# Color-Based Probabilistic Tracking

P. Pérez[1], C. Hue[2], J. Vermaak[1], and M. Gangnet[1]

[1] Microsoft Research, 7 JJ Thomson Av., Cambridge CB3 0FB, UK
{pperez, jacov, mgangnet}@microsoft.com
[2] Irisa, Campus de Beaulieu, F35042 Rennes Cedex, France
chue@irisa.fr

**Abstract.** Color-based trackers recently proposed in [3,4,5] have been proved robust and versatile for a modest computational cost. They are especially appealing for tracking tasks where the spatial structure of the tracked objects exhibits such a dramatic variability that trackers based on a space-dependent appearance reference would break down very fast. Trackers in [3,4,5] rely on the deterministic search of a window whose color content matches a reference histogram color model.
Relying on the same principle of color histogram distance, but within a probabilistic framework, we introduce a new Monte Carlo tracking technique. The use of a particle filter allows us to better handle color clutter in the background, as well as complete occlusion of the tracked entities over a few frames.
This probabilistic approach is very flexible and can be extended in a number of useful ways. In particular, we introduce the following ingredients: multi-part color modeling to capture a rough spatial layout ignored by global histograms, incorporation of a background color model when relevant, and extension to multiple objects.

## 1 Introduction

Tracking objects, and more generally features, through the frames of an image sequence is an omnipresent elementary task in online and offline image-based applications including visual servoing, surveillance, gestural human-machine interface and smart environments, video editing and compression, augmented reality and visual effects, motion capture, medical and meteorological imaging, etc.

The combination of tools used to accomplish a given tracking task depends on whether one tries to track (I) objects of a given nature, e.g., cars, people, faces, (II) objects of a given nature with a specific attribute, e.g., moving cars, walking people, talking heads, face of a given person, (III) objects of *a priori* unknown nature but of a specific interest, e.g., moving objects, objects of semantic interest manually picked in the first frame.

In each case part of the input video frame is searched against a reference model describing the appearance of the object. This reference can be based on image patches, thus describing how the tracked region should look like pixel-wise, on contours, thus describing the overall shape, and/or on global descriptors such as color models.

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2350, pp. 661–675, 2002.

For problems of type (I) and (II), the instantiation of this reference model is exogenous. The reference is either chosen in an *ad-hoc* way, e.g., an ellipse outline for faces [1,18], or extracted from a set of examples like gray level appearance templates in [2] or outlines in [8,10,12].

In contrast, in problems of type (III) the instantiation of the reference has to arise from the sequence under consideration, thus being endogenous. In that case the reference can be extracted from the first frame and kept frozen, as color models in [3,4,5] and gray-level templates in [9], or adapted on the fly, using the tracking results from the previous frames, as gray-level templates in [14,15,17], deformable outlines in [16], and color models in [18,21,20].

This paper falls in the category of trackers using global color reference models and endogenous initialization. Such trackers have recently been proved robust and versatile for a modest computational cost [3,4,5]. They have in particular been proved to be very useful for tracking tasks where the objects of interest can be of any kind, and exhibit in addition drastic changes of spatial structure through the sequence, due to pose changes, partial occlusions, etc. This type of tracking problem arises for instance in the context of video analysis and manipulation. For such applications, most trackers based on a space-dependent appearance reference would break down very fast. In contrast, using a global, though sufficiently discriminant, model of the color distribution within the region of interest is an appealing way to address such complex tracking tasks.

The techniques introduced independently by Bradski ("CamShift" in [3]) and by Comaniciu *et al.* ("MeanShift" in [5]), and modified later by Chen *et al.* [4], are based on the following principle: the current frame is searched for a region, a fixed-shape variable-size window, whose color content best matches a reference color model. The search is deterministic. Starting from the final location in the previous frame, it proceeds iteratively at each frame so as to minimize a distance measure to the reference color histogram. Excellent tracking results on complex scenes are demonstrated in the three studies. This deterministic search might however run into problems when parts of the background nearby exhibit similar colors or when the tracked object is completely occluded for a while.

Improved handling of such situations is one of the benefits of the new color-based probabilistic tracking we propose. Relying on the same principle of comparing the color content of candidate regions to a reference color histogram, we embed it within a sequential Monte Carlo framework. This requires the building of a color likelihood based on color histogram distances, the coupling of this data model with a dynamical state space model, and the sequential approximation of the resulting posterior distribution with a particle filter. These different steps are described in Sect. 2. The use of a sample-based filtering technique permits in particular the momentary tracking of multiple posterior modes. This is the key to escape from background distraction and to recover after partial or complete occlusions, as demonstrated in Sect. 3.1.

This probabilistic approach is also very versatile in that is does not impose many constraints on the type of ingredients that can be incorporated in the state

space, the dynamics, and the data likelihood definitions. Hence our color-based tracking can be extended in a number of useful ways:

- Extension to multi-part color modeling as a way of incorporating a gross spatial layout ignored by global histogramming in Sect. 3.2;
- Incorporation of a background color model when relevant in Sect. 3.3;
- Extension to multiple objects in Sect. 3.4.

The treatment of the two last items within a particle filtering approach bears connections with the "BraMBLe" tracker introduced in [11]. We discuss these connections in detail in Sect. 4.

## 2 Probabilistic Tracking

### 2.1 Sequential Monte Carlo Tracking

Sequential Monte Carlo techniques for filtering time series [6] and their use in the specific context of visual tracking [10] have been described at length in the literature.

