IDIAP RESEARCH REPORT

EMBEDDING MOTION IN MODEL-BASED STOCHASTIC TRACKING

Jean-Marc Odobez [*] Daniel Gatica-Perez [*]
Sileye Ba [*]

IDIAP–RR 03-72

DECEMBER 10, 2003

SUBMITTED FOR PUBLICATION

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[*] IDIAP, Martigny, Switzerland

IDIAP Research Report 03-72

EMBEDDING MOTION IN MODEL-BASED STOCHASTIC TRACKING

Jean-Marc Odobez Daniel Gatica-Perez Sileye Ba

DECEMBER 10, 2003

SUBMITTED FOR PUBLICATION

Abstract. Particle filtering is now established as one of the most popular methods for visual tracking. Within this framework, two assumptions are generally made. The first is that the data are temporally independent given the sequence of object states. In this paper, we argue that in general the data are correlated, and that modeling such dependency should improve tracking robustness. The second assumption consists of the use of the transition prior as proposal distribution. Thus, the current observation data is not taken into account, requesting the noise process of this prior to be large enough to handle abrupt trajectory changes. Therefore, many particles are either wasted in low likelihood area, resulting in a low efficiency of the sampling, or, more importantly, propagated on near distractor regions of the image, resulting in tracking failures. In this paper, we propose to handle both issues using motion. Explicit motion measurements are used to drive the sampling process towards the new interesting regions of the image, while implicit motion measurements are introduced in the likelihood evaluation to model the data correlation term. The proposed model allows to handle abrupt motion changes and to filter out visual distractors when tracking objects with generic models based on shape or color distribution representations. Experimental results compared against the CONDENSATION algorithm have demonstrated superior tracking performance.

1 Introduction

Visual tracking is an important problem in computer vision, with applications in teleconferencing, visual surveillance, gesture recognition, and vision based interfaces [4]. Though tracking has been intensively studied in the literature, it is still a challenging task in adverse situations, due to the presence of ambiguities (e.g. when tracking an object in a cluttered scene or when tracking multiple instances of the same object class), the noise in image measurements (e.g. lighting problems), and the variability of the object class (e.g. pose variations).

In the pursuit of robust tracking, Sequential Monte Carlo methods [1, 6, 4] have shown to be a successful approach. In this temporal Bayesian framework, the probability of the object configuration given the observations is represented by a set of weighted random samples, called particles. This representation allows to simultaneously maintain multiple-hypotheses in the presence of ambiguities, unlike algorithms that keep only one configuration state [5], which are therefore sensitive to single failure in the presence of ambiguities or fast or erratic motion.

Visual tracking with a particle filter requires the definition of two main elements : a data likelihood term and a dynamical model. The data likelihood term evaluates the likelihood of the current observation given the current object state, and relies on the object representation we have chosen. The object representation corresponds to all the information that explicitly or implicitly characterize the object like the target position, geometry, appearance, motion etc. Parametrized shapes like splines [4] or ellipses [18] and color distributions [13, 5, 11, 18] are often used as target representation. One drawback of these generic representations is that they are quite unspecific which augment the chances of ambiguities. One way to improve the robustness of a tracker consists of combining low-level measurements such as shape and color [18]. A step further to render the target more discriminative is to use appearence-based models such as templates [15, 16], leading to very robust trackers. However, such representations do not allow for large changes of appearence, unless adaptation is performed or more complex global appearence models are used (e.g. eigen-space [2] or set of examplars [17]).

The dynamical model characterizes the prior on the state sequence. Examples of such models range from simple constant velocity models to more sophisticated oscillatory ones or even mixtures of these [8]. A common assumption in particle filtering approaches is to use the dynamics as proposal distribution (or importance function), that is, as the function that predicts the new state hypotheses where the data likelihood will be evaluated. Thus, with this assumption, the variance of the noise process in the dynamical model implicitly defines some search range for the new hypotheses. This assumption raises difficulties in the modeling of the dynamics since this term should fulfill two contradictory objectives. On one hand, as prior, dynamics should be tight to avoid the tracker being confused by distractors in the vicinity of the true object configuration, a situation that is likely to happen for unspecific object representations such as generic shapes or color distributions. On the other hand, as proposal distribution, it should be broad enough to cope with abrupt motion changes. Besides, the proposal distribution does not take into account the most recent observations.

