Colour Model Selection and Adaptation in Dynamic Scenes

Yogesh Raja[1], Stephen J. McKenna[2], and Shaogang Gong[1]

[1] Department of Computer Science, Queen Mary and Westfield College, England
{jpmetal,sgg}@dcs.qmw.ac.uk
[2] Department of Applied Computing, University of Dundee, Scotland
stephen@dcs.qmw.ac.uk

Abstract. We use colour mixture models for real-time colour-based object localisation, tracking and segmentation in dynamic scenes. Within such a framework, we address the issues of model order selection, modelling scene background and model adaptation in time. Experimental results are given to demonstrate our approach in different scale and lighting conditions.

1 Introduction

Colour has been used in machine vision for tasks such as segmentation [1, 2], tracking [3] and recognition [4, 5]. Colour offers many advantages over geometric information in dynamic vision such as robustness under partial occlusion, rotation in depth, scale changes and resolution changes. Furthermore, using colour enables real-time performance on modest hardware platforms [1].

Swain and Ballard [5] described a scheme which used histograms for modelling the colours of an object. The colour space was quantised through the histogram's structure which comprised a number of "bins". An algorithm known as "histogram intersection" was used for matching image histograms with model histograms. Although colour histograms can be used to estimate densities in colour space, the level of quantisation imposed on the colour space influences the resulting density. If the number of bins n is too high, the estimated density will be "noisy" and many bins will be empty. If n is too low, density structure is smoothed away. Histograms are effective only when n can be kept relatively low and where sufficient data are available. A potentially more effective semi-parametric [6] approach for colour density estimation is to use Gaussian mixture models. With this approach, a number of Gaussian functions are taken as an approximation to a multi-modal distribution in colour space and conditional probabilities are then computed for colour pixels [1, 3]. Gaussian mixture models can also be viewed as a form of generalised radial basis function network in which each Gaussian component is a basis function or 'hidden' unit. The component priors can be viewed as weights in an output layer. Finite mixture models have also been discussed at length elsewhere [6–12] although most of this work has concentrated on the general studies of the properties of mixture models rather than developing vision models for use with real data from dynamic scenes.

However, the use of colour mixture models in dynamic scenes is not without its difficulties. A common problem associated with density-based modelling of statistical data involves the selection of the number of parameters for a model, known as the *model order selection* problem [6]. With colour mixture models, this involves the selection of the number of Gaussian components. The goal is to generate a model that provides accurate predictions for new data. Too few parameters can lead to a poor model which over-generalises the data (high bias), while too many parameters can result in an overfit of the model to the training data (high variance) [13]. In either case, the underlying distribution responsible for the training data is not reflected accurately and performance on new data will be poor. Existing methods for model selection are usually rather *ad hoc*. An exception is the recursive algorithm of Priebe and Marchette [10]. It was extended to model non-stationary data series through the use of temporal windowing. Their algorithm adds new components dynamically when the mixture model fails to account well for a new data point. The approach in this paper is different in that an iterative algorithm is used to determine model order based on a fixed data set. The mixture model is then adapted on-line by updating components' parameters while keeping the number of components fixed. The assumption made here is that the number of components needed to accurately model an object's colour does not alter significantly with changing viewing conditions.

Methods have been proposed for colour-based detection and tracking of skin-coloured objects (e.g. [1, 14–17]). In particular, a system constructed by Wren *et al.* [18] enabled tracking of entire people in controlled environments with static cameras. Each pixel in an image had an associated feature vector comprising spatial and colour components. These feature vectors were clustered, which led to a collection of "blobs" defined by spatial and spectral similarity. A collection of blobs constituted a representation of a person. This limited tracking to people with homogeneously coloured regions with an unchanging background.

Most colour cameras provide an RGB (red, green, blue) signal. In order to model objects' colour distributions, the RGB signal is first transformed to make the intensity or brightness explicit so that it can be discarded in order to obtain a high level of invariance to the intensity of ambient illumination. Here the HSV (hue, saturation, value) representation was used and colour distributions were modelled in the 2D hue-saturation space. Hue corresponds to our intuitive notion of 'colour' whilst saturation corresponds to our idea of 'vividness' or 'purity' of colour. At low saturation, measurements of hue become unreliable and are discarded. Likewise, pixels with very high intensity are discarded. It should be noted that the HSV system does not relate well to human vision. In particular, the usual definition of intensity as $max(R+G+B)$ is at odds with our perception of intensity. However, this is not important for the tracking application described here. If in other applications it was deemed desirable to relate the colour models to human perception then perceptually-based systems like CIE $L^*u^*v^*$ and CIE $L^*a^*b^*$ should be used instead of HSV.

The main difficulty in modelling colour robustly is the *colour constancy* problem which arises due to variation in colour values brought about by lighting changes. This problem is addressed here by employing colour adaptation.

