Robust Fusion of Colour Appearance Models for Object Tracking

Christopher Town
University of Cambridge
Computer Laboratory
Cambridge CB3 0FD, UK

Sean Moran
University of Cambridge
St Catharine's College
Cambridge CB2 1RL, UK

Abstract

This paper reports on work which fuses three different appearance models to enable robust tracking of multiple objects on the basis of colour. Short-term variation in object colour is modelled non-parametrically using adaptive binning histograms. Appearance changes at intermediate time scales are represented by semi-parametric (Gaussian mixture) models while a parametric subspace method (Robust PCA) is employed to model long term stable appearance. Fusion of the three models is achieved through particle filtering and the Democratic integration method. It is shown how robust estimation and adaptation of the models both individually and in combination results in improved visual tracking accuracy.

1 Introduction and Related Work

Appearance models play a vital role in visual tracking of objects. They must remain robust to confounding factors such as noise, occlusions, lighting changes, and background variation while adapting to appearance changes caused by motions and deformations of the tracked entity. This paper shows how robust adaptation and fusion of models based on *global statistics*, *Gaussian mixtures*, and *view-based subspace models* enables a multi-object tracking framework to exploit the advantages of each method.

Global statistics techniques such as colour histograms have been frequently used for object tracking due to their simplicity and versatility, see e.g. [6]. McKenna et al. [5] used Gaussian mixture models (GMM) to model the colour distribution of an object in order to perform tasks such as real-time tracking and segmentation. GMMs were shown to adapt over time to changes in appearance due to factors such as slowly-varying lighting conditions. Furthermore, many computer vision tasks can be posed as problems of learning low dimensional linear or multi-linear models. Principal Component Analysis (PCA) in particular is a popular view-based technique for parameterising shape, appearance, and motion [1].

To overcome the drawbacks of particular methods, approaches which fuse multiple cues by means such as CONDENSATION [11, 6, 7] and Bayesian networks [9, 12] have been gaining prominence. Concurrent probabilistic integration of multiple complementary and redundant cues can greatly increase the robustness of multi-hypothesis tracking.

BMVC 2004 doi:10.5244/C.18.70

2 Object Appearance Modelling

2.1 Background Modelling and Foreground Detection

In order to detect candidate objects (blobs) whose appearance can be modelled and compared to previously tracked objects, we implemented a system based on adaptive background modelling and motion-based foreground detection. The tracking system maintains a background model and foreground motion history (obtained by frame differencing) which are adapted over time using an exponential rate of decay to determine the decreasing influence of previous frames im_{i-1} in the history. The motion history M_i is used to identify a background image bim_i of pixels undergoing sufficiently slow change which can then be used to reliably update the background model B_i and estimate its variance. Pixels are deemed to be part of the dynamic foreground if they exceed a difference threshold which is a multiple of the background variance σ_i^B and if they are not deemed to be part of a shadow as determined by the DNM1 algorithm described in [8].

Foreground pixels are clustered using connected components analysis to identify moving regions (“blobs”). Blob positions are tracked using a Kalman or particle filter with a second order motion model. Tracked objects are matched to detected blobs using a weighted dissimilarity metric which takes into account differences in predicted object location vs blob location and changes in object appearance as modelled by the methods below. Object arrivals, departures and occlusions are inferred using a Bayesian network following the approach of [12].

2.2 Adaptive Binning Colour Histogram

Non-parametric density estimation techniques such as histograms assume no functional form for the underlying distribution and are robust to changes in orientation, relative position and occlusion of objects. Their simplicity and versatility make them suitable for modelling appearance over short time scales and during the initialisation phases of GMM and subspace estimation. The number of histogram bins is usually specified manually and remains fixed. If it is too small then the estimated density is very spiky whereas if it is too large then some of the true structure in the density is smoothed out. In this work, the optimal value for the bin width is determined adaptively following the method of Leow et al [4] who show that the mean error obtained by adaptive binning is about half that of fixed binning.

The optimal number and width of histogram bins is determined by means of k-means clustering with colour differences computed in the CIELAB space using the CIE94 distance d_{kp}. This procedure is repeated n times or until no pixels are left unclustered.

