Towards Improved Observation Models for Visual Tracking: Selective Adaptation

Jaco Vermaak, Patrick Pérez, Michel Gangnet, and Andrew Blake

Microsoft Research Cambridge, Cambridge CB3 0FB, UK
http://www.research.microsoft.com/vision

Abstract. An important issue in tracking is how to incorporate an appropriate degree of adaptivity into the observation model. Without any adaptivity, tracking fails when object properties change, for example when illumination changes affect surface colour. Conversely, if an observation model adapts too readily then, during some transient failure of tracking, it is liable to adapt erroneously to some part of the background. The approach proposed here is to adapt selectively, allowing adaptation only during periods when two particular conditions are met: that the object should be both present and in motion. The proposed mechanism for adaptivity is tested here with a foreground colour and motion model. The experimental setting itself is novel in that it uses combined colour and motion observations from a fixed filter bank, with motion used also for initialisation via a Monte Carlo proposal distribution. Adaptation is performed using a stochastic EM algorithm, during periods that meet the conditions above. Tests verify the value of such adaptivity, in that immunity to distraction from clutter of similar colour to the object is considerably enhanced.

1 Introduction

Visual tracking of objects in video sequences is becoming an increasingly important technology in a wide range of computer vision applications, including video teleconferencing, security and surveillance, video segmentation and editing. Numerous visual tracking algorithms have been developed based on *e.g.* the Kalman Filter [2], and Extended Kalman Filter [6], the mean-shift technique [3], exemplars [12] and particle filters [7].

One important issue that remains to be resolved concerns the adaptation and individualisation of object observation models. Adaptation is important because of its contribution in distinguishing between different objects, and in making tracking more robust to appearance variations due to changing illumination and pose. The importance of adaptation for tracking has been acknowledged by the tracking community, and some recent progress in the application of adaptive luminance and colour models has been reported [9,13,14].

Whilst adaptation of observation models is necessary, over-eager adaptation is a hazard. This is because transitory failures of tracking are compounded if, while attending to a piece of clutter, the adaptation algorithm learns clutter

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2350, pp. 645–660, 2002.
© Springer-Verlag Berlin Heidelberg 2002

characteristics and "forgets" the characteristics of the object. A solution to this problem, proposed here, is to allow adaption only *selectively* when two particular conditions are met. The first condition is that there is an object present, being tracked properly, as flagged by a binary indicator variable appended to the object state vector. The second condition is that the object should be moving, as a further protection against inadvertently adapting to a (stationary) background region. For this purpose, raw motion is observed via frame-difference signals in certain filter channels.

The setting for experiments is a somewhat complete tracking system. Information from the complementary modalities of motion and colour, are fused to enhance tracking accuracy and robustness. A novel aspect of the system is that observations are made entirely via a fixed bank of regularly spaced filters [8,11], some of which are colour sensitive while others are motion sensitive. When the object is moving strong localisation cues are provided by the motion measurements, whereas the colour measurements can undergo substantial fluctuations due to changes in the object pose and illumination. Conversely, when the object is stationary or near-stationary the motion information disappears, and colour information dominates to provides a reliable localisation cue. A Bayesian approach allows the balance between motion and colour to be captured automatically, using likelihood models for image measurements, both under the object hypothesis (foreground) and for the background.

The tracking engine itself is a particle filter [5,7], implementing Bayesian inference. The object of interest here is the face of a person in a video sequence, and its outline is modelled as an ellipse that is allowed to translate and scale subject to a Langevin dynamical model. Automatic object detection and particle filter initialisation are based on a segmentation of the motion measurements, and incorporated in a novel proposal distribution. Finally, adaptation of the object colour likelihood model is performed using a stochastic version of the EM algorithm during periods of motion.

The remainder of the paper is organised as follows. Section 2 formulates the tracking objectives in more detail and describes the object configuration and state-space, the measurement process, and the likelihood models for the motion and colour modalities. Section 3 outlines the particle filter tracking algorithm. The details of the particle proposal distribution and automatic initialisation strategy are presented in Section 4. Section 5 discusses the Monte Carlo EM algorithm to adapt the parameters of the object colour likelihood model. The performance of the proposed tracking algorithm is evaluated in Section 6. Finally, some conclusions are reached in Section 7.

2 Problem Formulation

The objective is to detect and track the face of a single person in a video sequence taken from a stationary camera. This section describes the necessary ingredients to achieve this within a particle filtering framework.

2.1 Object Description

For simplicity the face is modelled as an ellipse with a 1.25 aspect ratio, *i.e.* $(x/a)^2 + (y/b)^2 = 1$, with $b = 1.25a$. The object state is given by $\mathbf{x} = (x, y, s, r)$, where (x, y) is the location of the ellipse centre in the image, s is its scale, and r is a binary indicator signifying whether the object is present in the image ($r = 1$) or not ($r = 0$). The position and scale components are assumed to follow independent Langevin dynamical priors [1] when the object is present, whereas the priors for these components are undefined if the object is absent. The prior for the object indicator is taken to be a two state Markov process with initial state and state transition probabilities given by $P_0 = 0.5$ and $P_{01} = P_{10} = 10^{-3}$, respectively. More specifically, the prior model can be expressed as