For the remaining sections of the paper, we first describe briefly in Section 2 colour mixture models in HS-space for dynamic scene segmentation and object tracking. In Section 3, we describe a method for automatic model selection in multi-colour mixture models. Section 4 focuses on the issue of modelling object colour in the context of a given scene. We then introduce a mechanism for model adaptation over time and under changing lighting conditions in Section 6. We discuss some experimental results in Section 7 before drawing conclusions in Section 8.

2 Colour Mixture Models

Colour histograms [5] are a simple non-parametric method for modelling. However, the use of histograms for estimating colour densities is only possible because n can be kept relatively small and because there are many data points (pixels) available. A more effective approach is to use Gaussian mixture models. Let the conditional density for a pixel $\boldsymbol{\xi}$ belonging to a multi-coloured object $\mathcal{O}$ be a mixture with M component densities:

$$p(\boldsymbol{\xi}|\mathcal{O}) = \sum_{j=1}^{M} p(\boldsymbol{\xi}|j)P(j) \tag{1}$$

where a mixing parameter $P(j)$ corresponds to the prior probability that pixel $\boldsymbol{\xi}$ was generated by component j and where $\sum_{j=1}^{M} P(j) = 1$. Each mixture component is a Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, i.e. in the case of a 2D colour space:

$$p(\boldsymbol{\xi}|j) = \frac{1}{2\pi|\boldsymbol{\Sigma}_j|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(\boldsymbol{\xi}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{\xi}-\boldsymbol{\mu}_j)} \tag{2}$$

Expectation-Maximisation (EM) provides an effective maximum-likelihood algorithm for fitting such a mixture to a data set [6, 19]. Figure 1 shows an example of a Gaussian mixture model of a multi-coloured object in HS-space. Outlier points, which can be caused by image noise and specular highlights, have little influence upon the mixture model. Pixels with very low intensity were discarded because the observed hue and saturation became unstable for such pixels. Likewise, pixels with very high intensity were discarded. Once a model has been estimated it can be converted into a look-up table for efficient on-line indexing of colour probabilities.

Given a colour mixture model of an object, the object can then be effectively located and tracked in the scene by computing a probability map for the pixels in the image within a search window. The size and position of the object are then estimated from the resulting distribution in the image plane. Although the HS-space representation permits a degree of robustness against limited brightness

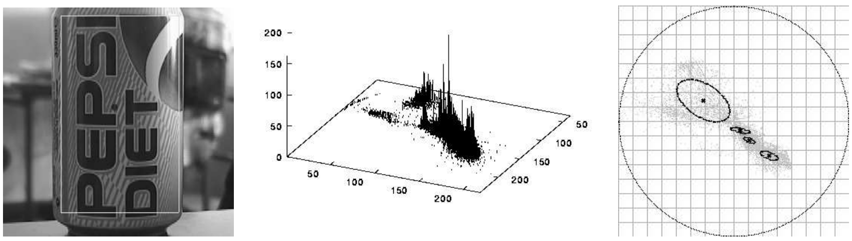

Fig. 1. *Left: a multi-coloured object (a PEPSI can). Centre: its colour histogram in HS-space. It can be noted that such a histogram representation is only viable when a large amount of data is available due to being non-parametric. Right: its Gaussian mixture model. The mixture components are shown as elliptical contours of equal probability.*

change, tracking is only reliable provided that the lighting conditions are relatively stable. Object position at frame t is taken to be the mean $\boldsymbol{m}_t = (m_x, m_y)$ and object size is estimated from the standard deviation $\boldsymbol{\sigma}_t = (\sigma_x, \sigma_y)$. More precisely, for a given frame t, the object position $\boldsymbol{m}_t$ is estimated as an offset from the position $\boldsymbol{m}_{t-1}$:

$$\boldsymbol{m}_t = \boldsymbol{m}_{t-1} + \frac{\sum_{\mathbf{x}} p(\boldsymbol{\xi}_{\mathbf{x}})(\mathbf{x} - \boldsymbol{m}_{t-1})}{\sum_{\mathbf{x}} p(\boldsymbol{\xi}_{\mathbf{x}})}$$

where $\mathbf{x}$ ranges over all image coordinates in the region of interest and $\boldsymbol{\xi}_{\mathbf{x}}$ is the colour point at image position $\mathbf{x}$. To improve accuracy, probabilities $p(\boldsymbol{\xi}_{\mathbf{x}})$ are thresholded. Values lower than the threshold are taken to be background and are consequently set to zero in order to nullify their influence on the estimation of $\boldsymbol{m}_t$ and $\boldsymbol{\sigma}_t$.