Matching of tracked objects with candidate blobs is performed using weighted correlation. The similarity between two histogram bins is calculated by using a weighted product of the bin counts $H[i]$ and $H[j]$, where the weight w_{ij} is determined from the volume of intersection V_s between the two bins and d is the cluster separation:

$$w_{ij} = \frac{V_s}{V} = \begin{cases} 1 - \frac{3}{4}\frac{d}{R} + \frac{1}{16}(\frac{d}{R})^3 & \text{if } 0 \leq \frac{d}{R} \leq 2 \\ 0 & \text{otherwise} \end{cases} \quad ; \quad w_{ij} \in [0,1] \qquad (1)$$

Dissimilarity of histograms H_p and H_q with n and n' bins respectively is then given by:

$$D_{pq} = 1 - \sum_{i=1}^{n} \sum_{j=1}^{n'} w_{ij} H_p[i] H_q[j]; \; where \; \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} H_p[i] H_p[j] = \sum_{i=1}^{n'} \sum_{j=1}^{n'} w_{ij} H_q[i] H_q[j] = 1 \tag{2}$$

In order to incorporate some longer term appearance variation and smooth over fluctuations, the histograms are adapted using exponential averaging. Given a colour histogram H_t calculated for a blob at frame t and a smoothed object colour histogram S_{t-1} from frame t-1, the new smoothed object colour histogram S_t for frame t is given by $S_t = \alpha H_t + (1-\alpha) S_{t-1}$ where $\alpha = 1 - exp(-\frac{1}{\lambda})$ determines the rate of adaptation. This is set to increase with increasing object speed in order to keep track of rapidly moving objects.

2.3 Gaussian Mixture Model

Gaussian mixture models (GMM) are a type of semi-parametric density estimation. Their use in colour modelling combines advantages of both parametric and non-parametric approaches. Most notably they are not restricted to certain functional forms (as for parametric approaches) and the model only grows with the complexity of the problem and not the size of the data set (as for non-parametric approaches). The conditional density for a pixel ψ belonging to an object O is represented by a mixture of M Gaussians:

$$p(\psi|O) = \sum_{j=1}^{M} P(\psi|j) P(j); \quad \sum_{j=1}^{M} P(j) = 1; \quad 0 \leq P(j) \leq 1 \tag{3}$$

We perform mixture modelling is in Hue-Saturation (HS) space to gain a degree of illumination invariance. Model estimation is performed on blob pixels using subsampling for efficiency and discarding samples whose intensity value is very low or close to saturation. Components are estimated using k-means with priors computed from the proportion of samples in each cluster. The parameters of the Gaussians (mean and covariance) are calculated from the clusters. Model order selection is performed using cross validation on a training and validation set randomly selected from the pixel samples. The training set is used to train a number of models of different order, iteratively applying the EM algorithm and splitting the component j with the lowest responsibility for the validation set as given by

$$r_j = \sum_{\xi} p(j|\xi) = \sum_{\xi} \frac{p(\xi|j)P(j)}{\sum_{i=1}^{M} p(\xi|i)P(i)} \tag{4}$$

Component splitting involves creating two new components from an existing component, and then discarding the existing component. The process terminates once a maximum in the likelihood function is found or the maximum number of iterations has been exceeded.

Adaptation of the GMM over time is performed using the approach suggested in [5]. Given previous recursive estimates (μ_{t-1}, Σ_{t-1}, π_{t-1}), the estimates derived for the new data ($\mu^{(t)}$, $\Sigma^{(t)}$, $\pi^{(t)}$), and estimates based on old data ($\mu^{(t-L-1)}$, $\Sigma^{(t-L-1)}$, $\pi^{(t-L-1)}$), the new mixture parameters for mixture model component j are derived thus:

$$\mu_t = \mu_{t-1} + \frac{r^t}{D_t}(\mu^t - \mu_{t-1}) - \frac{r^{(t-L-1)}}{D_t}(\mu^{(t-L-1)} - \mu_{t-1}) \tag{5}$$

$$\mathbf{\Sigma}_t = \mathbf{\Sigma}_{t-1} + \frac{r^{(t)}}{D_t}(\mathbf{\Sigma}^{(t)} - \mathbf{\Sigma}_{t-1}) - \frac{r^{(t-L-1)}}{D_t}(\mathbf{\Sigma}^{(t-L-1)} - \mathbf{\Sigma}_{t-1}) \tag{6}$$

$$\pi_t = \pi_{t-1} + \frac{N^{(t)}}{\Sigma_{\tau=t-L}^{t} N^{(\tau)}}(\pi^{(t)} - \pi_{t-1}) - \frac{N^{(t-L-1)}}{\Sigma_{\tau=t-L}^{t} N^{(\tau)}}(\pi^{(t-L-1)} - \pi_{t-1}) \tag{7}$$

In the above equations $D_t = \Sigma_{\tau=t-L}^{t} r^{(\tau)}$. The adaptivity of the model is controlled by the parameter L.