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = p(x_t|x_{t-1}, r_t, r_{t-1})p(y_t|y_{t-1}, r_t, r_{t-1})p(s_t|s_{t-1}, r_t, r_{t-1})p(r_t|r_{t-1}),$$

where the first three factors are similarly defined, with the one for x given by

$$p(x_t|x_{t-1}, r_t, r_{t-1}) = \begin{cases} \text{undefined} & \text{if } r_t = 0 \\ p_L(x_t|x_{t-1}) & \text{if } r_t = 1 \text{ and } r_{t-1} = 1 \\ \mathcal{U}_{\mathcal{R}_x}(x_t) & \text{if } r_t = 1 \text{ and } r_{t-1} = 0, \end{cases}$$

where p_L denotes the Langevin dynamical model, $\mathcal{U}_{\mathcal{R}}$ denotes the uniform distribution over the set $\mathcal{R}$, and $\mathcal{R}_x$ is the valid range for the x location. In the above, the first case corresponds to objects being absent from the scene, whereas the second case corresponds to a valid object persisting in the scene. The third case corresponds to a new object entering the scene, in which case the prior for the state component is taken to be uniform over the valid range of the component.

2.2 Image Measurements

The raw image stream contains all the information necessary to detect and track objects. However, working with the raw images directly is difficult, and a certain amount of preprocessing is required to emphasise the salient features of the objects of interest. Two important modalities for visual tracking are motion and colour. These modalities complement each other in the sense that when the object is moving the colour information may become unreliable due to changes in the object pose and illumination, whereas strong localisation cues may be obtained from the motion information. Conversely, when the object is stationary or near-stationary the motion information disappears, whereas the colour information becomes more reliable. Here the hue and saturation parameterisation is used to capture the colour information to minimise the sensitivity to changes in illumination. The motion information is captured by computing the absolute value of the luminance frame-difference.

In the spirit of [8,11], the final image measurements are obtained by processing each of the channels (hue, saturation and frame-difference) on a regular

filter grid. At each of the G gridpoints an isotropic Gaussian filter is applied to each channel independently, so that the measurement for the i-th gridpoint becomes $\mathbf{y}_i = (H_i, S_i, D_i)$, with H_i, S_i and $D_i = |\Delta I_i|$ the outputs of the filters on the hue, saturation and absolute frame-difference channels, respectively. The standard deviation for the Gaussian filters is set to a quarter of the gridpoint separation to ensure some degree of independence between the image measurements. The set of measurements over the entire image is denoted by $\mathbf{y} = (\mathbf{y}_1 \cdots \mathbf{y}_G)$.

2.3 Likelihood Models

Within a statistical framework a likelihood function facilitates the evaluation of the goodness of a hypothesis $\mathbf{x}$ in the light of a given set of measurements $\mathbf{y}$. Assuming all the gridpoints to be independent, the likelihood here is of the form

$$L(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{G} L_i(\mathbf{y}_i|\mathbf{x}) = \prod_{i \in F(\mathbf{x})} L^F(\mathbf{y}_i) \times \prod_{i \in B(\mathbf{x})} L_i^B(\mathbf{y}_i),$$

where $F(\mathbf{x})$ is the set of foreground gridpoints covered by the object parameterised by $\mathbf{x}$, and $B(\mathbf{x})$ is the set of remaining gridpoints in the background. Note that each gridpoint i is associated with a unique background likelihood model L_i^B, whereas there is a single likelihood model L^F for all the gridpoints in the foreground. The expression above can be divided by the product of all the background likelihoods, which is a constant for any valid hypothesis, yielding

$$L(\mathbf{y}|\mathbf{x}) \propto \prod_{i \in F(\mathbf{x})} L^F(\mathbf{y}_i)/L_i^B(\mathbf{y}_i).$$

Thus, the likelihood only needs to be evaluated over the gridpoints in the foreground, resulting in significant computational savings. At each gridpoint the channel measurements are assumed to be conditionally independent given the state, so that the foreground and background likelihoods can be further decomposed as

$$\begin{aligned} L^F(\mathbf{y}_i) &= L^{FH}(H_i)L^{FS}(S_i)L^{FM}(D_i) \\ L_i^B(\mathbf{y}_i) &= L_i^{BH}(H_i)L_i^{BS}(S_i)L^{BM}(D_i). \end{aligned}$$

Note that each gridpoint is associated with unique background likelihood models for the colour components, whereas the background motion likelihood model is shared by all the gridpoints. For both the foreground and background colour likelihoods a histogram model with B bins is adopted. For a given scalar colour measurement c this model is defined as $L(c) = \gamma_{[c]}$, where $[c]$ is the index of the bin corresponding to the measurement, and $\sum_{i=1}^{B} \gamma_i = 1$. The background colour models are trained by collecting measurements at the gridpoints over a training sequence without any objects. The foreground colour models are initialised to skin-colour distributions by training on a set of labelled face images. To prevent numerical problems associated with empty bins each of the colour models is

supplied with a uniform component for which the mixture weight is typically set to 10^{-3}.

As is shown by the empirical evidence in Figure 1, the background frame-difference are gamma distributed, *i.e.*

$$L^{BM}(D_i) \propto D_i^{a_\Delta - 1} \exp(-b_\Delta D_i), \tag{1}$$

with typical values for the parameters given by $a_\Delta = 2.62$ and $b_\Delta = 326.90$. This distribution accounts for small camera motions due to vibration and electronic noise in the image acquisition process.