The starting point is a standard state space model, where a Markovian prior on the hidden states is coupled with a conditionally independent observation process. Denoting by $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ respectively the hidden state and the data at time $t$, and fixing the order of the dynamics to one, the sequence of filtering distributions $p(\boldsymbol{x}_t|\boldsymbol{y}_{0:t})$ to be tracked obeys the recursion

$$p(\boldsymbol{x}_{t+1}|\,\boldsymbol{y}_{0:t+1}) \propto p(\boldsymbol{y}_{t+1}|\,\boldsymbol{x}_{t+1}) \int_{\boldsymbol{x}_t} p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) p(\boldsymbol{x}_t|\,\boldsymbol{y}_{0:t}) d\boldsymbol{x}_t, \tag{1}$$

with the notation $\boldsymbol{x}_{0:t} \doteq (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t)$ and similarly for $\boldsymbol{y}$. In the case of linear Gaussian state space models, (1) can be handled analytically yielding the Kalman filter.

Unfortunately, in visual tracking problems the likelihood is non-linear, and often multi-modal, with respect to the hidden state. The reason being that the hidden variables indicate which part of the data set to look at. As a result the Kalman filter and its approximations are usually not suitable.

The recursion can however be used within a sequential Monte Carlo framework where the posterior $p(\boldsymbol{x}_t|\,\boldsymbol{y}_{0:t})$ is approximated by a finite set $\{\boldsymbol{x}_t^m\}_{m=1\cdots M}$ of $M$ samples, the particles. The generation of samples from $p(\boldsymbol{x}_{t+1}|\,\boldsymbol{y}_{0:t+1})$ is then obtained as follows. All the particles are moved independently by sampling from an appropriate proposal transition kernel $f(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t, \boldsymbol{y}_{t+1})$. If the $\boldsymbol{x}_t^m$'s are fair samples from the filtering distribution at time $t$, the new particles, denoted by $\tilde{\boldsymbol{x}}_{t+1}^m$, associated with the importance weights

$$\pi_{t+1}^m \propto \frac{p(\boldsymbol{y}_{t+1}|\,\tilde{\boldsymbol{x}}_{t+1}^m) p(\tilde{\boldsymbol{x}}_{t+1}^m|\,\boldsymbol{x}_t^m)}{f(\tilde{\boldsymbol{x}}_{t+1}^m; \boldsymbol{x}_t^m, \boldsymbol{y}_{t+1})} \text{ with } \sum_{m=1}^{M} \pi_{t+1}^m = 1, \tag{2}$$

approximate the new filtering distribution well. Resampling these particle according to their weights provides a set $\{\boldsymbol{x}_{t+1}^m\}_{m=1\cdots M}$ of fair samples from the filtering distribution $p(\boldsymbol{x}_{t+1}|\,\boldsymbol{y}_{0:t+1})$.

It can be shown (see [6]) that the optimal proposal density is proportional to $p(\boldsymbol{y}_{t+1}|\,\tilde{\boldsymbol{x}}_{t+1})p(\tilde{\boldsymbol{x}}_{t+1}|\,\boldsymbol{x}_t)$, but its normalization $\int_{\tilde{\boldsymbol{x}}_{t+1}} p(\boldsymbol{y}_{t+1}|\,\tilde{\boldsymbol{x}}_{t+1})p(\tilde{\boldsymbol{x}}_{t+1}|\,\boldsymbol{x}_t)$ cannot be computed analytically in our case. The chosen proposal density must then be sufficiently close to the optimal one such that the weights do not become all extremely small in the re-weighting process, resulting in a degeneracy of the sample approximation. The default choice (bootstrap filter) consists in taking $f(\boldsymbol{x}_{t+1};\boldsymbol{x}_t,\boldsymbol{y}_{t+1}) = p(\boldsymbol{x}_{t+1}|\,\boldsymbol{x}_t)$. In this case the weights become the data-likelihood associated with each hypothesized state $\tilde{\boldsymbol{x}}_{t+1}^m$.

Based on the discrete approximation of $p(\boldsymbol{x}_t|\,\boldsymbol{y}_{0:t})$, different estimates of the "best" state at time $t$ can be devised. We use, in a standard way, the Monte Carlo approximation of the expectation $\hat{\boldsymbol{x}}_t \doteq \frac{1}{M}\sum_{m=1}^M \boldsymbol{x}_t^m \approx \mathbb{E}(\boldsymbol{x}_t|\boldsymbol{y}_{0:t})$ as the tracker output at time $t$.

After the different ingredients of the model are defined in the next section, the complete procedure will be summarized in Proc. 1.

### 2.2 State Space and Dynamics

We aim to track a region of interest in the image plane. The shape of this region is fixed *a priori* through the definition of a 0-centered window $W$. It can be an ellipse or a rectangular box as in [3,4,5]. In our case, there is no restriction on the class of shapes that can be used. More complex hand-drawn or learned regions can be used if relevant. In any case, tracking then amounts to estimating in each frame the parameters of the transformation to be applied to $W$. Affinity or similitude transforms are classically considered. Given the global nature of the color information on which the proposed tracking relies, the choice of a simple similitude seems appropriate. Moreover, when the aspect ratio of the chosen region is close to one, the color information gathered over transformed regions will be rather insensitive to the rotation component of the similitude. As in [3,5] we thus only consider here the location $\boldsymbol{d} \doteq (x, y)$ in the image coordinate system and the scale $s$ as the hidden variables to be estimated.