Matching of blobs to objects is performed by calculating the blob's normalised data log-likelihood $\mathscr{L}$ with respect to the object's GMM:

$$\mathscr{L} = \frac{1}{N^{(t)}} \Sigma_{\xi \in X^{(t)}} \log p(\xi | O) \tag{8}$$

The log-likelihood threshold g for accepting a match is adapted over time to take into account current and previous log-likelihoods. Given an array of n most recent data log-likelihoods calculated for the previous n frames, it is set to $g = \upsilon - k\sigma$ where υ is the median and σ is the standard deviation of the previous n data log-likelihood values.

2.4 Robust Principal Components Analysis

In order to acquire a stable model of object appearance over longer timescales, an extension of the Robust Principal Components Analysis (RPCA) method proposed in [2] is applied. RPCA enhances standard PCA by means of a pixel outlier process using M-estimators: Given n training images represented by column vectors $\mathbf{d}_i$ with d elements and with scale parameters $\sigma = [\sigma_1 \sigma_2 \ldots \sigma_d]^T$, this essentially entails minimizing the following robust energy function to obtain RPCA robust mean μ, bases $\mathbf{B}$, and coefficients $\mathbf{C}$:

$$\begin{aligned} E_{rpca}(\mathbf{B}, \mathbf{C}, \mu, \sigma) = & \quad \Sigma_{i=1}^{n} e_{rpca}(\mathbf{d}_i - \mu - \mathbf{B}\mathbf{c}_i, \sigma) \\ = & \quad \Sigma_{i=1}^{n} \Sigma_{p=1}^{d} \rho\left(d_{pi} - \mu_p - \Sigma_{j=1}^{k} b_{pj} c_{ji}, \sigma_p\right) \end{aligned} \tag{9}$$

where ρ is the Geman-McClure error function $\rho(x, \sigma_p) = \frac{x^2}{x^2 + \sigma_p^2}$ and σ_p is a scale parameter that controls convexity and hence determines which residual errors are treated as outliers. To robustly compute the mean and the subspace spanned by the first k principal components, we minimise equation 9 using gradient descent with a local quadratic approximation.

To ensure adequate performance for tracking, we have extended RPCA using a robust incremental subspace learning technique to efficiently re-compute the Eigenspace. In addition, rather than computing RPCA over image intensity alone, two approaches were implemented to retain colour information. The simpler approach maintains separate RPCA subspaces for the hue, saturation and luminance channel and performs matching through weighted summation of the Eigenspace Euclidean distances (see below). Best results were achieved by weighting distance in hue space with 0.5, saturation with 0.3, and luminance with 0.2, again reflecting the desirability of discounting absolute brightness values to achieve illumination invariance.

Secondly, we applied RPCA to one-dimensional colour statistics histograms derived from the HSV colour distribution of each object. Following Hanbury [3], a saturation-weighted hue histogram is calculated by using the HSV saturation values as a weight differentiating between chromatic and achromatic colours:

$$W_\theta = \Sigma_x S_x \delta_{\theta H_x} \tag{10}$$

where θ denotes a bin of the histogram over all pixel samples x with $\theta \in [0°, 1°, \ldots, 360°]$. S_x is the saturation of x, H_x the hue, and δ_{ij} is the Kronecker delta function. Alternatively, we implemented RPCA for a saturation-weighted hue mean histogram $H_{S\ell}$ or saturation-weighted mean length histogram $R_{n\ell}$. Both of these histograms are calculated at each sample luminance level. Given $N+1$ quantised luminance values, $\ell \in \{0, 1, 2, \ldots, N\}$, the following circular statistics descriptors are calculated for each value of ℓ:

$$A_{S\ell} = \Sigma_x S_x \, cos H_x \delta_{L_x \ell}; \; B_{S\ell} = \Sigma_x S_x \, sin H_x \delta_{L_x \ell}; \; H_{S\ell} = arctan\left(\frac{B_{S\ell}}{A_{S\ell}}\right); \; R_{n\ell} = \frac{\sqrt{A_{S\ell}^2 + B_{S\ell}^2}}{\Sigma_x \delta_{L_x \ell}} \tag{11}$$

RPCA based on the saturation-weighted hue mean histogram gave best results and is the method used in the experiments.