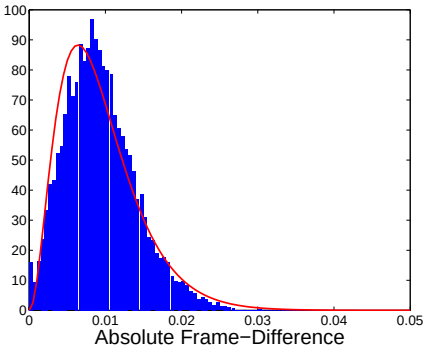

Fig. 1. Distribution of background frame-difference measurements. The absolute frame-difference measurements are gamma distributed if there is no motion in the scene.

Foreground frame-difference measurements depend on the magnitude of the motion, and the number and orientation of the foreground edges relative to the motion. Constructing a likelihood model to capture all these effects would be difficult. However, if the object is indeed moving, the foreground frame-difference measurements are generally substantially larger than the mean of the gamma background distribution, given by $\mu_\Delta = a_\Delta / b_\Delta$. For this reason a simple two component histogram model is adopted for the foreground frame-difference measurements, *i.e.*

$$L^{FM}(D_i) = \begin{cases} \beta & \text{if } 0 \leq D_i < n\mu_\Delta \\ m\beta & \text{if } n\mu_\Delta \leq D_i \leq D_{\max}, \end{cases} \tag{2}$$

where $D_{\max}$ is the maximum value for the absolute frame-difference measurements, and $\beta = (n(1-m)\mu_\Delta + mD_{\max})^{-1}$ is computed such that L^{FM} is a proper distribution. Typical values for the constants in the expressions above are $n = 3$ and $m = 10$. If the object is indeed moving the foreground frame-difference likelihood will typically be much larger than the background frame-difference likelihood, providing a strong localisation cue. However, if the object is present but

stationary, the background frame-difference likelihood will dominate, in which case object localisation cues are provided by the colour likelihood.

3 Particle Filter Tracking

For non-linear multi-modal models, as the one introduced here, the particle filter [5] provides a Monte Carlo solution to the recursive filtering equation $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto L(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}$ necessary for tracking. Starting with a weighted particle set $\{(\mathbf{x}_{t-1}^{(i)}, \pi_{t-1}^{(i)})\}_{i=1}^N$ approximately distributed according to $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$, the particle filter proceeds by predicting new samples from a suitably chosen proposal distribution which may depend on the old state and the new measurements, *i.e.* $\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)$. To maintain a consistent sample the new particle weights are set to

$$\pi_t^{(i)} \propto \pi_{t-1}^{(i)} L(\mathbf{y}_t|\mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i-1)}) / q(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t). \tag{3}$$

The new particle set $\{(\mathbf{x}_t^{(i)}, \pi_t^{(i)})\}_{i=1}^N$ is then approximately distributed according to $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. The particles are then resampled according to their weights to avoid degeneracy. Section 4 describes the proposal distribution used here in more detail.

4 Particle Proposal

The performance of the particle filter hinges on the quality of the proposal distribution. In [5] the optimal choice for the proposal distribution has been shown to be $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) \propto L(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})$. However, direct simulation from this distribution is intractable, calling for the design of a proposal distribution that best approximates it. Such a proposal should facilitate computationally efficient simulation and evaluation, and incorporate information about the measurements to guide the generation of new particles.

In terms of the behaviour of the object the particle filter exhibits three distinct operational phases, and the proposal must be designed to deal with each efficiently. The first phase is when an object first enters the scene. The proposal should be able to detect the object and spawn new particles in the region of the object. The second phase immediately follows the first and persists for as long as the object is in the scene. The proposal should allow the algorithm to successfully track the object, whether it is stationary or moving either slowly or rapidly. The third phase corresponds to the object leaving the scene. The proposal should detect this event and kill off the particles associated with the object. These ideas can be formalised by defining a proposal of the form

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t, \overline{P}_{1,t-1}) = q(r_t|r_{t-1}, \overline{P}_{1,t-1}) q(\mathbf{z}_t|\mathbf{z}_{t-1}, r_t, r_{t-1}, \mathbf{y}_t), \tag{4}$$

with

$$\overline{P}_{1,t} = \sum_{i=1}^{N} \pi_t^{(i)} r_t^{(i)} \tag{5}$$

$$q(r_t = 1 | r_{t-1} = 0, \overline{P}_{t-1}) = q_{10} = \begin{cases} P_{birth} & \text{if } \overline{P}_{1,t-1} = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$q(r_t = 0 | r_{t-1} = 1) = q_{01} = P_{death}$$

$$q(\mathbf{z}_t | \mathbf{z}_{t-1}, r_t, r_{t-1}, \mathbf{y}_t) = \begin{cases} \text{undefined} & \text{if } r_t = 0 \\ p_L(\mathbf{z}_t | \mathbf{z}_{t-1}) & \text{if } r_t = 1 \text{ and } r_{t-1} = 1 \\ \mathcal{N}\left(\mathbf{z}_t; \widehat{\boldsymbol{\mu}}_t, \widehat{\boldsymbol{\Sigma}}_t\right) & \text{if } r_t = 1 \text{ and } r_{t-1} = 0, \end{cases}$$

where $\mathbf{z} = (x, y, s)$ denotes the continuous components of the state-space, and $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Note that the proposal above also depends on the quantity $\overline{P}_{1,t-1}$, which is computed from the empirical particle distribution at the previous time step. Recent theoretical results [4] show that dependence on the empirical particle distribution in the proposal still leads to a particle filter that converges to the correct distribution.

