension of the color space with the bandwidth $\sigma_j$. The color space required to represent the image is chosen depending on the requirement. If the model is to be invariant to illumination then 2D color space is used eg a, b from LAB space. 3D color space is used when illumination information is important. (A quick tutorial on color spaces can be found in [9, 10]).

### 1.3. Body part segmentation using Color Model

The method assumes that people are in the upright position such that the body can be segmented into vertically aligned blobs corresponding to head, torso and bottom. Mathematically this is represented as $M = \{A_i\}$ where $A_i$ is a blob representing a major color region of the body along the vertical axis. It is assumed the people's body can be segmented into different color regions because of the way they dress up generally (shirt or trouser etc). A blob is associated with a color distribution $c$ and also has a spatial distribution $(x, y)$ based on its position with respect to the body. The probability of a color $c$ appearing at a location $(x, y)$ is given by

$$P_A(x,y,c) = f_A(x)\,g_A(y)\,h_A(c) \qquad (2)$$

where $g_a(y)$ and $f_a(x)$ are the density of vertical and horizontal position of the blob and $h_A(c)$ is the color density. The pixel color is represented using a 3D color vector $X = (r, g, s)$ where $r$ *=R/(R+G+B)* and g *=G/(R+G+B)* are chromaticity variables and $s$ = *(R+G+B)/3* is the brightness variable. In this case the color density function of the pixel takes the form

$$h_A(r,g,s) = \frac{1}{N}\sum_{i=1}^{N} K_{\sigma_r}(r-r_i)\,K_{\sigma g}(g-g_i)\,K_{\sigma_s}(s-s_i) \qquad (3)$$

Given the above color distribution and the position of the blob we can classify the pixel into one of the three blobs using the maximum likelihood classification. Once the pixels are classified, consecutive blobs are segmented by finding a line between them that minimizes the classification error [11]. This color model is made adaptive by performing blob segmentation at every frame (while the individual is in isolated) and by padding information from new frames and the old ones at the same time. At this stage we have the color model for the person being tracked. Given this model the person can be tracked even when in a group. This is achieved by hypothesis testing where the probability of the $c$ appearing in the position $x, y$ given a person model $M = \{A_i\} i = 1..n$ is can be given by

$$P_A\left(x, y, c \mid M\right) = \frac{f_A\left(x - x_0\right)}{C\left(y - y_0\right)} \sum_{i=1}^{n} g_{Ai}\left(y - y_0\right) h_{A_i}\left(c\right)$$

where $C$ is the normalization constant and the values of M $x_o$ and $y_o$ are the origins with respect to which $x, y$ are measured respectively. The figure 2 shows the example of blob segmentation.

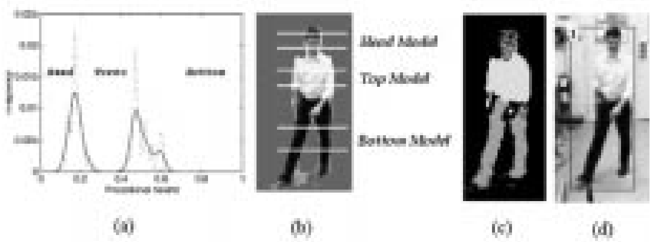


**Figure 1(Courtesy [3]):** (a) Blob separator histogram. (b) Confidence bands. (c) Blob segmentation. (d) Detected blob separators.

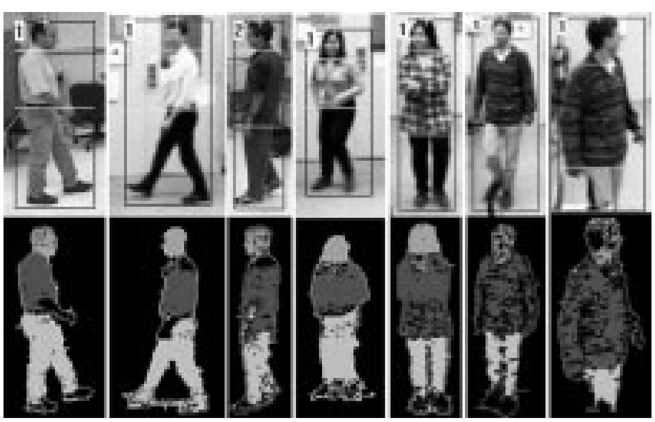

**Figure 2(Courtesy [3]): Example of blob segmentation**

## 2. Color modeling using Gaussian Mixture Models

(Details of the algorithm can be found in [14])