The arity of the pixel sample sets for Eigenspace computation was normalised by sub-sampling (and if necessary re-sampling) object pixels or through normalisation of the colour statistics histograms. Re-estimation of the RPCA model can be performed in batch mode by maintaining a moving window of previous samples (usually 10 or more). This approach was found to be cumbersome and consequently a far more efficient incremental algorithm was devised by adapting the method proposed in [10] to re-estimate the RPCA coefficients. Incremental learning of the subspace parameters also has the advantage of increased robustness in the context of an online estimation problem such as that of appearance modelling for tracking. Given the current RPCA robust mean $\mu^{(t)}$, bases $\mathbf{B}^{(t)}$, coefficients $\mathbf{C}^{(t)}$ and data sample $\mathbf{x}$, then at each frame t the algorithm proceeds as follows:

1. Project the data sample $\mathbf{x}$ into the current Eigenspace defined by a matrix $\mathbf{U}^{(t)} = [\mathbf{u}_1, \ldots \mathbf{u}_n]$ of Eigenvectors $\mathbf{u}_i$ and form the reconstruction $\mathbf{y}$ of the data set:

$$\mathbf{c} = \mathbf{U}^{(t)T}(\mathbf{x} - \mu^{(t)}); \quad \mathbf{y} = \mathbf{U}^{(t)}\mathbf{c} + \mu^{(t)} \tag{12}$$

2. Compute the residuum vector $\mathbf{r} = \mathbf{x} - \mathbf{y}$, which is orthogonal to $\mathbf{B}^{(t)}$, and form matrices $\mathbf{B}_e$ and $\mathbf{C}_e$:

$$\mathbf{B}_e = \left[\mathbf{B}^{(t)} \; \frac{\mathbf{r}}{||\mathbf{r}||}\right]; \quad \mathbf{C}_e = \begin{bmatrix} \mathbf{C}^{(t)} & \mathbf{c} \\ \mathbf{0} & ||\mathbf{r}|| \end{bmatrix} \tag{13}$$

3. Compute Robust PCA on $\mathbf{C}_e$, and obtain the updated robust mean μ_s and robust bases $\mathbf{B}_s$. Discard the least significant Eigenvector of the new basis $B_s = B_s(:, 1:k)$ and project the coefficients $\mathbf{C}_e$ to the new basis $\mathbf{B}_s$ to obtain the coefficient matrix for frame $t+1$:

$$\mathbf{C}^{(t+1)} = \mathbf{B}_s^T(\mathbf{C}_e - \mu_s \mathbf{1}_{1 \times t+1}) \tag{14}$$

where $\mathbf{1}_{m \times n}$ denotes a matrix of dimension $m \times n$ where all the elements are 1.

4. Calculate the new bases matrix $\mathbf{B}^{(t+1)}$ and new mean $\mu^{(t+1)}$ for frame $t+1$:

$$\mathbf{B}^{(t+1)} = \mathbf{B}_e \mathbf{B}_s; \quad \mu^{(t+1)} = \mu^{(t)} + \mathbf{B}_e \mu_s \tag{15}$$

In order to compute the match distance between a candidate blob represented by sample column vector **e**, and an object represented by RPCA basis vectors **B**, we compute the coefficients c_i which minimize:

$$E(\mathbf{c}) = \Sigma_{j=1}\rho\left(\left(\mathbf{e}_j - \left(\Sigma_{i=1}^{t} c_i B_{ij}\right)\right), \sigma\right) \tag{16}$$

where ρ is the Geman-McClure error function. The distance is then defined as the minimum of the Euclidean distances between the blob sample coefficients and each of the object Eigenspace coefficients.

3 Adaptive Integration

3.1 Motivation and Overview

Tracking algorithms that fuse multiple complementary cues have been shown to be much more robust than those that utilise only a single cue [11, 9, 6, 12, 7].

Adaptive colour histograms can be completely re-estimated easily from frame to frame, and they are robust to the sort of short term noise and blur that would confuse the RPCA model. However this may cause them to de-generate due to object motion or deformation. GMMs can be adapted selectively and they combine aspects of both parametric and non-parametric estimation. Their explicit probabilistic interpretation via model likelihoods lends itself to incorporation in a wide variety of tracking and modelling frameworks. However they still suffer from some of the disadvantages of a global statistic. RPCA has stability due to the statistical outlier process but is unable to cope well with short term changes in the object's appearance since these may appear as outliers. RPCA creates robust long term appearance models which can be used to re-acquire objects which have been temporarily lost due to occlusions or deformations.