Thus, particles drawn from it will probably have a low likelihood, which results in a low efficiency of the sampling mechanism. Overall, such a particle filter is likely to be distracted by background clutter.
Different approaches have been proposed to address these issues. For instance, auxiliary information, if available, can be used to draw samples from, like color in [7], or audio in the case of audio-visual tracking [3]. An important advantage of this approach is to allow for automatic (re)initialization. However, from a filtering point of view, one drawback is that, since these additional samples are not related to the previous samples, the evaluation of the transition prior term for one new sample involves all past samples, which can become very costly. To avoid this effect, [12] proposed another auxiliary particle filter. The idea is to use the likelihood of a first set of predicted samples at time $t+1$ to resample the seed samples at time t, and to then apply the standard propagation and evaluation steps on these seed samples. The feedback from the new data acts by increasing or decreasing the number of descendents of a sample according to its "predicted" likelihood. Such a scheme, however, works well only if the variance of the transition prior is small, which is usually not the case in vision tracking. [14] proposed to use the unscented particle filter to generate importance densities. Although attractive, it is still likely to fail in the presence of abrupt motion changes, and the method needs to convert likelihood evaluations (e.g. color) into state space measurements (e.g. translation, scale). This would be difficult with color distribution likelihoods and for some state parameters. In [14], only a translation state is considered. .

In this paper we propose a new particle filter tracking method based on visual motion. More precisely, we first argue that a standard hypothesis of this filter, namely the independence of observations given the state sequence [2, 4, 7, 14, 17, 18], is inaccurate in the case of visual tracking. In this view, we propose a model that assumes that the current observation depends on the current and previous object configuration as well as on the past observation. As we will show, the proposed model can be exploited to introduce an implicit object motion likelihood in the data term. Secondly, we will make a further use of visual motion by exploiting explicit motion measurements in the proposal distribution and in the likelihood term. The benefits of this new model are two-fold. On one hand, it increases the sampling efficiency by handling unexpected motion, allowing for a reduced noise variance in the propagation process as well as the introduction of non-gaussian prior. On the other hand, the introduction of data-correlation between successive images will turn generic trackers like shape or color histogram trackers into more specific ones without resorting to complex appearence based models. As a consequence, it reduces the sensitivity of the algorithm to the difference noise variances setting in the proposal and prior since, when using a larger values, potential distractors should be filtered out by the introduced correlation and visual motion measurements.

The rest of the paper is organized as follows. In the next Section, we briefly present the standard particle filter algorithm. Our approach is motivated in Section 3, while Section 4 describes the proposed model. Section 5 presents the results and Section 6 provides some concluding discussions.

2 Particle filtering

Particle filtering is a technique for implementing a recursive Bayesian filter by Monte-Carlo simulations. The key idea is to represent the required density function by a set of random samples with associated weights. Let $c_{0:k} = \{c_l, l = 0, \ldots, k\}$ (resp. $z_{1:k} = \{z_l, l = 1, \ldots, k\}$) represents the sequence of states (resp. of observations) up to time k. Furthermore, let $\{c^i_{0:k}, w^i_k\}_{i=1}^{N_s}$ denote a set of weighted samples that characterizes the posterior probability density function (pdf) $\mathrm{p}(c_{0:k}|z_{0:k})$, where $\{c^i_{0:k}, i = 1, \ldots, N_s\}$ is a set of support points with associated weights w^i_k. The weights are normalized such that $\sum_i w^i_k = 1$. Then, a discrete approximation of the true posterior at time k is given by :

$$\mathrm{p}(c_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} w^i_k \delta(c_{0:k} - c^i_{0:k}) \,. \tag{1}$$

The weights are chosen using the principle of Importance Sampling (IS). More precisely, suppose that we could draw the samples $c^i_{0:k}$ from an importance (also called proposal) density $q(c_{0:k}|z_{1:k})$. Then the proper weights in (1) that lead to an approximation of the posterior are defined by :

$$w^i_k \propto \frac{\mathrm{p}(c^i_{0:k}|z_{1:k})}{q(c^i_{0:k}|z_{1:k})} \,. \tag{2}$$