From the proposal above it is evident that the birth of a new object is only allowed if there is not already an object alive in the scene, *i.e.* $\overline{P}_{1,t-1} = 0$. On the other hand, particles that are associated with an object that is alive in the scene are subjected to a fixed death probability. It is up to the reweighting and resampling stages to determine whether the decision to kill a particle should be enforced. Typical values for the birth and death probabilities are $P_{birth} = P_{death} = 0.1$. Note that these values are substantially larger than those for the corresponding parameters in the dynamical model. This is to ensure that a large enough proportion of the particles undergoes a birth/death process, enabling the instantaneous detection of objects entering/leaving the scene. If an object particle is dead ($r_t = 0$), the proposal for the object location and scale is undefined. If it is alive and persisting, the proposal for its location and scale is set to the object dynamics for these components. This is sufficient to maintain track under the assumptions that the particles are already locked on to the object, and the chosen dynamical model is broad enough to capture any expected object motion. For a new object entering the scene values for the object location and scale are simulated from a Gaussian birth proposal. The parameters of this proposal are computed using the motion measurements, as described below.

Object Detection

When an object enters the scene it is moving, and the motion measurements can be used to determine a rectangular region within which the object is likely to lie. This section proposes an efficient algorithm based on searching the horizontal and vertical projections of the frame-difference measurements to locate such a region. The size and location of this region can then be used to set the parameters of the birth proposal.

The image processing filters are arranged on a regular grid. Let G_x and G_y denote the number of filters in each row and column, respectively, with $G = G_x G_y$, and $\mathbf{y}_{i,j}$ the vector of measurements at the i-th row and j-th column of the grid. Since objects (faces) are assumed to be horizontally separated in most video sequences of interest, the object region boundaries in the x direction can be efficiently obtained by searching over the column projections of the frame-difference measurements. The object boundaries in the y direction can then be found by searching over the row projections of the frame-difference measurements within the region boundaries in the x direction. This two-step procedure is graphically illustrated in Figure 2.

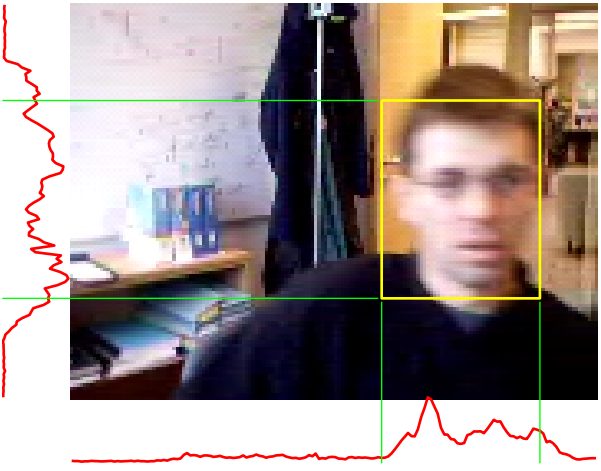

Fig. 2. Motion-based object detection. The object x region is located first by searching over the column projections of the frame-difference measurements (shown below the image). The object y region is then located by searching over the row projections of the frame-difference measurements within the object x region (shown to the left of the image). The size of the y region is constrained to yield within $\pm 10\%$ a 1.25 aspect ratio rectangular region.

The object search proceeds similarly in both directions, and only the first case is described in detail below. Let $\overline{\mathbf{D}}^x = (\overline{D}_1^x \cdots \overline{D}_{G_x}^x)$, with $\overline{D}_i^x = 1/G_y \sum_{j=1}^{G_y} D_{i,j}$, denote the vector of column projections of the frame-difference measurements, and denote by (d_x, W_x) the midpoint and halfwidth of the hypothesised object x region. Assuming the components of $\overline{\mathbf{D}}^x$ to be independent and following an argument similar to that in Section 2.3, the likelihood for the hypothesised object x region can be written as[1]

[1] The foreground and background likelihood models used here are identical to those defined in (2) and (1), respectively. Even though the statistics for the frame-difference projections (sums) are expected to differ from those for the single frame-difference measurements, the models were found to predict the projections equally well for the summation ranges in the application considered here.

$$L(\overline{\mathbf{D}}^x|d_x, W_x) \propto \prod_{i \in \mathcal{I}_x} L^{FM}(\overline{D}_i^x)/L^{BM}(\overline{D}_i^x),$$

with $\mathcal{I}_x = \{1 \cdots G_x\} \cap [d_x - W_x, d_x + W_x]$. An estimate of the object x region can then be obtained by maximising this likelihood, *i.e.*

$$(\widehat{d}_x, \widehat{W}_x) = \underset{(d_x, W_x)}{\arg\max}\, L(\overline{\mathbf{D}}^x|d_x, W_x),$$

under the constraint that $L(\overline{\mathbf{D}}^x|\widehat{d}_x, \widehat{W}_x) > 1$. This maximisation is performed by a computationally efficient coarse-to-fine hierarchical search over (d_x, W_x). An estimate of the object y region $(\widehat{d}_y, \widehat{W}_y)$ can be found in a similar way by maximising the likelihood over the vector of row projections within the object x boundaries. The size of the y region is constrained to yield within $\pm 10\%$ a 1.25 aspect ratio rectangular region.