The size of the object is estimated by computing the standard deviation of the image probability density:

$$\boldsymbol{\sigma}_t = \sqrt{\frac{\sum_{\mathbf{x}} p(\boldsymbol{\xi}_{\mathbf{x}})\{(\mathbf{x} - \boldsymbol{m}_{t-1}) - \boldsymbol{m}_t\}^2}{\sum_{\mathbf{x}} p(\boldsymbol{\xi}_{\mathbf{x}})}}$$

3 Model Order Selection

Model order selection is the problem of choosing the number of parameters that facilitates the accurate modelling of an underlying distribution for a set of data. In this section, we describe a constructive method for automatic determination of the number of components for a colour mixture model.

A standard technique employed for model training, known as *cross validation*, attempts to find the model order that provides the best trade-off between bias and variance. A number of models of different order are trained by minimising an error function for a training set. These models are then evaluated by computing the error function for an independent *validation set*. The model with the lowest error for the validation set is considered to exhibit the best generalisation and its order is taken to be optimal.

This concept is applied to the generation of mixture models through an iterative scheme of splitting components and monitoring generalisation ability. The available data set is partitioned into disjoint training and validation sets. A mixture model is initialised with a small number of components, typically one. Model order is then adapted by iteratively applying EM and splitting components. The likelihood for the validation set is computed after every iteration, and it is assumed that the optimal model order corresponds to the peak in this likelihood function over time. Here, the techniques for selecting and splitting components are outlined.

3.1 Splitting Components

For each component j, let us define a total responsibility r_j as:

$$r_j = \sum_{\boldsymbol{\xi}} p(j|\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}} \frac{p(\boldsymbol{\xi}|j)P(j)}{\sum_{i=1}^{M} p(\boldsymbol{\xi}|i)P(i)} \tag{3}$$

Then the component k with the lowest total responsibility for the validation set is selected for splitting:

$$k = \arg\min_j (r_j)$$

Once the component k to be split has been selected, two new components with means $\boldsymbol{\mu}_{new1}$ and $\boldsymbol{\mu}_{new2}$, and covariance matrices $\boldsymbol{\Sigma}_{new1}$ and $\boldsymbol{\Sigma}_{new2}$ are computed by:

$$\boldsymbol{\mu}_{new1} = \boldsymbol{\mu}_k + \frac{\lambda_1}{2}\mathbf{u}_1$$

$$\boldsymbol{\mu}_{new2} = \boldsymbol{\mu}_k - \frac{\lambda_1}{2}\mathbf{u}_1$$

$$\boldsymbol{\Sigma}_{new1} = \boldsymbol{\Sigma}_{new2} = \boldsymbol{\Sigma}_k$$

where λ_1 is the largest eigenvalue of the covariance matrix $\boldsymbol{\Sigma}_k$ and $\mathbf{u}_1$ is the corresponding eigenvector.

The prior probabilities for the new components are assigned like so:

$$\pi_{new1} = \pi_{new2} = \frac{\pi_k}{2}$$

3.2 A Constructive Algorithm for Model Order Selection

Let i denote the iteration, M_i the number of components in a model at iteration i and $\mathcal{L}_i$ the likelihood of the validation set with respect to the model at iteration i. The initial number of components M_0 may be set to a low number (here $M_0 = 1$). With a validation set generated for the generalisation test, a constructive algorithm for model order selection is as follows:

1. Apply Expectation-Maximisation for model with M_i components.
2. Compute $\mathcal{L}_i$ for validation set
3. IF $(\mathcal{L}_i < \mathcal{L}_{i-1})$ STOP.
4. Save model
5. Find component j with lowest total responsibility
6. Split component j
7. Restart from step 1 with $M_{i+1} = M_i + 1$ and $i = i + 1$.

The algorithm terminates when a peak is located in the likelihood measurements for the validation set. The application of this algorithm to a data set for a multi-coloured object (the PEPSI can as shown in Figure 1) is illustrated in Figure 2.

4 Modelling Colour in Context: Foreground and Background Models

Thresholding probabilities generated by a foreground model alone is often ineffective due to severe overlap between background and foreground colour distributions. For dealing with multi-coloured objects in dynamic scenes, it is computationally desirable to model the colour distribution of the background scene in addition to the objects to be tracked. Given density estimates for both an object, $\mathcal{O}$, and the background scene, $\mathcal{S}$, the probability that a pixel, $\boldsymbol{\xi}$, belongs to the object is given by the posterior probability $P(\mathcal{O}|\boldsymbol{\xi})$:

$$P(\mathcal{O}|\boldsymbol{\xi}) = \frac{p(\boldsymbol{\xi}|\mathcal{O})P(\mathcal{O})}{p(\boldsymbol{\xi}|\mathcal{O})P(\mathcal{O}) + p(\boldsymbol{\xi}|\mathcal{S})P(\mathcal{S})} \tag{4}$$

The prior probability, $P(\mathcal{O})$, is set to reflect the expected size of the object within the search area of the scene $[P(\mathcal{S}) = 1 - P(\mathcal{O})]$. Pixels can be classified by assigning them to the class with the maximum posterior probability. This minimises the probability of misclassification error in a Bayesian sense. However, it is preferable to use the posterior probabilities directly in order to estimate the spatial extent of the object. Furthermore, the density estimates provide a measure of confidence. Pixels in areas of colour space where both foreground and background likelihoods are low are classified with low confidence.