The goal of the particle filtering algorithm is the recursive propagation of the samples and estimation of the associated weights as each measurement is received sequentially. After some calculus and using Bayes rule, we obtain the following recursive update equation [1, 6]:

$$w^i_k \propto w^i_{k-1} \frac{\mathrm{p}(z_k|c_{0:k}, z_{1:k-1})\mathrm{p}(c_k|c_{0:k-1}, z_{1:k-1})}{q(c_k|c_{0:k-1}, z_{1:k})} \,, \tag{3}$$

$$= w^i_{k-1} \, \mathrm{p}(z_k|c^i_k) \tag{4}$$

where Eq. 4 derives from three commonly made hypotheses :

H1 : The observations $\{z_k\}$, given the sequence of states, are independent. This leads to $\mathrm{p}(z_{1:k}|c_{0:k}) = \prod_{i=1}^{k} \mathrm{p}(z_k|c_k)$, which requires the definition of the individual data-likelihood $\mathrm{p}(z_k|c_k)$;

H2 : The state sequence $c_{0:k}$ follows a first-order Markov chain model, characterized by the definition of the dynamics $\mathrm{p}(c_k|c_{k-1})$.

H3 : The prior distribution $\mathrm{p}(x_{0:k})$ is employed as importance function. In this case, $q(c_k|c_{0:k-1}, z_{1:k}) = \mathrm{p}(c_k|c_{k-1})$.

It is known that importance sampling is usually inefficient in high-dimensionnal spaces [6], which is the case of the state space $c_{0:k}$ as k increases. To solve this problem, an additional resampling step is necessary, whose effect is to eliminate the particles with low importance weights and to multiply particles having high weights, giving rise to more variety around the modes of the posterior after the next importance sampling step. Altogether, we obtain the particle filter that is displayed in Fig. 2.

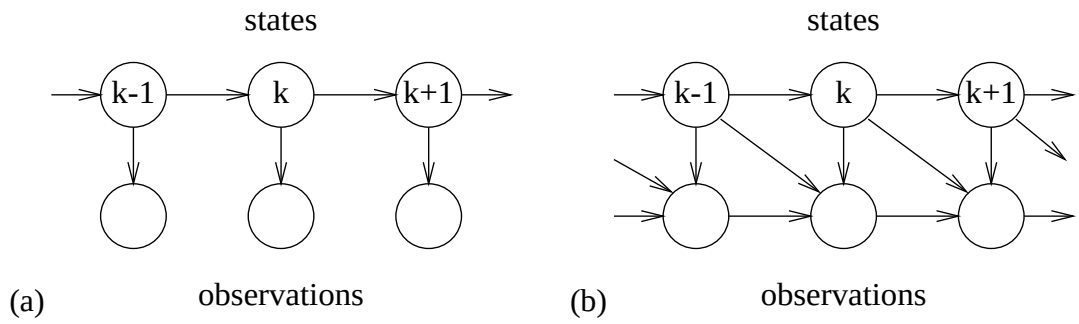

Figure 1: Graphical models for tracking. (a) standard model (b) proposed model.

1. Initialisation : For $i = 1, \ldots, N_s$, sample $c_0^i \sim \mathrm{p}(c_0)$ and set $k = 1$
2. Importance sampling step : For $i = 1, \ldots, N_s$, sample $\tilde{c}_k^i \sim q(c_k^i | c_{0:k-1}^i, z_{1:k})$ and evaluate the importance weights : $\tilde{w}_k^i$ using equations (3) or (4).
3. Selection step : Resample with replacement N_s particles $\{c_k^i, w_k^i = \frac{1}{N_s}\}$ from the sample set $\{\tilde{c}_k^i, \tilde{w}_k^i\}$. Set $k = k + 1$ and go to step 2

Figure 2: The particle filter algorithm.

3 Approach and motivations

In this Section, we propose a new method that embeds motion in the particle filter. This is first obtained by incorporating motion information into the measurement process. This can be achieved by modifying the traditional graphical model represented in Fig. 1a, by making the current observation dependent not only on the current object configuration but also on the object configuration and observation at the previous instant (see Fig. 1b). Secondly, we will propose to use explicit motion measurements in order to obtain a better proposal distribution. In this Section, we justify this approach.

3.1 Revisiting particle hypothesis

The filter described in Fig. 2 is based on the standard probabilistic model for tracking displayed in Fig. 1a and corresponding to hypotheses H1 and H2 of the previous section.