Combining all three allows one to model intrinsic long term appearance (RPCA) as well as short term incidental changes (adaptive histogram, GMM) and expected variation of appearance due to object movement and gradual deformations. Much of the utility derives not from the models themselves but from the methods for matching and re-estimation (or adaptation). The important point about using appearance models for tracking is to model not only current appearance but also allowable (and hence expected) appearance variation.

3.2 Integration through CONDENSATION

Particle filtering algorithms such as CONDENSATION [7] pose the problem of tracking as estimation of states **X** from observations **Z** using the recursion:

$$p(\mathbf{X}_t|\mathbf{Z}_t) \propto \mathscr{L}(\mathbf{Z}_t|\mathbf{X}_t)\int p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\mathbf{Z}_{t-1})d\mathbf{X}_{t-1} \tag{17}$$

where the dynamical model $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ describes state evolution and the observation likelihood model $\mathscr{L}(\mathbf{Z}_t|\mathbf{X}_t)$ gives the likelihood of any state in light of current observations. The posterior probability distribution 17 is then represented by a weighted set of 'particles':

$$p(\mathbf{X}_t|\mathbf{Z}_t) = \{s_t^{(n)}, \pi_t^{(n)}|n = 1\ldots N\} \tag{18}$$

where $s_t^{(n)}$ is the nth sample and $\pi_t^{(n)}$ is the corresponding weight such that $\Sigma_n \pi^{(n)} = 1$. At each step of the CONDENSATION algorithm the evolution of the weighted sample set is calculated by applying the dynamical model to the set. The observation likelihood function is then used to correct the prediction by calculating the weight π_t of each element in the set i.e. $\pi_t \propto \mathscr{L}(\mathbf{Z}_t|\mathbf{X}_t^{(n)})$. N samples are then drawn with replacement, by choosing a particular sample with probability $\pi^{(n)} = p(Z_t|X_t = s_t^{(n)})$. The mean state vector of an object in frame t is then modelled as the expectation $E[S] = \Sigma_{n=1}^{N} \pi^{(n)} s^{(n)}$.

Here, we model the observation density by a function that contains Gaussian peaks where the observation density is assumed to be high, that is, where an object could have generated set of blobs with high probability. Each Gaussian peak corresponds to the position of a blob, and the peak is scaled by the object-blob distance. The likelihood $\mathscr{L}$ for a particle is computed as :

$$\mathscr{L}(\mathbf{Z}_t|\mathbf{X}_t) \propto exp(-k \times dist^2) \tag{19}$$

where *dist* is a distance under one of the appearance models of the local image patch at a given particle and the object under consideration, and k is a constant.

Likelihoods are calculated for each particle for each of the three appearance modelling schemes above and combined as follows:

$$\mathscr{L}(\mathbf{Z}_t|\mathbf{X}_t) \propto [\mathscr{L}_{rpca}(\mathbf{Z}_t|\mathbf{X_t})]^{\alpha_1} [\mathscr{L}_{chist}(\mathbf{Z}_t|\mathbf{X}_t)]^{\alpha_2} [\mathscr{L}_{gmm}(\mathbf{Z}_t|\mathbf{X}_t)]^{\alpha_3} \tag{20}$$

where $0 \leq \alpha_1, \alpha_2, \alpha_3 \leq 1$ are the reliability weights for each appearance model (they need not sum to 1), initialised to $\frac{1}{3}$.

3.3 Adaptation of cue weights

Adaptation of the weights in equation 20 is performed dynamically during tracking by extending the idea of Democratic integration [11] to the CONDENSATION framework. Four separate observation likelihoods are computed: one for the joint appearance model, and three for each of the RPCA, adaptive histogram and GMM appearance cues. CONDENSATION is performed separately for each of the observation functions, resulting in four hypotheses, R_{fused}, R_{rpca}, R_{chist}, and R_{gmm}, which are regions where the object is thought to be in the current frame. Each region centroid is obtained by computing the expectation of the respective particle sets for each cue.