Once the object region is determined the parameters of the birth proposal are set to (omitting the time subscript for brevity)

$$\widehat{\boldsymbol{\mu}} = (\widehat{d}_x, \widehat{d}_y, 0.5(\widehat{W}_x/a + \widehat{W}_y/b)), \quad \widehat{\boldsymbol{\Sigma}} = \mathrm{diag}(0.1(\widehat{W}_x, \widehat{W}_y, 1))^2.$$

Thus, the location and scale parameters are proposed independently. The proposal is centred on the object region, with its uncertainty proportional to the size of the region. The chosen values were found to give good performance in practice.

5 Colour Model Adaptation

Recall from Section 2.3 that the parameters of the foreground colour likelihood are initialised using a set of labelled face images. As such the model may be too broad to facilitate accurate localisation for a specific individual. Furthermore, it may be particularly sensitive to changes in the object pose and illumination. Thus, there is a large potential benefit to be gained by devising a strategy to individualise the colour model to a particular object and to adapt to colour changes due to changes in object pose and illumination.

The largest variations in the object appearance occur when it is moving. In this case the colour likelihood becomes less reliable, but strong localisation cues are provided by the complementary frame-difference likelihood. These cues anchor the particles tightly around the moving object. The colour information within the particle extents can then be used to update the colour likelihood parameters. The degree to which each particle contributes to this adaptation should be proportional to the strength of its frame-difference likelihood. Within a probabilistic setting this can be achieved by a stochastic version of the EM algorithm [10]. It should be emphasised that adaptation is only allowed if an object is present in the scene and moving. An object is said to be present if $\overline{P}_1$ in (5) is bigger than some threshold T_{obj}, which is typically set to $T_{obj} = 0.9$. Object motion is detected by monitoring when the average output of the frame-difference

filters covered by the particles encoding the object configuration, denoted by $\widetilde{D}$, exceeds some threshold T_Δ, which is typically set to $T_\Delta = 3\mu_\Delta$, where μ_Δ is the mean of the background frame-difference likelihood. The remainder of this section presents the details of the adaptation algorithm.

Denoting by $\boldsymbol{\theta} = (\boldsymbol{\gamma}^{FH}, \boldsymbol{\gamma}^{FS})$, with $\boldsymbol{\gamma}^{FH} = (\gamma_1^{FH} \cdots \gamma_B^{FH})$ and $\boldsymbol{\gamma}^{FS} = (\gamma_1^{FS} \cdots \gamma_B^{FS})$, the parameters of the histograms for the foreground colour likelihood, the objective at time t is to find the MAP parameter estimate, *i.e.* $\widehat{\boldsymbol{\theta}}_t = \arg\max_{\boldsymbol{\theta}_t} p(\boldsymbol{\theta}_t|\mathbf{y}_{1:t}, \widehat{\boldsymbol{\theta}}_{1:t-1})$. This estimation can be performed within the EM [10] framework by defining a Q-function of the form

$$\begin{aligned} Q(\boldsymbol{\theta}_t, \widehat{\boldsymbol{\theta}}_t) &= \mathbb{E}\left[\log p(\boldsymbol{\theta}_t|\mathbf{y}_{1:t}, \mathbf{x}_{1:t}, \widehat{\boldsymbol{\theta}}_{1:t-1})\right] \\ &\propto \mathbb{E}\left[\log p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}, \widehat{\boldsymbol{\theta}}_{1:t-1}, \boldsymbol{\theta}_t) + \log p(\mathbf{x}_{1:t}|\widehat{\boldsymbol{\theta}}_{1:t-1}, \boldsymbol{\theta}_t) + \log p(\boldsymbol{\theta}_t|\widehat{\boldsymbol{\theta}}_{1:t-1})\right], \end{aligned}$$

where $\widehat{\boldsymbol{\theta}}_t$ is now a preliminary MAP estimate of the parameters, and the expectation is relative to the full filtering distribution $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}, \widehat{\boldsymbol{\theta}}_{1:t})$. By eliminating terms independent of $\boldsymbol{\theta}_t$, this expression reduces to

$$Q(\boldsymbol{\theta}_t, \widehat{\boldsymbol{\theta}}_t) \propto \mathbb{E}\left[\log L(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_t)\right] + \log p(\boldsymbol{\theta}_t|\widehat{\boldsymbol{\theta}}_{t-1}),$$

where the expectation is now relative to the marginal filtering distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t}, \widehat{\boldsymbol{\theta}}_{1:t})$. This distribution is not known analytically, but using the particle set $\{(\mathbf{x}_t^{(i)}, \pi_t^{(i)})\}_{i=1}^N$ prior to resampling, a Monte Carlo approximation of the Q-function can be obtained as

$$\widehat{Q}_N(\boldsymbol{\theta}_t, \widehat{\boldsymbol{\theta}}_t) \propto \sum_{i=1}^N \overline{\pi}_t^{(i)} \log L(\mathbf{y}_t|\mathbf{x}_t^{(i)}, \boldsymbol{\theta}_t)\delta_1(r_t^{(i)}) + \log p(\boldsymbol{\theta}_t|\widehat{\boldsymbol{\theta}}_{t-1}),$$

where $\delta_x(\cdot)$ denotes the Kronecker delta function with mass at x, and $\overline{\pi}_t^{(i)} = \pi_t^{(i)} / \sum_{j=1}^N \pi_t^{(j)} \delta_1(r_t^{(j)})$. Note that the Monte Carlo approximation has been restricted to particles for which the object is alive in the scene, and the weights have been renormalised to sum to one over these particles. The M-step of the EM algorithm then updates the preliminary MAP parameter estimate by maximising this approximate Q-function, *i.e.*

$$\widehat{\boldsymbol{\theta}}_t \leftarrow \arg\max_{\boldsymbol{\theta}_t} \widehat{Q}_N(\boldsymbol{\theta}_t, \widehat{\boldsymbol{\theta}}_t). \tag{6}$$