In visual tracking, hypothesis H1 may not be very accurate.[1] In most of the tracking algorithms, the configuration state includes the parameters of a geometric transformation $\mathcal{T}$. Then, the measurements consist of implicitly or explicitly extracting some part of the image by :

$$\tilde{z}_{c_k}(\mathbf{r}) = z_k(\mathcal{T}_{c_k}\mathbf{r}) \qquad \forall \mathbf{r} \in R\,, \tag{5}$$

where $\mathbf{r}$ denotes a pixel position, R denotes a fixed reference region, and $\mathcal{T}_{c_k}\mathbf{r}$ represents the

[1] For contour tracking, the assumption is quite valid as the temporal auto-correlation function of contours is peaked.

Figure 3: Images at time t and $t+3$. The two local patches corresponding to the head and extracted from the two images are strongly correlated.

application of the transform $\mathcal{T}$ parameterized by c_k to the pixel $\mathbf{r}$. The data likelihood is then computed from this local patch : $\mathrm{p}(z_k|c_k) = \mathrm{p}(\tilde{z}_{c_k})$, with $\tilde{z}_{c_k}$ denoting the patch casted in the reference frame according to (5). However, if c_{k-1} and c_k correspond to two consecutive states of a given object, it is reasonable to assume :

$$\tilde{z}_{c_k}(\mathbf{r}) = \tilde{z}_{c_{k-1}}(\mathbf{r}) + noise \qquad \forall \mathbf{r} \in R \tag{6}$$

where $noise$ usually takes some small value. This point is illustrated in Figure 3. Equation (6) is at the core of all motion estimation and compensation algorithms like MPEG and is indeed a valid hypothesis. Thus, according to this equation, the independence of the data given the sequence of states is not a valid assumption. More precisely :

$$\mathrm{p}(z_k|z_{1:k-1}, c_{1:k}) \neq \mathrm{p}(z_k|c_k) \tag{7}$$

which means that we can not reduce the left hand side to the right one as usually done. A better model for visual tracking is thus represented by the graphical model of Fig. 1b.

The new model can be incorporated in the particle framework. All calculus leading to Eq. 3 are general and do not depend on assumptions H1, H2 and H3. Starting from there, replacing H1 by the new model and keeping H2 and H3, it is easy to see that the new weight update equation is given by :

$$w_k^i \propto w_{k-1}^i \ \mathrm{p}(z_k|z_{k-1}, c_k^i, c_{k-1}^i) \tag{8}$$

in replacement of equation (4).

3.2 Proposal and dynamical model

Modeling the dynamics, i.e. the transition prior, of the state sequence is a very important step. However, in visual tracking, finding a good model is very difficult because of the low temporal sampling rate and the presence of fast and unexpected motions, due either to camera or object (human) movments. To illustrate this, let us consider the following simple dynamical model :

$$c_k = c_{k-1} + \dot{c}_{k-1} + w_k \tag{9}$$

where $\dot{c}$ denotes the state derivative and models the evolution of the state. As state, consider the horizontal position of the head of the sequence in Fig. 8. We manually ground-truthed the head position in 200 images of this sequence. Fig. 4a reports the prediction error w calculated using ground-truth data and obtained by estimating $\dot{c}$ with a simple auto-regressive model :

$$\dot{c}_{k-1} = c_{k-1} - c_{k-2} \tag{10}$$

As can be seen, this prediction is noisy. Furthermore, there are large peak errors (up to 30% of the head width). To cope with these peaks, the noise variance in the dynamics, used as proposal distribution, has to be set to a large enough value, with the downside that many particles are wasted in low likelihood areas, or spread on local distractors that can ultimately lead to tracking failure. On the other hand, exploiting the inter-frame motion to estimate $\dot{c}$ and predict the new state value (using the coefficient of a robustly estimated affine motion model, see Section 4.2) can lead to a reduction of both the noise variance and of the error peaks (Fig. 4b).

There is another advantage of using image-based motion estimates. Let us first note that the previous state values (here c_{k-1}, c_{k-2}) used to predict the new state value c_k are affected by noise, due to measurement errors and uncertainty. Thus, in the standard AR approach, both the state c_{k-1} and state derivative $\dot{c}_{k-1}$ in Eq. 9 are affected by this noise, resulting in large errors (Fig. 4c). When using the inter-frame motion estimates, the estimation of $\dot{c}$ is almost not affected by noise (whose effect is to slightly modify the support region used to estimate the motion), as illustrated in Fig. 5, resulting again in a lower noise variance process (Fig. 4d).