A second-order auto-regressive dynamics is chosen on these parameters. In accordance with the first-order formalism used in the previous subsection, we define the state at time $t$ as $\boldsymbol{x}_t = (\boldsymbol{d}_t, \boldsymbol{d}_{t-1}, s_t, s_{t-1})$. The dynamics then reads

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{x}_{t-1} + C\boldsymbol{v}_t,\ \ \boldsymbol{v}_t \sim \mathcal{N}(0, \Sigma). \tag{3}$$

Matrices $A$, $B$, $C$ and $\Sigma$ defining this dynamics could be learned from a set of representative sequences where correct tracks have been obtained in some way. For the time being we use an ad-hoc model composed of three independent constant velocity dynamics on $x_t$, $y_t$ and $s_t$ with respective standard deviations 1 pixel/frame, 1 pixel/frame, and 0.1 frame$^{-1}$.

### 2.3 Color Model

Color models are obtained by histogramming techniques in the Hue-Saturation-Value (HSV) color space [7] in order to decouple chromatic information from shading effects. Color information is however only reliable when both the saturation and the value are not too small. Hence, we populate an HS histogram with $N_h N_s$ bins using only the pixels with saturation and value larger than two thresholds set to 0.1 and 0.2 respectively in our experiments. The remaining "color-free" pixels can however retain a crucial information when tracked regions are mainly black and white. We thus found useful to populate $N_v$ additional value-only bins with them. The resulting complete histogram is thus composed of $N = N_h N_s + N_v$ bins. We shall denote $b_t(\boldsymbol{u}) \in \{1, \ldots, N\}$ the bin index associated with the color vector $\boldsymbol{y}_t(\boldsymbol{u})$ at pixel location $\boldsymbol{u}$ in frame $t$.

Given an occurrence of the state vector $\boldsymbol{x}_t$, the candidate region in which color information will be gathered is defined as $R(\boldsymbol{x}_t) \doteq \boldsymbol{d}_t + s_t W$. Within this region a kernel density estimate $\boldsymbol{q}_t(\boldsymbol{x}) = \{q_t(n; \boldsymbol{x})\}_{n=1\cdots N}$ of the color distribution at time $t$ is given by [5]

$$q_t(n; \boldsymbol{x}) = K \sum_{\boldsymbol{u} \in R(\boldsymbol{x})} w(|\boldsymbol{u} - \boldsymbol{d}|) \delta[b_t(\boldsymbol{d}) - n] \tag{4}$$

where $\delta$ is the Kronecker delta function, $K$ is a normalization constant ensuring $\sum_{n=1}^{N} q_t(n; \boldsymbol{x}) = 1$, $w$ is a weighting function, and locations $\boldsymbol{u}$ lie on the pixel grid, possibly sub-sampled for efficiency reasons. This model associates a probability to each of the $N$ color bins. In [3,4,5] the weight function is a smooth kernel such that the gradient computations required by the iterative optimization process can be performed. This is not required by our approach where competing hypotheses associated with the particles simply have to be evaluated. Hence we set $w \equiv 1$, which amounts to standard bin counting.

At time $t$, the color model $\boldsymbol{q}_t(\boldsymbol{x})$ associated with a hypothesized state $\boldsymbol{x}$ will be compared to the reference color model $\boldsymbol{q}^\star = \{q^\star(n)\}_{n=1\cdots N}$, with $\sum_{n=1}^{N} q^\star(n) = 1$. In our experiments, the reference distribution is gathered at an initial time $t_0$ at a location/scale $\boldsymbol{x}_{t_0}^\star$ (Fig. 1), which is either manually selected, as in [3,4,5], or automatically provided by a detection module (as in Sect. 3.5). In either case:

$$\boldsymbol{q}^\star = \boldsymbol{q}_{t_0}(\boldsymbol{x}_{t_0}^\star). \tag{5}$$

The data likelihood must favor candidate color histograms close to the reference histogram, we therefore need to choose a distance $D$ on the HSV color distributions. Such a distance is used in the deterministic techniques [3,4,5] as the criterion to be minimized at each time step. In [5], $D$ is derived from the Bhattacharyya similarity coefficient, and defined as

$$D[\boldsymbol{q}^\star, \boldsymbol{q}_t(\boldsymbol{x})] = \left[1 - \sum_{n=1}^{N} \sqrt{q^\star(n) q_t(n; \boldsymbol{x})}\right]^{\frac{1}{2}} \tag{6}$$

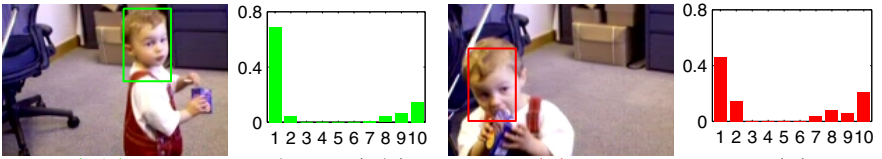


**Fig. 1. Color histograms used for tracking.** A reference color histogram $\boldsymbol{q}^\star$ is gathered at time $t_0$ within a region picked either manually, or automatically if only a given class of objects is searched and a corresponding detector can be devised. At time $t$ and for a hypothesized state $\boldsymbol{x}$, the candidate color histogram $\boldsymbol{q}_t(\boldsymbol{x})$ is gathered within the region $R(\boldsymbol{x})$.

with the argument that, contrary to the Kullback-Leibler divergence, this distance between probability distributions is a proper one, is bounded within $[0, 1]$, and empty bins are not a source of concern.

We use the same distance. Gathering statistics on a number of window sequences obtained from successful tracking runs (obtained by a contour-based tracker on sequences with no background clutter), we observed a consistent exponential behavior for the squared distance $D^2$. Letting $p(\boldsymbol{y}_t|\,\boldsymbol{x}_t) \propto p(D^2[\boldsymbol{q}^\star, \boldsymbol{q}_t(\boldsymbol{x})])$,[1] we thus choose:

$$p(\boldsymbol{y}_t|\,\boldsymbol{x}_t) \propto \exp -\lambda D^2[\boldsymbol{q}^\star, \boldsymbol{q}_t(\boldsymbol{x}_t)]. \tag{7}$$

Lacking a way to estimate satisfactorily the parameter $\lambda$, we fixed it to the same value $\lambda = 20$ in all the experiments reported in the paper. This value is in good agreement with the range of values estimated on the labeled sequences mentioned above. As for the bin numbers, we used the default setting $N_h = N_s = N_v = 10$ in all experiments.