The Euclidean distances $E_{k,t}$ between the centroid of R_{fused} and the centroids of R_{rpca}, R_{chist}, R_{gmm} are then calculated. Since the joint observation function is assumed to exhibit the best performance, colour cues which result in relatively large values of $E_{k,t}$ are considered less reliable in the current frame and their reliability weight is lowered accordingly. A score $\gamma_{k,t}$ is computed for each colour cue k as follows:

$$\gamma_{k,t} = \frac{tanh(-aE_{k,t} + b) + 1}{2} \tag{21}$$

where a, b are constants (set to 2 and 5 respectively). Given $\gamma_{k,t}$, the weights $\alpha_{k,t}$ for each cue k are then adapted using first order exponential averaging:

$$\alpha_{k,t+1} = \beta \gamma_{k,t} + (1 - \beta)\alpha_{k,t} \tag{22}$$

where β controls the rate of adaptation. Performing CONDENSATION four times during each frame was found not to be a bottleneck since most of the computation time is required for the particle distances (which need only be computed once per frame).

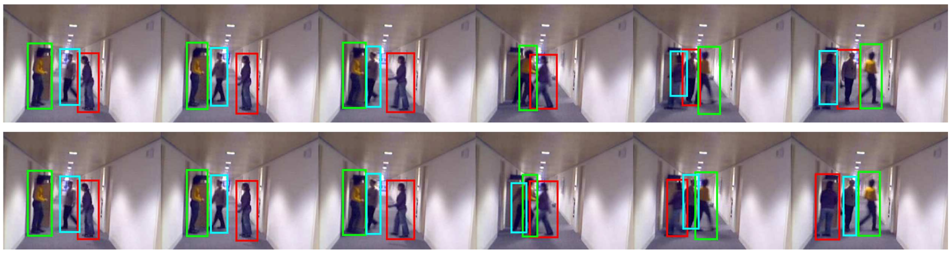

Figure 1: **Indoor tracking results. Top: tracking using only blob features and distances. Bottom: tracking using the robust fusion of adaptive appearance models as described above. Note how this allows identity of tracked entities (indicated by bounding box colour) to be maintained during and across occlusions.**

4 Results

To evaluate the adaptive appearance models and the fusion mechanism discussed above, testing was carried out on a number of indoor surveillance sequences. The tracking conditions are especially demanding due to the presence of intermittent bright lighting, fleshy coloured walls, motion blur and occlusions as the people interact. Figure 1 shows how the fusion framework makes tracking robust with respect to occlusions and movements of people. In figure 2 it is shown how the appearance modelling improves accuracy in light of erroneous blob hypotheses generated by the background differencing and blob detection framework.

In addition, video sequences and ground truth from the CAVIAR project[1] were used for quantitative performance evaluation. Each sequence has been annotated with the spatial location, angle of rotation and extent of bounding boxes around individuals and groups of people. Each such box is assigned a numerical label to identify it in subsequent frames. From this we derive a performance metric M_T which computes a weighted sum of relative centre-of-gravity distance, bounding box mass difference, and bounding box overlap for each of the tracked object compared to the manual annotations. Metric values are in the range 0 (all objects tracked perfectly) to 1.0 (none of the objects tracked).

Figure 3 shows results from the visual tracking for one frame of the CAVIAR sequences, while figure 4 shows results of adaptive fusion tracking over several frames of a sequence. For the sequence shown, we achieved an overall mean M_T of 0.874 using fusion of all three adaptive appearance models compared to scores of 0.723, 0.798 and 0.639 using only the histogram, GMM or RPCA model respectively.

5 Summary and Conclusions

Recent research has shown the benefits of employing robust statistical and machine learning techniques to improve the performance of visual object modelling and tracking. This paper shows how such methods can be employed to address the challenging problem of adaptive appearance modelling. While robust methods for adaptive parametric, non-parametric and semi-parametric colour modelling are shown to yield good results in iso-

[1] EC Funded CAVIAR project/IST 2001 37540, see http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