Pure Monte Carlo EM, as described above, requires a redraw of the particles before proceeding with the next EM iteration. However, if the particle proposal is independent of the unknown parameters (as is the case here), a redraw is not strictly necessary, and the particle weights can simply be updated according to (3), using the new value of the parameters in the likelihood term. If the proposal distribution were to depend on the parameters, then after a few steps of the EM algorithm the particle approximation may deviate too much from the posterior

with the current parameter estimate, so that the Monte Carlo approximation of the Q-function would be poor. In this case a particle redraw is required after every EM iteration. The algorithm is initialised with the final parameter estimate at the previous time step.

To complete the model it remains to specify a prior distribution on the model parameters $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$. It is desirable to choose a prior that facilitates a closed-form solution for the M-step in (6), and allows a large degree of flexibility in its specific form. For these reasons conjugate Dirichlet priors are adopted for the parameters. For both the hue and saturation models these distributions are specified by (omitting the time subscript for brevity)

$$\mathcal{D}i\left(\gamma_1 \cdots \gamma_B; \alpha_1 \cdots \alpha_B\right) \propto \prod_{i=1}^{B} \gamma_i^{(\alpha_i - 1)},$$

where $\boldsymbol{\alpha} = (\alpha_1 \cdots \alpha_B)$ are the parameters of the Dirichlet distribution, with $\alpha_i > 0$, $i = 1 \cdots B$. The mean and variance of this distribution are given by $\mathbb{E}[\gamma_i] = \alpha_i/\alpha$ and $\mathbb{V}[\gamma_i] = \alpha_i(\alpha - \alpha_i)/(\alpha^2 + \alpha^3)$, respectively, with $\alpha = \sum_{i=1}^{B} \alpha_i$. The uniform prior is obtained if $\alpha_i = 1$, $i = 1 \cdots B$. To maintain temporal coherence the prior can be centred on the previous parameter estimate, with the variance set to reflect the confidence in this estimate. This can be achieved by setting $p(\boldsymbol{\gamma}_t|\boldsymbol{\gamma}_{t-1}) = \mathcal{D}i\left(\boldsymbol{\gamma}_t; C\boldsymbol{\gamma}_{t-1}\right)$, with $C > 0$. The prior mean and variance then become $\mathbb{E}[\gamma_{i,t}] = \gamma_{i,t-1}$ and $\mathbb{V}[\gamma_{i,t}] = \gamma_{i,t-1}(1-\gamma_{i,t-1})/(1+C)$, respectively. Thus, as $C \to \infty$ the prior variance goes to zero.

With the Dirichlet priors and the form of the likelihood in Section 2.3, the M-step leads to update rules for the parameters of the hue and saturation models of the form

$$\gamma_i = \frac{\overline{n}_i + \alpha_i - 1}{\sum_{j=1}^{B}(\overline{n}_j + \alpha_j) - B}, \tag{7}$$

where $\overline{n}_i = \sum_{j=1}^{N} \overline{\pi}^{(j)} n_i^{(j)} \delta_1(r_t^{(j)})$ is the weighted average bin counts for the i-th bin, with the i-th bin count for the j-th particle formally defined as $n_i^{(j)} = |\{k \in F(\mathbf{x}^{(j)}) : [c_k] = i\}|$, where $[c_k]$ denotes the histogram bin corresponding to a hue/saturation measurement at gridpoint k, and $|\cdot|$ denotes the set size operator. Note from (7) that in the case of a uniform prior the new parameters simply become the normalised weighted average bin counts.

The adaptation method described above is not the only one available. One alternative strategy would be to include the colour model parameters in the particle state-space. However, the corresponding increase in the state-space dimensionality increases the complexity of the estimation problem to a degree where many orders of magnitude more particles may be required to achieve the same performance as the Monte Carlo EM scheme. Furthermore, the colour model parameters are essentially auxiliary variables, and their posterior distribution is not of particular interest for tracking.

To conclude this section a summary of the complete tracking and adaptation algorithm is given below.