Modelling foreground and background separately has the practical advantage that the object and scene data can be acquired independently. A single background scene model can subsequently be used with many different objects. This is useful in a virtual studio application, for example, where it enables a single studio model to be subsequently used with many different people.

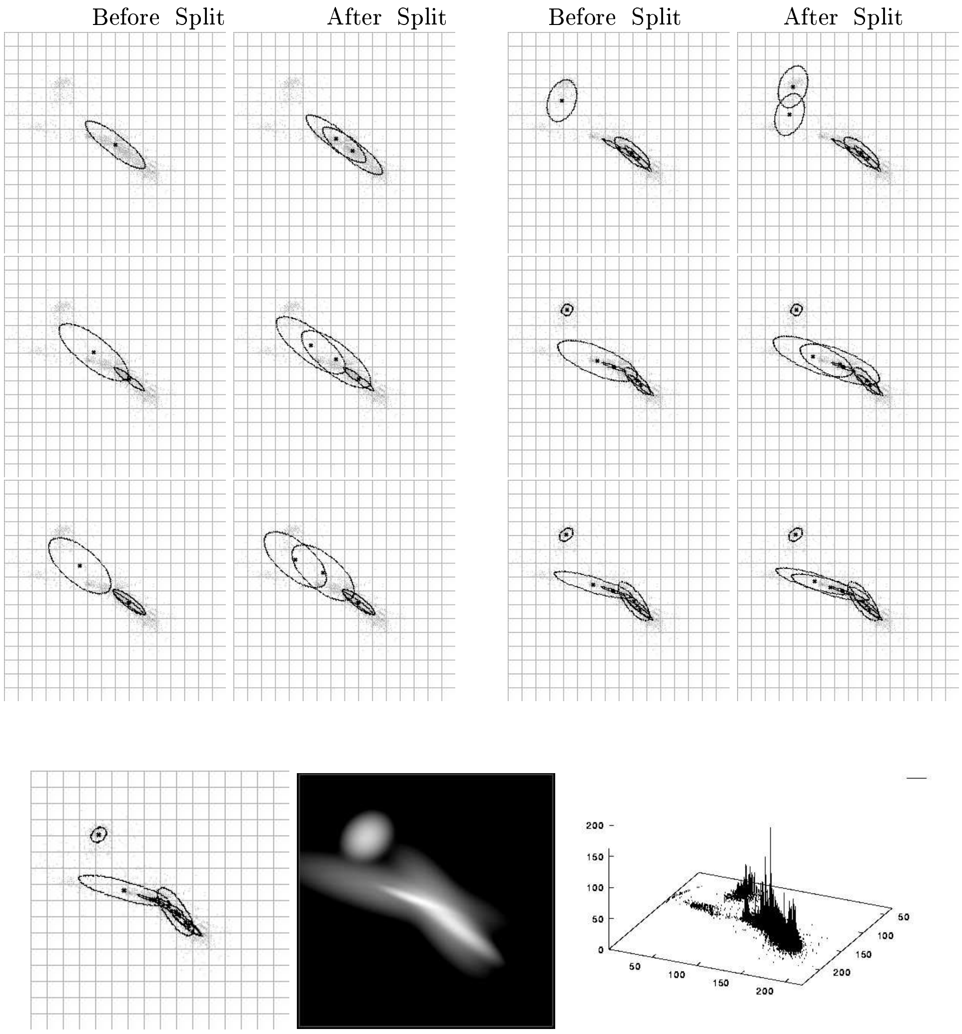

Fig. 2. *Automatic model selection for the colours of a PEPSI can. The top part illustrates the splitting process, with each pair of images showing convergence followed by the splitting of a component. The bottom part shows the final seven component model and the resulting probability density function. Finally, a histogram of the training data is shown.*

5 Modelling Colour in Time: Coping with Change

Colour appearance is often unstable due to changes in both background and foreground lighting. The colour constancy problem has been addressed mainly through the formulation of physics-based models (e.g.[20]). Colour appearance also varies over time due to changes in viewing geometry and changes in camera parameters (auto-iris adjustment). Under the assumption that viewing conditions change gradually over time, statistical colour models can be adapted to reflect the changing colour appearance of a tracked object (or the background scene against which it is tracked). The remainder of this section describes a method for adapting the Gaussian mixture colour models over time.