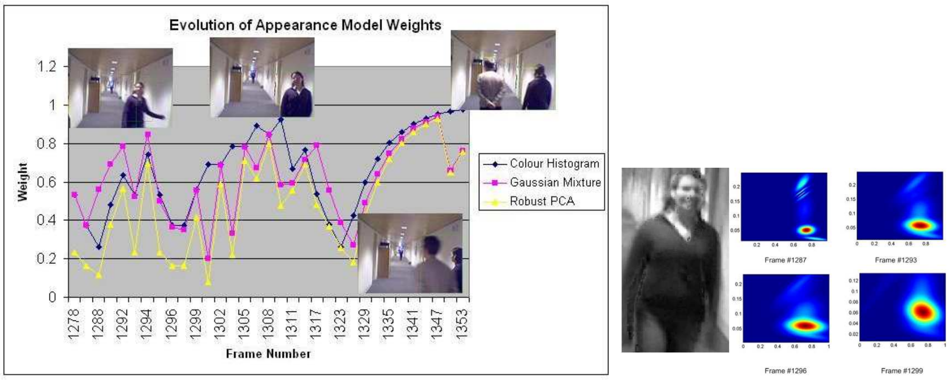

Figure 2: *Left*: **graph plotting the reliabilities of the appearance model colour cues for the woman shown in a test sequence. There is an initial rise in the reliability of all models due to the clear visibility of the woman. The large fall in reliability at frame 1320 onwards is due to occlusion by the man entering the scene. After the occlusion the appearance models successfully recover and their reliability increases very rapidly. Note the lag of the RPCA (and in some cases the Gaussian mixture) model behind the colour histogram model due to their slower adaptation.** *Right*: **Examples of RPCA and GMM appearance models during the sequence.**

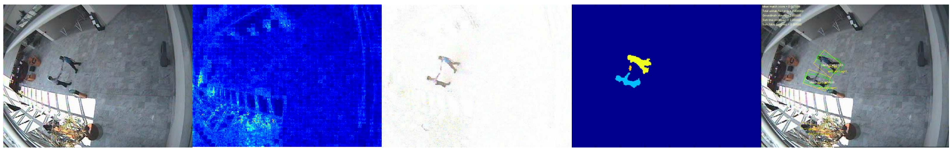

Figure 3: **Sample results for object detection and tracking on the CAVIAR data. From left to right: Original frame; background variances; background subtraction; detected blobs; resulting tracked objects (outlined in green) with ground truth data in yellow.**

lation, additional improvements in performance and robustness result from their adaptive probabilistic integration. The approach effectively leverages the strengths of the different cues while discounting their weaknesses.

References

[1] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *Proc. of the 5th European Conference on Computer Vision*, 1998.

[2] F. De la Torre and M. J. Black. Robust principal component analysis for computer vision. In *Proc. International Conference on Computer Vision*, 2001.

[3] A. Hanbury. Circular statistics applied to colour images. *8th Computer Vision Winter Workshop*, 2003.

[4] W.K. Leow and R. Li. Adaptive binning and dissimilarity measure for image retrieval and classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

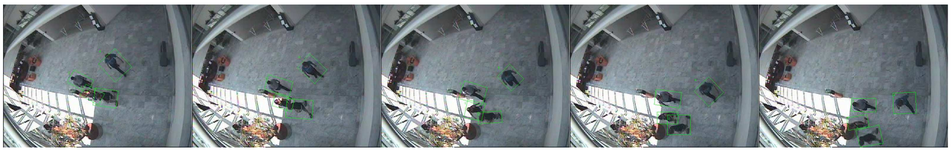

Figure 4: **Sample frames from a CAVIAR sequence showing tracked objects outlined in green.**

[5] S.J. McKenna, Y. Raja, and S. Gong. Object tracking using adaptive colour mixture models. *Asian Conference on Computer Vision*, 1351:607–614, 1998.

[6] K. Nummiaro, E. Koller-Meier, and L.V. Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21:99–110, 2003.

[7] P. Perez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. In *Proc. IEEE (issue on State Estimation), to appear*, 2004.

[8] A. Prati, I. Mikic, R. Cucchiara, and M. Trivedi. Comparative evaluation of moving shadow detection algorithms. In *Proc. Workshop on Empirical Evaluation Methods in Computer Vision*, 2001.

[9] J. Sherrah and S. Gong. Continuous global evidence-based Bayesian modality fusion for simultaneous tracking of multiple objects. In *Proc. International Conference on Computer Vision*, 2001.

[10] D. Skocaj and A. Leonardis. Robust continuous subspace learning and recognition. In *Proc. ERK'02*, 2002.

[11] M. Spengler and B. Schiele. Towards robust multi-cue integration for visual tracking. *Lecture Notes in Computer Science*, 2095:93–106, 2001.

[12] C.P. Town. Adaptive integration of visual tracking modalities for sentient computing. In *Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003.