## 3 Results and Extensions

### 3.1 Base Tracker

The probabilistic color-based tracker introduced in the previous section is summarized in Procedure 1.

As the deterministic color-based trackers in [3,4,5] it allows robust tracking of objects undergoing complex changes of shape and appearance (Fig. 2). Due to its Monte Carlo nature, however it better handles the confusion caused by similar color spots in the background (Fig. 3) and by complete occlusions (Fig. 4).

[1] In fact, the likelihood can only obey this relation if the mapping $\{b_t(\boldsymbol{u}), \boldsymbol{u} \in R(\boldsymbol{x})\} \mapsto \boldsymbol{q}_t(\boldsymbol{x}) \mapsto D^2[\boldsymbol{q}^\star, \boldsymbol{q}_t(\boldsymbol{x})]$ from quantized sub-images to $\mathbb{R}^+$ is such that the size of the pre-image of any distance value is independent from this value. It is easy to exhibit counter-examples to show that this does not hold.

**Procedure 1** Particle filter iteration for single object color-based tracking

Input: $\boldsymbol{q}^\star = \{q^\star(n)\}_{n=1\cdots N}$ reference color histogram

- Current particle set: $\{\boldsymbol{x}_t^m\}_{m=1\cdots M}$
- *Prediction*: for $m = 1 \cdots M$, draw $\tilde{\boldsymbol{x}}_{t+1}^m$ from second-order AR dynamics.
- *Computation of candidate histograms*: for $m = 1 \cdots M$, compute $\boldsymbol{q}_{t+1}(\tilde{\boldsymbol{x}}_{t+1}^m)$ according to (4).
- *Weighting*: for $m = 1 \cdots M$ compute

$$\pi_{i+1}^m = K \exp \sum_{n=1}^{N} \lambda \sqrt{q^\star(n) q_{t+1}(n; \tilde{\boldsymbol{x}}_{t+1}^m)}$$

  with $K$ such that $\sum_{k=1}^M \pi_{i+1}^k = 1$
- *Selection*: for $m = 1 \cdots M$, sample index $a(m)$ from discrete probability $\{\pi_{i+1}^k\}_k$ over $\{1 \cdots M\}$, and set $\boldsymbol{x}_{t+1}^m = \tilde{\boldsymbol{x}}_{t+1}^{a(m)}$.

#1 #40 #64 #71 #77

**Fig. 2. Color-based tracking.** Using a global color reference model picked in the initial frame, a region of interest (player 75 here) can be tracked robustly, despite large motions, important motion blur, dramatic shape changes, and partial occlusions. These results, similar to those obtained by MeanShift in [5], were obtained with our Monte Carlo tracker. Note that the *football* sequence we used was in addition subsampled by a factor of two in time, which makes the displacement and appearance changes from one frame to another even more extreme.

As for the computational cost, a non-optimized implementation tracks regions of average size $25 \times 25$ pixels at a rate of 50fps with $M = 100$ particles on a 747Mhz Pentium III. The bottle-neck is the building of the $M$ histograms at each time-step.

Beside the nice behavior demonstrated above, our approach can be extended in a number of useful ways. We introduce and demonstrate four extensions to the basic model in the remainder of this section. They respectively deal with the breaking up of tracked regions into several color patches, the introduction of a background model in the case of a still camera, the coupling with a skin detector for face tracking, and the extension to multi-object tracking.

### 3.2 Multi-part Color Model

If the tracked region contains different patches of distinct colors, e.g., the face and clothes of a person, the histogram-based modeling will capture them. However,

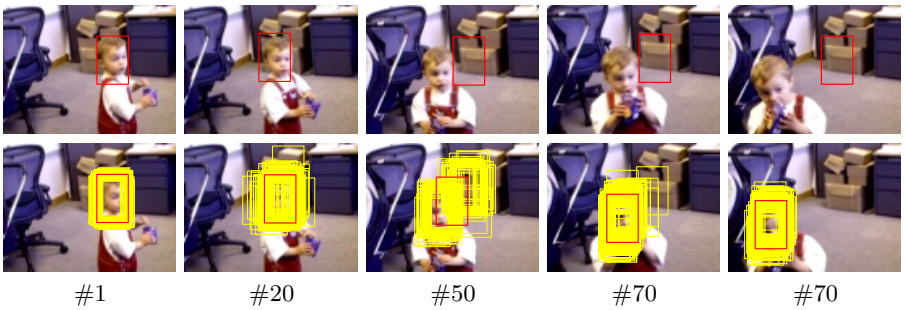


**Fig. 3. Probabilistic color-based tracking under distraction.** (Top) Deterministic color-based tracking can get stuck on local minima of the color-based metric, such as the boxes in the background. (Bottom) In contrast, by propagating a sample-based approximation of the filtering distribution, our Monte Carlo approach can be more robust to distraction. It can indeed track momentarily multiple modes (e.g., in frame 50) and then escape from distraction from the background.