5.1 Model Adaptation Over Time

At each frame, t, a new set of pixels, $X^{(t)}$, is sampled from the object and used to update the mixture model[1]. These new data sample a slowly varying non-stationary signal. Let $r^{(t)}$ denote the sum of the posterior probabilities of the data in frame t, $r^{(t)} = \sum_{\boldsymbol{\xi} \in X^{(t)}} p(j|\boldsymbol{\xi})$. The parameters are first estimated for each mixture component, j, using only the new data, $X^{(t)}$, from frame t:

$$\boldsymbol{\mu}^{(t)} = \frac{\sum p(j|\boldsymbol{\xi})\boldsymbol{\xi}}{r^{(t)}}, \qquad \pi^{(t)} = \frac{r^{(t)}}{N^{(t)}}$$

$$\boldsymbol{\Sigma}^{(t)} = \frac{\sum p(j|\boldsymbol{\xi})(\boldsymbol{\xi} - \boldsymbol{\mu}_{t-1})^T(\boldsymbol{\xi} - \boldsymbol{\mu}_{t-1})}{r^{(t)}}$$

where $N^{(t)}$ denotes the number of pixels in the new data set and all summations are over $\boldsymbol{\xi} \in X^{(t)}$. The mixture model components then have their parameters updated using weighted sums of the previous recursive estimates, $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \pi_{t-1})$, estimates based on the new data, $(\boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}, \pi^{(t)})$, and estimates based on the old data, $(\boldsymbol{\mu}^{(t-L-1)}, \boldsymbol{\Sigma}^{(t-L-1)}, \pi^{(t-L-1)})$:

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \frac{r^{(t)}}{D_t}(\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}_{t-1}) - \frac{r^{(t-L-1)}}{D_t}(\boldsymbol{\mu}^{(t-L-1)} - \boldsymbol{\mu}_{t-1})$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{t-1} + \frac{r^{(t)}}{D_t}(\boldsymbol{\Sigma}^{(t)} - \boldsymbol{\Sigma}_{t-1}) - \frac{r^{(t-L-1)}}{D_t}(\boldsymbol{\Sigma}^{(t-L-1)} - \boldsymbol{\Sigma}_{t-1})$$

$$\pi_t = \pi_{t-1} + \frac{N^{(t)}}{\sum_{\tau=t-L}^{t} N^{(\tau)}}\left(\pi^{(t)} - \pi_{t-1}\right) - \frac{N^{(t-L-1)}}{\sum_{\tau=t-L}^{t} N^{(\tau)}}\left(\pi^{(t-L-1)} - \pi_{t-1}\right)$$

[1] Throughout this paper, superscript $^{(t)}$ denotes a quantity based only on data from frame t. Subscripts denote recursive estimates.

where $D_t = \sum_{\tau=t-L}^{t} r^{(\tau)}$. The following approximations are used for efficiency:

$$r^{(t-L-1)} \approx \frac{D_{t-1}}{L+1}$$

$$D_t \approx (1 - 1/(L+1))D_{t-1} + r^{(t)}$$

The parameter L controls the adaptivity of the model[2].

During the processing of a sequence, new samples of data for adaptation are gathered from a region of appropriate aspect ratio centred on the estimated object centroid. It is assumed that these data form a representative sample of the objects' colours. This will hold for a large class of objects.

5.2 Selective Adaptation

An obvious problem with adapting a colour model over time is the lack of ground-truth. Any colour-based tracker can lose the object it is tracking due, for example, to occlusion. If such errors go undetected the colour model will adapt to image regions which do not correspond to the object. This is clearly undesirable. In order to help alleviate this problem, observed log-likelihood measurements were used to detect erroneous frames. Colour data from these frames were not used to adapt the object's colour model. In order to boot-strap the tracker for object detection and re-initialisation after a tracking failure, a set of predetermined generic object colour models which perform reasonably in a wide range of illumination conditions are used. Once an object is being tracked, the model adapts and improves tracking performance by becoming specific to the observed conditions.

The adaptive mixture model seeks to maximise the log-likelihood of the colour data over time. The normalised log-likelihood, $\mathcal{L}$, of the data, $X^{(t)}$, observed from the object at time t is given by:

$$\mathcal{L} = \frac{1}{N^{(t)}} \sum_{\boldsymbol{\xi} \in X^{(t)}} \log p(\boldsymbol{\xi}|\mathcal{O})$$

At each time frame, $\mathcal{L}$ is evaluated. If the tracker loses the object there is often a sudden, large drop in the value of $\mathcal{L}$. This provides a way to detect tracker failure. Adaptation is suspended when such an error is detected. The tracker is then re-bootstrapped by increasing the search space to the maximum size. Adaptation is re-activated when the object is again tracked with sufficiently high likelihood.