This algorithm uses the HSI color space for representing color of the object. This is to eliminate the intensity component to make the algorithm robust against illumination. In case of Gaussian mixture models, given a sample pixel **x** the probability density that it belongs to an object $O$, can be modeled as a mixture of $m$ Gaussians given by the equation

$$p\left(\mathrm{x} \mid O\right) = \sum_{j=1}^{m} p\left(\mathrm{x} \mid j\right) \pi(j) \tag{4}$$

Here $\pi(j)$ is the prior probability that the pixel **x** was generated from the $j^{th}$ component and is known as the mixture parameter. EM is applied on this mixture model to fit them to data. EM algorithm provides a iterative method of increasing the likelihood of the model until a local optimum is reached. The algorithm is sensitive to number of mixtures and its parameters. Initial values for these are obtained by visual inspection of object's color distribution. However this can be a bottleneck at times as the object required to be tracked will not be available for visual inspection. In this regard Non parametric density methods have an advantage over parametric estimators. The model defined above forms the initial model. This model is made invariant to small changes in the scene by adapting the parameter of the mixture within the model. While the parameters are adapted, the number of mixture components remains constant as it is assumed that object's color does not alter significantly, hence the number of components obtained during initialization process is optimal. The adaptation works as follows. Assume that the each initial mixture has a parameter set $\left(\mu_o, \Sigma_o, \pi_o\right)$, where $\mu_o$, is the mean, $\Sigma_o$ is the covariance of the component and $\pi_o$ is the prior probability that the pixel was generated by the component. At each new frame the parameters of each mixture components are estimated using only the new data [14] as follows:

$$\mu^{(t)} = \frac{\sum P(j \mid \mathrm{x})\mathrm{x}}{\psi^{(t)}} \tag{5}$$

$$\pi^{(t)} = \frac{\psi^{(t)}}{N^{(t)}} \tag{6}$$

$$\Sigma^{(t)} = \frac{\sum P(j \mid \mathrm{x})(\mathrm{x} - \mu_{t-1})^T (\mathrm{x} - \mu_{t-1})}{\psi^{(t)}} \tag{7}$$

Where $\psi^{(t)}$ is the sum of the posterior probabilities of the $N^{(t)}$ number of pixels at time $t$. The mixture parameters are then updated using the weighted sum of

previous estimates. The details of the derivation can be found in [14].
The figure below shows the result of tracking using the above algorithm.

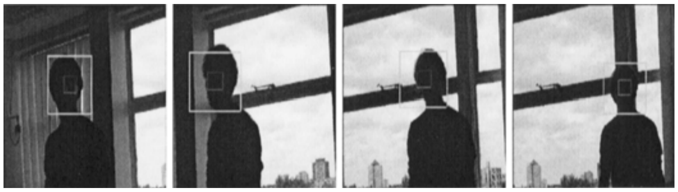

**Figure 4 (Courtesy [4])**. The sequence depicted in Fig. 2 tracked with an adaptive color model

**References**


1.http://homepages.inf.ed.ac.uk/cgi/rbf/CVONLINE/entries.pl?TAG661 - 30/01/2008.

2.T. Horprasert. D. Harwood and L. S. Davis, “A statistical approach for real-time robust background subtraction and shadow detection,”
IEEE Frame-Rate ApplicationsWorkshop, Kerkyra, Greece, 1999.

3.A. Elgammal, D. Harwood, and L. S. Davis, “Nonparametric background model for background subtraction,” in Proc. 6th Eur. Conf.
Computer Vision, vol. 2, 2000, pp. 751–767.

4. C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-time tracking of human body,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 780–785, July 1997.

5. Y. Raja, S. J. Mckenna, and S. Gong, “Tracking color objects using adaptive mixture models,” *Image Vision Comput.*, no. 17, pp.225–231, 1999.

6.http://www.maths.uwa.edu.au/%7Eduongt/seminars/intro2kde/ - 30/01/2008.

7.http://parallel.vub.ac.be/research/causalModels/tutorial/kde.html - 30/01/2008.

8. R. O. Duda, D. G. Stork, and P. E. Hart, *Pattern Classification*. New York: Wiley, 2000.

9. http://www.cambridgeincolour.com/tutorials/color-spaces.htm - 30/01/2008

10. http://en.wikipedia.org/wiki/Color_space – 30/01/2008

11. A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.

12.http://www.neurosci.aist.go.jp/~akaho/MixtureEM.html - 30/01/2008.

13. C.Bishop, Neural Networks for Pattern Recognition, Oxford University Press, New York, 1995.

14. Y. Raja, S. J. Mckenna, and S. Gong, “Tracking color objects using adaptive mixture models,” *Image Vision Comput*, no. 17, pp.225–231, 1999