Thus, despite needing more computation resources, inter-frame motion estimates are usually more precise than auto-regressive models to predict new state values of geometric transformation parameters; as a consequence, they are a better choice when designing a proposal function. This observation is supported by experiments on other parameters -vertical position, scale- and on other sequences.

4 The proposed model

In this Section, we describe more precisely the implementation of our method.

4.1 Object representation and state space

We follow an image-based standard approach, where the object is represented by a region R subject to some valid geometric transformation, and is characterized by a shape. For geometric transformations, we have chosen a subspace of the affine transformations comprising a translation $\mathbf{T}$, a scaling factor s, and an aspect ratio e :

$$\mathcal{T}_\alpha \mathbf{r} = \begin{pmatrix} \mathbf{T}_x + x s_x \\ \mathbf{T}_y + y s_y \end{pmatrix} \ , \tag{11}$$

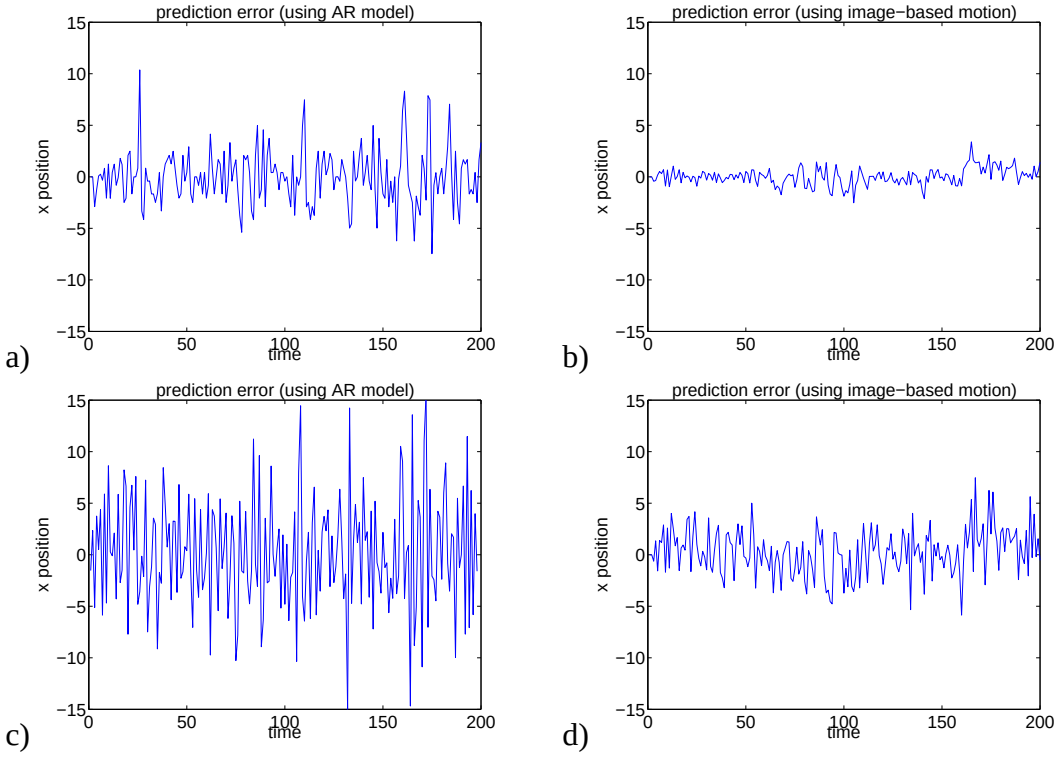

Figure 4: a) Prediction error of the x position, when using an AR2 model (σ_w=2.7). b) Prediction error, but exploiting the inter-frame motion estimation (σ_w=0.83). c) and d), same as a) b) but now adding a random gaussian noise (stdev=2 pixels) on the x measurements used for the prediction. In the AR2 model (Fig. c) both the previous state and state derivative estimates are affected by noise (σ_w=5.6), while with visual-motion (Fig. d) the noise only affects the previous measurement (σ_w=2.3).

where $\mathbf{r} = (x, y)$ denotes a point position in the reference frame, $\alpha = (\mathbf{T}, s, e)$, and :

$$\left\{ \begin{array}{l} s = \frac{s_x + s_y}{2} \\ e = \frac{s_x}{s_y} \end{array} \right. \text{ and } \left\{ \begin{array}{l} s_x = \frac{2es}{1+e} \\ s_y = \frac{2s}{1+e} \end{array} \right. \tag{12}$$

A state is then defined as $c_k = (\alpha_k, \alpha_{k-1})$.