A temporal median filter was used to compute a threshold, T. Adaptation was only performed when $\mathcal{L} > T$. The median, ν, and standard deviation, σ, of $\mathcal{L}$ were computed for the n most recent above-threshold frames, where $n \leq L$. The threshold was set to $T = \nu - k\sigma$, where k was a constant.

[2] Setting $L = t$ and ignoring terms based on frame $t-L-1$ gives a stochastic algorithm for estimating a Gaussian mixture for a stationary signal [6, 21].

6 Experiments

In the following we describe a set of experiments in which colour mixture models were applied to object segmentation and tracking in dynamic scenes. All the experiments ran in real-time (15-20Hz) on a standard 200MHz PC with a Matrox Meteor board.

6.1 Experiment 1: Colour-based Tracking

Figure 3 shows samples from one continuous sequence where a skin colour mixture model was used on an active camera with pan, tilt and zoom capabilities for tracking a face with occlusion, lighting and scale changes. It is clear that the model copes well with the changes in object appearance. Here the colour mixture model is relatively simple in the sense that the object of interest has almost a uniform colour. Therefore, the number of components can be easily determined.

Fig. 3. *A face is tracked against a cluttered background by an active camera which pans, tilts and zooms.*

6.2 Experiment 2: Modelling Colour in Context

Model selection becomes more difficult with multi-coloured objects. Figure 4 illustrates the multi-coloured object foreground and background models in HS colour space. These resulted from running the constructive algorithm with automatic model selection. A context-dependent object model can be given by a combined posterior density (shown in the bottom right) which defines decision boundaries between object foreground and scene background, even when significant overlap exists between the object and the background.

Figure 5 shows an application of the context-dependent object (person) model in segmentation and tracking. Pixels in the scene were classified as person or background using Equation (4) with the prior probabilities set to $P(\mathcal{S}) = P(\mathcal{O}) = 0.5$. A multi-resolution approach was taken in which segmentation was performed in a coarse-to-fine manner. The segmented object was superimposed

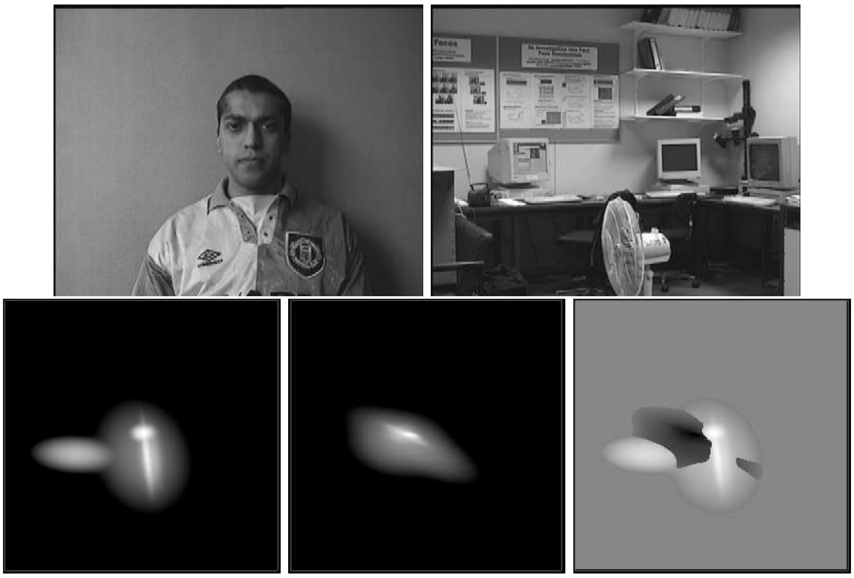

Fig. 4. *Colour mixture models of a multi-coloured object (person model) and the context (scene model). The first row shows the data used to build the foreground (person) and the background (laboratory) models. The second row illustrates the probability density estimated from mixture models for the object foreground and scene background. The rightmost image is the combined posterior density in the HS colour space. Here the "bright" regions represent foreground whilst the "dark" regions give the background. The "grey" areas are regions of uncertainty.*

onto another dynamic scene. Only pixels inside the search area of the tracker were classified. All pixels outside this area were rendered as background. The results are surprisingly good for individual pixel classification alone, but imperfect. However, geometrical models such as PDMs may be integrated into the current setup to exploit the classification results and generate boundary estimates for a more accurate segmentation result.

6.3 Experiment 3: Coping with Change

Results shown in Figures 6 and 7 illustrate the advantage in using an adaptive model. In this sequence the illumination conditions coupled with the camera's auto-iris mechanism resulted in large changes in the apparent colour of the object of interest (the face of a person) as it approached the window. Towards the end of the sequence, the face became very dark, making hue and saturation measurements unreliable. In Figure 6, a non-adaptive model was estimated based on the first image of the sequence only and was used throughout. It was unable to cope with the varying conditions and failure eventually occurred. In Figure 7, the model was allowed to adapt and successfully maintained lock on the face.