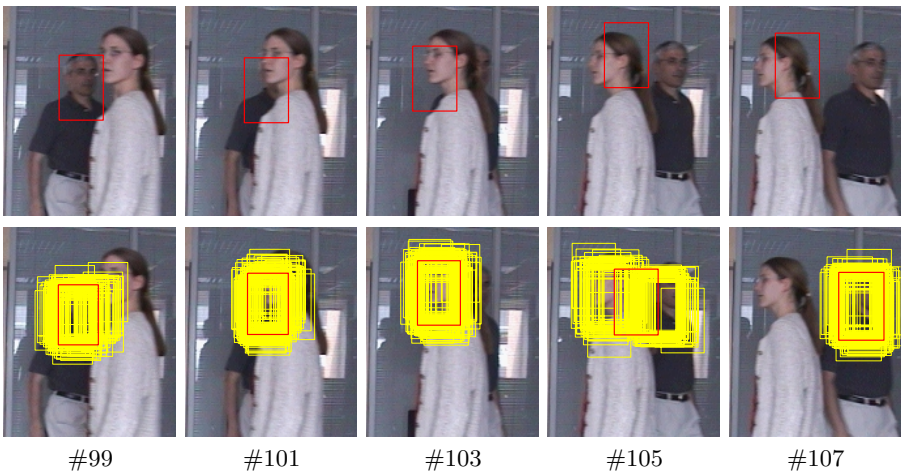


**Fig. 4. Probabilistic color-based tracking through occlusions.** (Top) Complete occlusion of the tracked region during a few frames (two in this case) can make deterministic color-based tracking break down. (Bottom) The ability of particle filtering to momentarily track multiple modes of the posterior (e.g., in frame 105) allows us to keep track of the region of interest even after the complete occlusion behind a region of similar color.

all information about the relative spatial arrangement of these different patches within the window will be lost. Keeping track of this coarse spatial layout might be beneficial to improve the tracker performance. Such a goal is easily achieved within our model by splitting the tracked region into sub-regions with individual reference color models. Formally, we consider the partition $R(\boldsymbol{x}) = \cup_{j=1}^{J} R_j(\boldsymbol{x})$ associated with the set $\{\boldsymbol{q}_j^\star\}_{j=1\ldots J}$ of reference color histograms. These regions

are rigidly linked in that the state space is kept unchanged with a unique location and scale vector per hypothesized object.

Assuming conditional independence of the image data within the different sub-regions defined by the state $\boldsymbol{x}_t$, the likelihood becomes:

$$p(\boldsymbol{y}_t|\,\boldsymbol{x}_t) \propto \exp -\lambda \sum_{j=1}^{J} D^2[\boldsymbol{q}_j^\star, \boldsymbol{q}_{j,t}(\boldsymbol{x}_t)] \tag{8}$$

where the histogram $\boldsymbol{q}_{j,t}(\boldsymbol{x}_t)$ is collected in region $R_j(\boldsymbol{x}_t)$ of image $\boldsymbol{y}_t$.

By capturing the coarse spatial layout of colors, this multi-part extension to our probabilistic color-based tracker is more accurate (better positioning on the tracked object, better capturing of the scale) thus avoiding the drift, and possible subsequent loss, experienced sometimes by the single-part version. This is illustrated in Fig. 5.

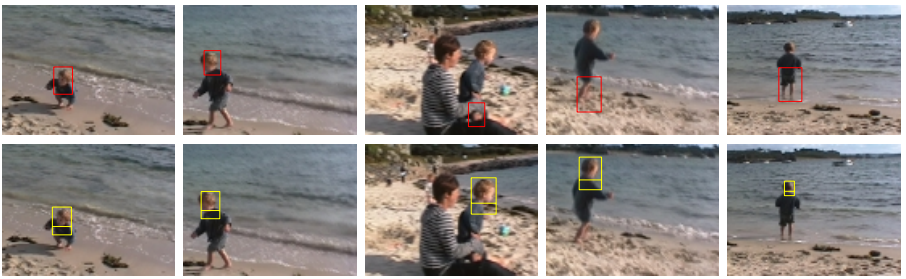


**Fig. 5. Multi-part color model improves tracking**. (Top) Because a single histogram does not capture any information on the spatial arrangement of colors, a noticeable drift can occur in some cases. Here, the tracker shifted from the head and jumper of the baby, to the jumper and the legs, confusing two regions with similar global color contents but different spatial layouts. (Bottom) The splitting of the region of interest into two parts with individual color models improved drastically the tracking in this shaky video, with large and chaotic motions, distracting colors in the background (the sand, the face of the mother very close in frame 209), and important scale changes at the end.

### 3.3 Background Modeling

In particular situations such as surveillance, desktop interaction, or smart rooms, where the camera is fixed and views of the background can be acquired off-line, the robustness of tracking can be dramatically enhanced by incorporating background knowledge in the model. We assume here that a background reference image $\tilde{\boldsymbol{y}}$ is available. The incorporation of the background information in the deterministic approach could be done by trying to minimize

$D^2[\boldsymbol{q}^\star, \boldsymbol{q}_t(\boldsymbol{x})] - D^2[\boldsymbol{q}^\bullet(\boldsymbol{x}), \boldsymbol{q}_t(\boldsymbol{x})]$ where $\boldsymbol{q}^\bullet(\boldsymbol{x})$ is the counterpart of $\boldsymbol{q}_t(\boldsymbol{x})$ computed in the reference background image $\tilde{\boldsymbol{y}}$ (Fig. 6).