4.2 Proposal distribution

As mentionned in the previous Section, we use inter-frame motion estimates to predict the new state values. More precisely, an affine displacement model $\vec{d}_\Theta$ parameterized by $\Theta = (a_i), i = 1..6$ is computed using a gradient-based robust estimation method described in [10][2]. $\vec{d}_\Theta$ is defined by:

$$\vec{d}_\Theta(\mathbf{r}) = \left(\begin{array}{l} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{array} \right) \text{ , } \mathbf{r} = (x, y) \text{ , } \tag{13}$$

[2]We use the code available at http://www.irisa.fr/vista

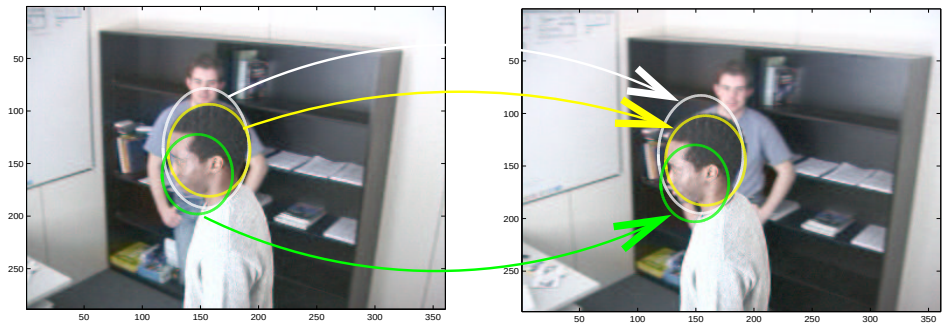

Figure 5: Example of motion estimates between two images from noisy states. The 3 ellipses correspond to different state values. Although the estimation support regions only cover part of the head and enclose textured background, the head motion estimate is still good.

This method takes advantage of a multiresolution framework and an incremental scheme based on the Gauss-Newton method. It minimizes an M-estimator criterion to ensure the goal of robustness, as follows :

$$\widehat{\Theta}(c_k) = \underset{\Theta}{\operatorname{argmin}} \sum_{\mathbf{r}\in R(c_k)} \rho\left(\mathrm{DFD}_\Theta(\mathbf{r})\right)$$

$$\textit{with} \quad \mathrm{DFD}_\Theta(\mathbf{r}) = z_{k+1}(\mathbf{r} + \vec{d}_\Theta(\mathbf{r})) - z_k(\mathbf{r}), \tag{14}$$

where z_k and z_{k+1} are the images, and $\rho(x)$ is a robust estimator (bounded for high values of x). Owing to the robustness of the estimator, an imprecise region definition $R(c_k)$ due to a noisy state value does not sensibly affect the estimation (see Fig. 5). Moreover, the algorithm delivers the covariance matrix of the affine parameters. From these estimates, we can construct an estimate $\widehat{\dot{\alpha}}_k$ of the variation of the coefficients between the two instant, with their variance $\widehat{\dot{\Lambda}}_k$. For instance, assuming that coordinates in Eq. 13 are expressed with respect to the current object center (located at **T** in the image), we have for the derivative estimates :

$$\left\{ \begin{array}{l} \dot{\mathbf{T}}_x = a_1 \\ \dot{\mathbf{T}}_y = a_4 \end{array} \right. \text{ and } \left\{ \begin{array}{l} \dot{s}_x = a_2 s_x \\ \dot{s}_y = a_6 s_y \end{array} \right. \text{ and } \left\{ \begin{array}{l} \dot{s} = \frac{s}{1+e}(a_2 e + a_6) \\ \dot{e} = e(a_2 - a_6) \end{array} \right. \tag{15}$$

Denoting the predicted value $\widehat{\alpha}_{k+1} = \alpha_k + \widehat{\dot{\alpha}}_k$, and assuming the noise on the estimate $\widehat{\dot{\alpha}}_k$ independent of the noise process w (see Eq. 9), we define the proposal distribution to be used in Equation 3 as :

$$q(c_{k+1}|c_{0:k}, z_{1:k+1}) \propto \mathcal{N}(\alpha_{k+1}; \widehat{\alpha}_{k+1}, \widehat{\Lambda}_{k+1}) \tag{16}$$

where $\mathcal{N}(.; \mu, \Lambda)$ represents a gaussian distribution with mean μ and Λ variance, and $\widehat{\Lambda}_{k+1} = \widehat{\dot{\Lambda}}_k + \Lambda_{w_p}$, Λ_{w_p} being the variance of the process noise w_p.