Fig. 5. Segmentation results. The top row outlines the tracked region for segmentation and the second row illustrates superimposition onto an alternative sequence.

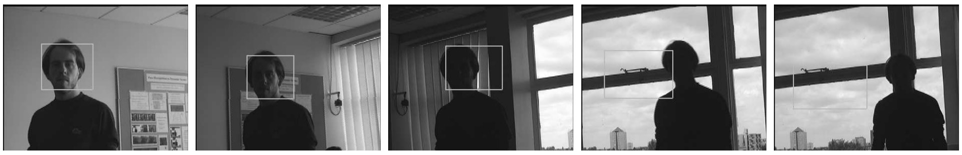

Fig. 6. *Five frames from a sequence in which a face was tracked using a non-adaptive model. The apparent colour of the face changes due to (i) varying illumination and (ii) the camera's auto-iris mechanism which adjusts to the bright exterior light.*

The experiment shown in Figure 8 illustrates the advantage of selective adaptation. The person moved through challenging tracking conditions, before approaching the camera at close range (frames 50-60). Since the camera was placed in the doorway of another room with its own lighting conditions, the person's face underwent a large, sudden and temporary change in apparent colour. When adaptation was performed in every frame, this sudden change had a drastic effect on the model and ultimately led the tracker to fail when the person receded into the corridor. With selective adaptation, these sudden changes were treated as outliers and adaptation was suspended, permitting the tracker to recover.

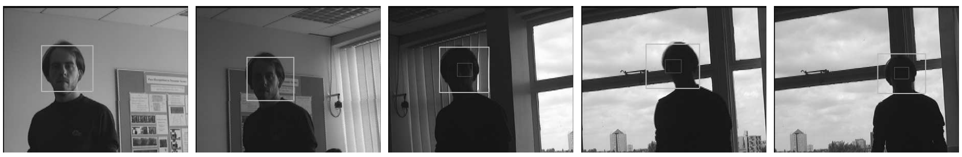

Fig. 7. *The sequence depicted in* Figure 6 *tracked with an adaptive colour model. Here, the model adapts to cope with the change in apparent colour.*

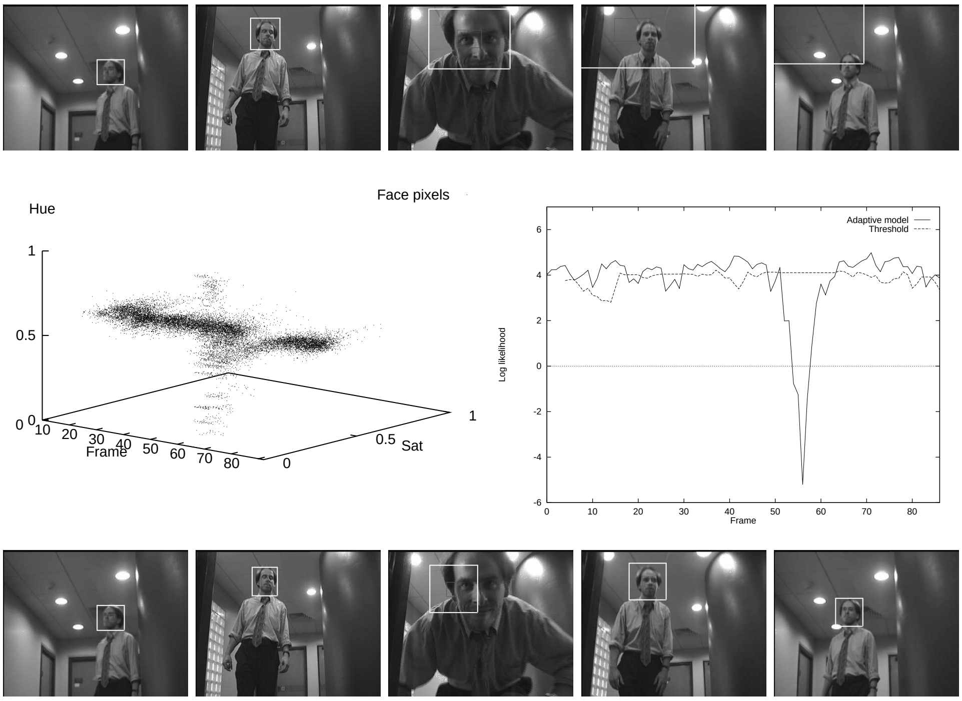

Fig. 8. *At the top are frames 35, 45, 55, 65 and 75 from a sequence. There is strong directional and exterior illumination. The walls have a fleshy tone. At around frame 55, the subject rapidly approaches the camera which is situated in a doorway, resulting in rapid changes in illumination, scale and auto-iris parameters. This can be seen in the 3D plot of the hue-saturation distribution over time. In the top sequence, the model was allowed to adapt in every frame, resulting in failure at around frame 60. The lower sequence illustrates the use of selective adaptation. The right-hand plot shows the normalised log-likelihood measurements and the adaptation threshold .*