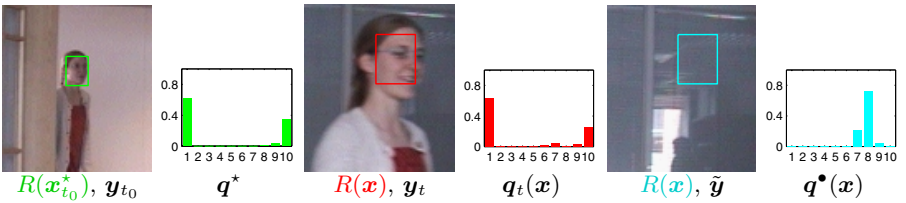


**Fig. 6. Color histograms for tracking with background model.** The color histogram $\boldsymbol{q}_t(\boldsymbol{x})$ associated with the hypothesized region $R(\boldsymbol{x})$ at time $t$ is not only compared to the reference $\boldsymbol{q}^\star$ but also to the color histogram $\boldsymbol{q}^\bullet(\boldsymbol{x})$ collected within the same region in the reference background image $\tilde{\boldsymbol{y}}$.

As in Section 3.2, statistics of the difference of the squared Bhattacharyya distance, which lies in $[-1, 1]$, are gathered on a set of cropped training sequences, and exhibits a shifted exponential distribution. We thus choose the new likelihood as:

$$p(\boldsymbol{y}_t | \boldsymbol{x}_t) \propto \exp -\lambda \left( D^2[\boldsymbol{q}^\star, \boldsymbol{q}_t(\boldsymbol{x})] - D^2[\boldsymbol{q}^\bullet(\boldsymbol{x}), \boldsymbol{q}_t(\boldsymbol{x})] \right) \tag{9}$$

where the shift is absorbed in the normalization factor. The merit of this complementary information will be illustrated in the challenging case of multiple face tracking with automatic initialization (Section 3.5).

### 3.4 Multiple Objects

We now extend the model to the simultaneous tracking of multiple objects. To this end, the likelihood is defined conditioned on the number of objects $K_t$ present in the scene at time $t$. If $K_t = k$, then the state $\boldsymbol{x}_t = (\boldsymbol{x}_{1,t} \cdots \boldsymbol{x}_{k,t})$ is a concatenation of $k$ single-object states. Each object $\boldsymbol{x}_{i,t}$ is associated with a reference color model $\boldsymbol{q}^\star_i$. If the $k$ corresponding regions $R(\boldsymbol{x}_{i,t})$, $i = 1 \cdots k$, do not overlap, the data-likelihood is simply the product of single object likelihoods. As soon as at least two objects overlap, one has to be careful not to explain the same piece of image data several times [13]. In the overlapping case, we form the likelihood by marginalizing out all the possible relative depth orderings for each non-empty region intersection. For instance, for two overlapping objects indexed $i$ and $j$ the data likelihood reads:

$$p(\boldsymbol{y}_t | \boldsymbol{x}_{i,t}, \boldsymbol{x}_{j,t}) = 0.5[p_{ij}(\boldsymbol{y} | \boldsymbol{x}_{i,t}, \boldsymbol{x}_{j,t}) + p_{ji}(\boldsymbol{y} | \boldsymbol{x}_{i,t}, \boldsymbol{x}_{j,t})] \tag{10}$$

where $p_{ij}(\boldsymbol{y}_t | \boldsymbol{x}_{i,t}, \boldsymbol{x}_{j,t})$ is the data likelihood under the hypothesis that object $i$ occludes object $j$. In the absence of background subtraction, and for single-part objects, it is defined as

$$p_{ij}(\boldsymbol{y} | \boldsymbol{x}_{i,t}, \boldsymbol{x}_{j,t}) \propto \exp -\lambda \left( D^2[\boldsymbol{q}^\star_i, \boldsymbol{q}_t(\boldsymbol{x}_{i,t})] + D^2[\boldsymbol{q}^\star_j, \boldsymbol{q}_t(\boldsymbol{x}_{j,t} | \boldsymbol{x}_{i,t})] \right) \tag{11}$$

where the histogram $\boldsymbol{q}_t(\boldsymbol{x}_{j,t}|\boldsymbol{x}_{i,t})$ is obtained on the subregion $R(\boldsymbol{x}_{j,t}) \setminus R(\boldsymbol{x}_{i,t})$.

Note that the two possible relative depth orderings of the two objects are assumed equally likely. We could also enforce a temporal continuity constraint that prevents the depth ordering from changing during the time that objects cross. This would require, however, the capturing of the global depth ordering of the whole set of objects maintained in the state space. Note that when tracking is performed within the 3D space, as in [11], the depth ordering is directly accessible.

Figure 7 shows an example where two persons are tracked from a manual initialization. The ability to track an unknown and varying number of objects without manual initialization must rely on a prior knowledge of the background and/or a rough appearance model of the objects of interest. We already presented in Section 3.3 a simple way to incorporate background knowledge in our approach. To achieve a full fledged multiple object tracker, we address below the issue of appearance-based automatic initialization in the case of face tracking.

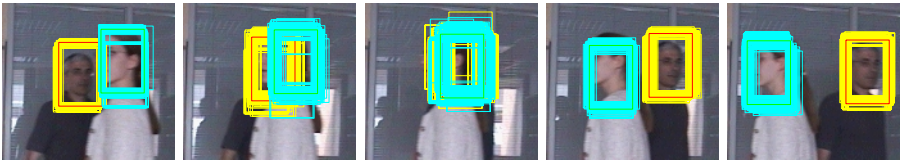


**Fig. 7. Multiple object tracking through occlusion.** The combination of the state space dynamics along with the definition of one color-model per tracked entity and the averaging of the color-likelihood over possible depth orderings, permits to track objects that cross each other, without identity swapping or loss of track, even if the two color models are similar. This sequence is the same as in Fig. 4

### 3.5 Automatic Initialization on Skin

A number of tracking applications focus on people. Many trackers specialized for this use rely, at least partially, on skin color models (e.g., [1,18]). Such a specialization is easily incorporated within our system. A normalized HSV histogram $\tilde{\boldsymbol{q}}$ with $N = N_h N_s + N_v$ bins is learned from hand-labeled face images, 20 frontal identity photographs with various illumination conditions, in our experiments. At each instant pixels are labeled as skin/non-skin by thresholding their likelihood under this skin color model. At time $t$, when a cluster of skin-labeled pixels falls within the specified size range for the objects of interest, and is not too close to an existing cloud of particles, if any, a new object is originated from this cluster. The large regions of false alarms produced by, e.g., wooden doors and desks, can be eliminated in the case of still camera, by passing detected pixels through a second motion-based sieve obtained by thresholding the

frame-difference (Fig. 8). The good performance of the multiple face tracker with automatic initialization and reference background is illustrated in Fig. 9.