Algorithm 1: Monte Carlo EM Particle Filter

With $\{(\mathbf{x}_{t-1}^{(i)}, \pi_{t-1}^{(i)})\}_{i=1}^{N}$ and $\widehat{\boldsymbol{\theta}}_{t-1}$ the particle set and parameter estimate at the previous time step, proceed as follows at time t:

- Particle prediction: simulate $\widetilde{\mathbf{x}}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t, \overline{P}_{1,t-1})$, $i = 1 \cdots N$ (see (4)).
- Parameter adaptation:
  If $\overline{P}_{1,t-1} > T_{obj}$ and $\widetilde{D} > T_{\Delta}$:
  - Initialisation: set $\boldsymbol{\theta}^{(0)} = \widehat{\boldsymbol{\theta}}_{t-1}$, $\boldsymbol{\alpha}^{FH} = C\widehat{\boldsymbol{\gamma}}_{t-1}^{FH}$, $\boldsymbol{\alpha}^{FS} = C\widehat{\boldsymbol{\gamma}}_{t-1}^{FS}$.
  - MCEM loop: for $j = 1 \cdots L$:
    - Weight update: $\kappa_j^{(i)} \propto \pi_{t-1}^{(i)} \frac{L(\mathbf{y}_t|\widetilde{\mathbf{x}}_t^{(i)}, \boldsymbol{\theta}^{(j-1)}) p(\widetilde{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i-1)})}{q(\widetilde{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t, \overline{P}_{1,t-1})}$, $i = 1 \cdots N$.
    - Weight renormalisation: $\overline{\kappa}_j^{(i)} = \kappa_j^{(i)} / \sum_{k=1}^{N} \kappa_j^{(k)} \delta_1(r_t^{(k)})$, $i = 1 \cdots N$.
    - Parameter update (see (7)):

$$\boldsymbol{\theta}^{(j)} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \overline{\kappa}_j^{(i)} \log L(\mathbf{y}_t|\widetilde{\mathbf{x}}_t^{(i)}, \boldsymbol{\theta}) \delta_1(r_t^{(i)}) + \log p(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}_{t-1}).$$

  - Termination: set $\widehat{\boldsymbol{\theta}}_t = \boldsymbol{\theta}^{(L)}$.

  Else:
  - Set $\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1}$ and $\kappa_L^{(i)} \propto \pi_{t-1}^{(i)} \frac{L(\mathbf{y}_t|\widetilde{\mathbf{x}}_t^{(i)}, \widehat{\boldsymbol{\theta}}_t) p(\widetilde{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i-1)})}{q(\widetilde{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t, \overline{P}_{1,t-1})}$, $i = 1 \cdots N$.
- Particle reweighting: set $\widetilde{\pi}_t^{(i)} = \kappa_L^{(i)}$, $i = 1 \cdots N$.
- Resampling: set $\mathbf{x}_t^{(i)} = \widetilde{\mathbf{x}}_t^{j(i)}$, $\pi_t^{(i)} = 1$, $j(i) \sim \{\widetilde{\pi}_t^{(k)}\}_{k=1}^{N}$, $i = 1 \cdots N$.

■

6 Results

This section evaluates the performance of the proposed tracking and adaptation algorithm. For illustrative purposes the initial discussion will focus on the video sequence summarised in Figure 3[2]. The setting is a standard office environment with several objects in the background that can potentially confuse a generic skin-colour model. The first 50 frames are empty, and these were used to calibrate the background models. A person then enters the scene from the right and moves around, resulting in changes in pose. In the final part of the sequence the person is stationary. The size of the video frames is 160×120 and the video rate during acquisition was 30fps.

The algorithm was tested on this sequence with the adaptation first disabled, and subsequently enabled. In both cases $N = 100$ particles were used, with all the particles initialised to have no object alive in the scene. The free parameters

[2] Videos for all the results described in this section are available as AVI files at `http://research.microsoft.com/users/jacov/msr_work.htm`.

of the system were set according to the strategies described elsewhere in the text, but the algorithm proved to be robust over sensible ranges for these parameters. The adaptation algorithm was initialised with the colour model at the previous time step, and the prior was set to be centred on the initial values, with the variance constant fixed to $C = 100$. In all cases the adaptation algorithm was found to converge rapidly, and hence only a single EM iteration was performed at each time step. For the given image size and a 22×17 filter grid a non-optimised C++ implementation of the algorithm ran at 15fps on a 736MHz Pentium III with 512MB of RAM. With some care in the implementation real-time performance can easily be achieved.

The tracking results are summarised in Figure 3. In both cases the object is successfully detected and tracked during the period of motion when the frame-difference likelihood dominates. In the adaptive case the colour model is able to individualise to the specific object and adapt to changes in pose and illumination. When the object is stationary and the motion cues disappear, the generic skin-colour model is easily confused by skin-coloured objects in the background (the carpet in this case), and track is lost. In the adaptive case, however, the individualised colour model allows the algorithm to maintain lock on the object. Further results confirming that adaptivity allows successful tracking under adverse conditions are summarised in Figure 4.