7 Conclusion

We have described how the colour distributions of multi-coloured objects can be modelled using mixtures of Gaussians. A constructive algorithm based on maximum-likelihood EM was presented for training a Gaussian mixture model. This algorithm selects an appropriate model order automatically. It avoids the need for initialisation of component means, a procedure which is usually performed using rather *ad hoc* methods. Cross-validation was used to perform early stopping during training.

The colour mixture models were used to perform robust object detection and tracking in real-time using only modest hardware. The use of separate colour models for foreground objects and the background scene was described. Successful tracking was thus performed even when there was significant overlap between object and background colour distributions. Combined foreground and background models were also used for segmentation to facilitate the virtual studio application described. Segmentation quality is insufficient for broadcasting requirements with pixel-wise colour classification. However, it can be exploited to facilitate the application of geometrical models.

The apparent colour of an object varies over time due to changes in illumination, viewing geometry and camera parameters. Rather than attempt to model these changes directly, they were accommodated by allowing the colour models to adapt over time. An algorithm for adapting colour mixture models on-line was presented including a mechanism for detecting tracking errors. This was shown to improve tracking performance under large changes in apparent colour.

We are currently investigating on-line adaptation of model order. This involves both splitting and merging of Gaussian components during tracking. Work is also being done to integrate motion and colour to enable real-time tracking of multiple targets for dynamic gesture recognition. The integration of shape and colour for robust contour tracking and accurate segmentation is also being pursued. This involves the application of Point Distribution Models to colour probabilities to recover good estimates of object boundaries. Colour edges are then used to refine boundary estimates and enable highly accurate segmentation for broadcasting requirements.

References

1. Y. Raja, S. McKenna, and S. Gong, "Segmentation and tracking using colour mixture models," in *Asian Conference on Computer Vision*, Hong Kong, January 1998.
2. W. Skarbek and A. Koschan, "Colour image segmentation - a survey," Tech. Rep., Tech. Univ. of Berlin, 1994.
3. S. McKenna, S. Gong, and Y. Raja, "Face recognition in dynamic scenes," in *BMVC*, 1997.
4. J. Matas, R. Marik, and J. Kittler, "On representation and matching of multi-coloured objects," in *ICCV*, 1995, pp. 726–732.
5. M. J. Swain and D. H. Ballard, "Colour indexing," *IJCV*, pp. 11–32, 1991.
6. C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
7. B. S. Everitt and D. J. Hand, *Finite Mixture Distributions*, Chapman and Hall, New York, 1981.
8. G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*, Marcel Dekker Inc., New York, 1988.
9. C. E. Priebe, "Adaptive mixtures," *J. Amer. Stat. Assoc.*, vol. 89, no. 427, pp. 796–806, 1994.
10. C. E. Priebe and D. J. Marchette, "Adaptive mixtures: Recursive nonparametric pattern recognition," *Pattern Recognition*, vol. 24, no. 12, pp. 1197–1209, 1991.
11. C. E. Priebe and D. J. Marchette, "Adaptive mixture density estimation," *Pattern Recognition*, vol. 26, no. 5, pp. 771–785, 1993.
12. D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*, John Wiley, New York, 1985.
13. S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, 1992.

14. R. Kjeldsen and J. Kender, "Finding skin in color images," in *2nd Int. Conf. on Auto. Face and Gest. Recog.*, 1996.
15. D. Saxe and R. Foulds, "Toward robust skin identification in video images," in *2nd Int. Conf. on Auto. Face and Gest. Recog.*, 1996.
16. B. Schiele and A. Waibel, "Gaze tracking based on face-color," in *IWAFGR*, 1995, pp. 344–349.
17. H. Wu, Q. Chen, and M. Yachida, "An application of fuzzy theory: face detection," in *IWAFGR*, Zurich, June 1995, pp. 314–319.
18. C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder:real-time tracking of the human body," *IEEE PAMI*, vol. 19, no. 7, pp. 780–785, 1997.
19. R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.
20. D. A. Forsyth, *Colour Constancy and its Applications in Machine Vision*, Ph.D. thesis, University of Oxford, 1988.
21. H. G. C. Traven, "A neural network approach to statistical pattern classification by "semiparametric" estimation of probability density functions," *IEEE Trans. Neural Networks*, vol. 2, no. 3, pp. 366–378, 1991.