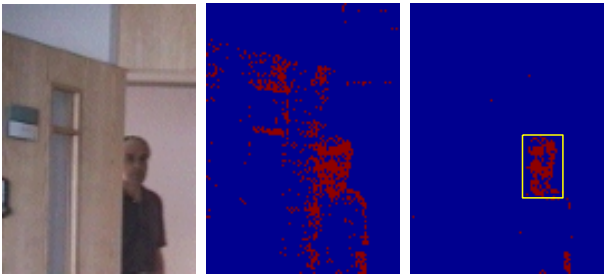

**Fig. 8. Simple color-based face detection.** A color histogram model learnt offline is used to assign each pixel with a probability to be in a skin region. (Middle) The thresholding of this probability, at level 0.3 in this example, already provides a good localization of skin patches, but a large number of false alarms in the background as well. (Right) Combining this skin detection with a very crude motion detection (thresholding the absolute frame difference, at level 10 here) permits the extraction of correct skin patches among which faces can be detected based on the aspect ratio of the region.

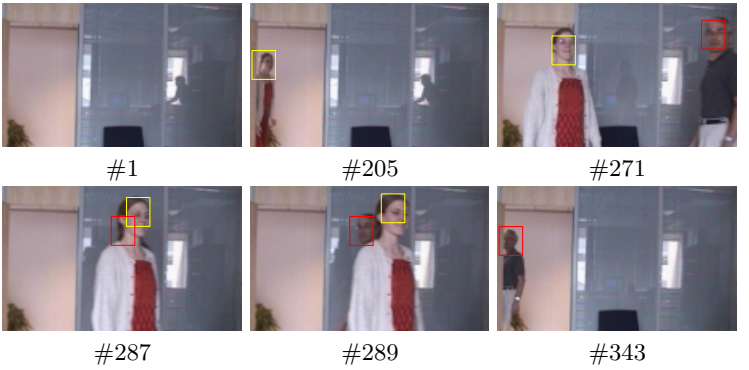


**Fig. 9. Automatic multiple face tracking in still scenes.** Combining the multi-object color-based tracker with a color modeling of the background in image 1 and the detection of moving skin patches, faces of moving persons can be detected as they enter the scene, and tracked without confusion even when they cross each other for a few frames.

## 4 Discussion

Embedding the color-based deterministic trackers introduced in [3,5] within a probabilistic framework we further improve their robustness and versatility. We achieve in particular the robust tracking of fast moving and changing regions within color distractions (Figs. 3 and 5), as well as the simultaneous tracking of multiple objects under temporary occlusions (Figs. 4 and 9).

In the case where a background model is used for multiple object tracking, our Monte Carlo tracker bears connections with the "Bramble" tracker introduced in [11]. In Bramble, an individual Gaussian mixture model capturing both colors and color gradients is associated with each point (on a subsampled grid) of the background. A generic Gaussian mixture model captures the same type of information for the foreground (e.g., the persons). The ratio of the two likelihoods over the grid-points within an hypothesized region (projection of a 3D generalized cylinder in the image plane) provides the weights for the particles.

As for the definition of hypothesis likelihood in terms of foreground and background models, an interesting connection can be worked out, as detailed in the Appendix. One nice aspect of the data likelihood in Bramble is that it explains the whole scene irrespective of the number of hypothesized objects. This enables the inclusion of the number of objects as part of the state space, since two particles maintaining different numbers of objects at the same instant can be legitimately compared using the weights based on this likelihood. Using birth and death processes, the handling of the varying object numbers can then be done consistently within the Bayesian framework. In practice, however, our experience of this type of approach is that after resampling, most of the particles share the same number of objects.

Another important difference related to the color modeling is that the instantiation of the foreground reference model in our approach makes it specific to the tracked object. As demonstrated in Sects. 3.5-6, this prevents swapping objects at crossings, a problem encountered by Bramble.

As for background modeling, the fact that it is related to individual spatial locations in Bramble, whereas it is built on the fly within the hypothesized region in our case, might make our use of the reference background more robust to slight camera motions. The experimental assessment of this could be part of the continuing exploration of our probabilistic tracking framework.

Other research perspectives concern the learning of the data likelihood, the automatic split of tracked regions into multiple color patches, the optimal combination of the color-based likelihood with more classic contour-based likelihoods with the following questions: how to assess adaptively which modality is the best to rely on at a given instant, and how to make the color and/or contour reference evolve in time for improved tracking robustness [19].