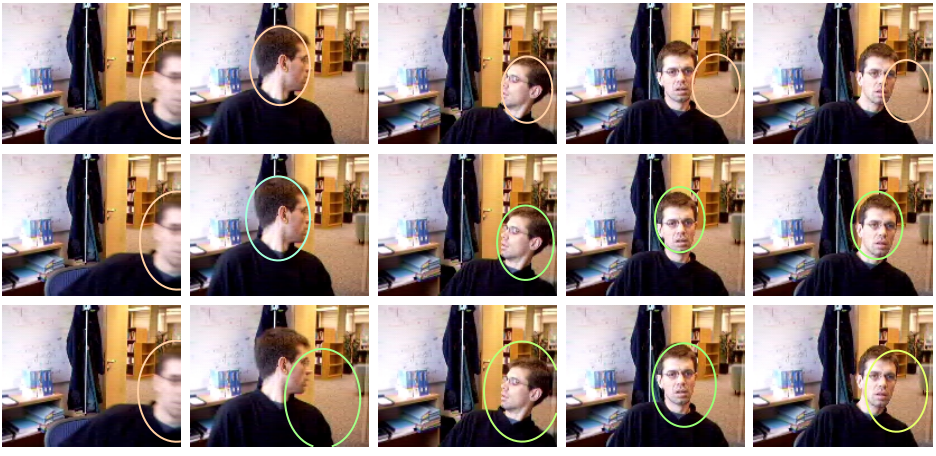

Fig. 3. Tracking results with adaptation disabled (top), enabled (middle and bottom) and the frame-difference likelihood disabled (bottom). The object is successfully detected (#58) and tracked during periods of motion (#106, #131) if the frame-difference likelihood is enabled (top, middle). With the object stationary (#275, #362) track is lost in the non-adaptive case (top) due to the skin-coloured carpet in the background, whereas the individualised model maintains lock (middle). With the frame-difference likelihood disabled (bottom) lock is soon lost and the colour model drifts to the background.

Fig. 4. Adaptivity allows tracking under adverse conditions. In the top sequence the head of the person is successfully detected and tracked despite substantial variations in pose and illumination, and the person momentarily leaving the scene. The first frame in the bottom sequence shows the face successfully tracked and the colour model adapted to the conditions. The person then leaves the scene and the particles are killed. Upon re-entering the tracker is wrongly initialised on the jumper, but the individualised colour model allows lock to be re-established after only a few frames, and the subsequent tracking proceeds successfully.

These results are further exemplified by the log-likelihood ratio maps in Figure 5. When the object is in motion the frame-difference likelihood clearly dominates, providing strong localisation cues. For a stationary object the generic skin-colour likelihood generates strong false positives in the background. These false positives are better suppressed if the colour model is allowed to adapt during periods of motion.

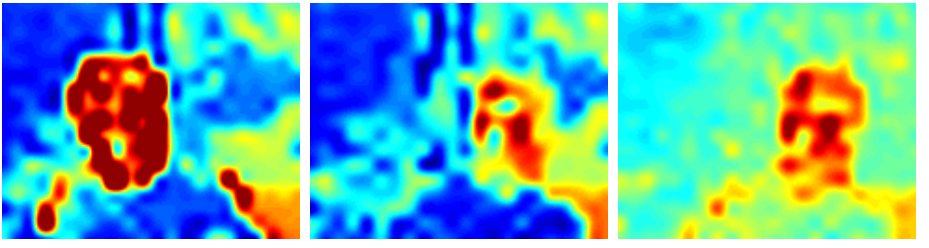

Fig. 5. Log-likelihood ratio maps. When the object is in motion the frame-difference likelihood dominates (left). With the object stationary the generic skin-colour model generates strong false positives in the background (middle). The false positives are better suppressed and the foreground model is stronger in the adaptive case (right).

The need for an anchor likelihood is easily demonstrated by disabling the frame-difference likelihood, and allowing adaptation at every time step, albeit

with a stronger prior. In this case the colour model invariably drifts and becomes fixated on some part of the background, as is illustrated in the bottom of Figure 3.

7 Conclusions

This paper demonstrated how the fusion of motion and colour measurements, automatic initialisation and colour model adaptation can increase the accuracy and robustness of a particle filter tracker. The adaptation algorithm, based on a stochastic version of the EM algorithm, allows the individualisation of the colour model to a specific object, and accommodates changes in the object pose and illumination. It relies heavily on the frame-difference likelihood to anchor the particles and prevent drift to the background.

In the extension to multiple objects the adaptation strategy is of crucial importance to establish and maintain object identity. The adaptation strategy can also be applied to the background colour models when it is certain that the corresponding filters are not occluded by a foreground object. Furthermore, update rules based on the same principles can also be derived for the parameters of the Gaussian mixture colour models used in [8]. All these issues are topics of ongoing and future research.

References

1. K.J. Astrom. *Introduction to Stochastic Control Theory.* Academic Press, 1970.
2. A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *Int. J. Computer Vision*, 11(2):127–145, October 1993.
3. D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. Conf. Comp. Vision Pattern Rec.*, pages II: 142–149, 2000.
4. A Doucet and D Crisan. A survey of convergence results on particle filtering for practitioners. *IEEE Trans. Signal Processing*, 2001. To Appear.
5. A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice.* Springer-Verlag, New York, 2001.
6. C. Harris. Tracking with rigid models. In A. Blake and A.L. Yuille, editors, *Active Vision*, pages 59–74. MIT, 1992.
7. M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *Int. J. Computer Vision*, 28(1):5–28, 1998.
8. M. Isard and J. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *Proc. Int. Conf. Computer Vision*, pages II: 34–41, 2001.
9. H.T. Nguyen, M. Worring, and R. van den Boomgaard. Occlusion robust adaptive template tracking. In *ICCV*, pages I: 678–683, 2001.
10. C. P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer-Verlag, New York, 1999.

11. J. Sullivan, A. Blake, M. Isard, and J. MacCormick. Bayesian object localisation in images. *IJCV*, 44(2):111–135, September 2001.
12. K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Proc. Int. Conf. Computer Vision*, pages II: 50–57, 2001.
13. Y. Wu and T. Huang. Color tracking by transductive learning. In *CVPR*, pages I: 133–138, 2000.
14. Y. Wu and T.S. Huang. A co-inference approach to robust visual tracking. In *Proc. Int. Conf. Computer Vision*, pages II: 26–33, 2001.