**Acknowledgments.** The authors thank Andrew Blake for his comments and encouragements.

## References


1. S.T. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proc. Conf. Comp. Vision Pattern Rec.*, pages 232–237, Santa Barbara, CA, June 1998.
2. M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. J. Computer Vision*, 26(1):63–84, 1998.
3. G.R. Bradski. Computer vision face tracking as a component of a perceptual user interface. In *Workshop on Applications of Computer Vision*, pages 214–219, Princeton, NJ, Oct. 1998.
4. H.T. Chen and T.L. Liu. Trust-region methods for real-time tracking. In *Proc. Int. Conf. Computer Vision*, pages II: 717–722, Vancouver, Canada, July 2001.
5. 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, Hilton Head, SC, June 2000.
6. A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.
7. J. Foley, A Van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, 1990.
8. D. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. Europ. Conf. Computer Vision*, Dublin, Ireland, June 2000.
9. G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(10):1025–1039, 1998.
10. M. Isard and A. Blake. Condensation–conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1):5–28, 1998.
11. M. Isard and J. MacCormick. BraMBLe: a Bayesian multiple-blob tracker. In *Proc. Int. Conf. Computer Vision*, pages II: 34–41, Vancouver, Canada, July 2001.
12. C. Kervrann and F. Heitz. A hierarchical Markov modeling approach for the segmentation and tracking of deformable shape. *Graph. Mod. Image Proc.*, 60(3):173–195, 1998.
13. J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. Int. Conf. Computer Vision*, pages 572–578, 1999.
14. H.T. Nguyen, M. Worring, and R. van den Boomgaard. Occlusion robust adaptive template tracking. In *Proc. Int. Conf. Computer Vision*, pages I: 678–683, Vancouver, Canada, July 2001.
15. N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. Robotics and Automation*, 9:14–35, 1993.
16. N. Peterfreund. The velocity snake: Deformable contour for tracking in spatio-velocity space. *Computer Vision and Image Understanding*, 73(3):346–356, 1999.
17. A. Rahimi, L.P. Morency, and T. Darrell. Reducing drift in parametric motion tracking. In *Proc. Int. Conf. Computer Vision*, pages I: 315–322, Vancouver, Canada, July 2001.
18. M. Spengler and B. Schiele. Towards robust multi-cue integration for visual tracking. In *Int. Workshop on Computer Vision Systems*, Vancouver, Canada, July 2001.
19. J. Vermaak, P. Pérez, M. Gangnet, and A. Blake. Towards improved observation models for visual tracking: selective adaptation. In *Proc. Europ. Conf. Computer Vision*, Copenhagen, Denmark, May 2002.

20. Y. Wu and T. Huang. Color tracking by transductive learning. In *Proc. Conf. Comp. Vision Pattern Rec.*, pages I:133–138, Hilton Head, SC, June 2000.
21. Y. Wu and T.S. Huang. A co-inference approach to robust visual tracking. In *Proc. Int. Conf. Computer Vision*, pages II: 26–33, Vancouver, Canada, July 2001.

## Appendix: A Connection with Bramble [11]

If our histogram-based likelihood was defined using an exponential distribution on the Kullback-Leibler divergence (instead of the squared Bhattacharyya distance), but with arguments swapped as compared to the choice in [4], then the likelihood can be re-interpreted in terms of individual grid-point contributions. Indeed

$$KL[\boldsymbol{q}_t(\boldsymbol{x})\|\boldsymbol{q}^\star] = \sum_{n=1}^{N} q_t(n;\boldsymbol{x}) \log \frac{q_t(n;\boldsymbol{x})}{q^\star(n)} = \mathbb{E}_{\boldsymbol{q}_t(\boldsymbol{x})}\left[\log \frac{\boldsymbol{q}_t(\boldsymbol{x})}{\boldsymbol{q}^\star}\right]. \tag{12}$$

If the window is large enough, the expectation on the right-hand-side can be approximated by the average log-ratio over the window, since the $b_t(\boldsymbol{u})$ for $\boldsymbol{u} \in R(\boldsymbol{x})$ can be seen as samples from $\boldsymbol{q}_t(\boldsymbol{x})$:

$$\mathbb{E}_{\boldsymbol{q}_t(\boldsymbol{x})}\left[\log \frac{\boldsymbol{q}_t(\boldsymbol{x})}{\boldsymbol{q}^\star}\right] \approx \frac{1}{|R(\boldsymbol{x})|} \sum_{\boldsymbol{u}\in R(\boldsymbol{x})} \log \frac{q_t[b_t(\boldsymbol{u});\boldsymbol{x}]}{q^\star[b_t(\boldsymbol{u})]}. \tag{13}$$

Using this approximation, we obtain:

$$\exp -\lambda KL[\boldsymbol{q}_t(\boldsymbol{x})\|\boldsymbol{q}^\star] \approx \prod_{\boldsymbol{u}\in R(\boldsymbol{x})} \left(\frac{q^\star[b_t(\boldsymbol{u})]}{q_t[b_t(\boldsymbol{u});\boldsymbol{x}]}\right)^{\frac{\lambda}{|R(\boldsymbol{x})|}} \tag{14}$$

In the case where background subtraction is incorporated, as described in Sect. 3.3, the likelihood ratio is then approximated by:

$$\exp -\lambda(KL[\boldsymbol{q}_t(\boldsymbol{x})\|\boldsymbol{q}^\star] - KL[\boldsymbol{q}_t(\boldsymbol{x})\|\boldsymbol{q}^\bullet(\boldsymbol{x})]) \approx \prod_{\boldsymbol{u}\in R(\boldsymbol{x})} \left(\frac{q^\star[b_t(\boldsymbol{u})]}{q^\bullet[b_t(\boldsymbol{u});\boldsymbol{x}]}\right)^{\frac{\lambda}{|R(\boldsymbol{x})|}}. \tag{15}$$

This is, as in [11] (apart from the raising to power $\frac{\lambda}{|R(\boldsymbol{x})|}$) the product over the grid points in the candidate region of the ratio of point-wise data likelihoods, under the foreground model and under the location-dependent background model respectively.