4.3 Dynamics definition

To model the prior, we use a standard second order auto-regressive model (cf Eq. 10) for each of the components of α. However, to account for outliers (i.e. unexpected and abrupt changes) and reduce the sensitivity of the prior in the tail, we model the noise process with a Cauchy distribution $\rho_c(x, \sigma^2) = \frac{\sigma}{\pi(x^2+\sigma^2)}$. This leads to :

$$\mathrm{p}(c_{k+1}|c_k) = \prod_{j=1}^{4} \rho_c \left(\alpha_{k+1,j} - (2\alpha_{k,j} - \alpha_{k-1,j}), \sigma^2_{w_{d,j}}\right) . \tag{17}$$

where $\sigma^2_{w_{d,j}}$ denotes the dynamics noise variance of the j^{th} component.

4.4 Data likelihood modeling

To implement the new particle filter, we considered the following data likelihood :

$$\mathrm{p}(z_k|z_{k-1}, c_k, c_{k-1}) = \mathrm{p}_c(z_k|z_{k-1}, c_k, c_{k-1}) \times \mathrm{p}_o(z_k|c_k) \tag{18}$$

where the first probability $\mathrm{p}_c()$ models the correlation between the two observations and $\mathrm{p}_o()$ is an object likelihood. This choice decouples the model of the dependency existing between two images, whose implicit goal is to ensure that the object trajectory follows the optical flow field implied by the sequence of images, from the shape or appearence object model. We assumed that these two terms are independent. When the object is modeled by a shape, this assumption is valid since shape measurement will mainly involve measurements on the border of the object, while the correlation term will apply to the regions inside the object.

Object shape observation model

The observation model assumes that objects are embedded in clutter. Edge-based measurements are computed along L normal lines to a hypothesized contour, resulting for each line l in a vector of candidate positions $\{\nu^l_m\}$ relative to a point lying on the contour ν^l_0. With some usual assumptions [4], the shape likelihood $\mathrm{p}_o(z_k|c_k) = \mathrm{p}_{sh}(z_k|c_k)$ can be expressed as

$$\mathrm{p}_{sh}(z_k|c_k) \propto \prod_{l=1}^{L} \max\left(K, \exp(-\frac{\|\hat{\nu}^l_m - \nu^l_0\|^2}{2\sigma^2})\right), \tag{19}$$

where $\hat{\nu}^l_m$ is the nearest edge on l, and K is a constant used when no edges are detected.

Image correlation measurement

We model this term in the following way :

$$\mathrm{p}_c(z_k|z_{k-1}, c_k, c_{k-1}) \propto \mathrm{p}_{c1}(\widehat{\alpha}_k, \alpha_k)\mathrm{p}_{c2}(\tilde{z}_{c_k}, \tilde{z}_{c_{k-1}}) \tag{20}$$

with :

$$\mathrm{p}_{c1}(\widehat{\alpha}_k, \alpha_k) \propto \mathcal{N}(\widehat{\alpha}_k; \alpha_k, \widehat{\Lambda}_k) \quad (21)$$

$$\mathrm{p}_{c2}(\tilde{z}_{c_k}, \tilde{z}_{c_{k-1}}) \propto \exp^{-\lambda_c \mathrm{d}_c(\tilde{z}_{c_k}, \tilde{z}_{c_{k-1}})} \quad (22)$$

where d_c denotes a distance between two image patches. The first probability term in this expression compares the parameter values predicted using the estimated motion with the sampled values. This term assumes a Gaussian noise process in parameter space. This assumption, however, is only valid around the predicted value. Thus, to introduce a non-Gaussian modeling, we use a second term that compares directly the patches around c_k and c_{k-1} using the similarity distance d_c. Its purpose can be illustrated using Fig. 5. While all the three predicted configurations will be weighted equally from p_{c1} (assuming their estimated variance are approximately the same), the second term p_{c2} will downweight the two predictions whose corresponding support region is covering part of the background which is undergoing a different motion than the head.

The definition of p_{c2} requires the specification of a patch distance. Many such distances have been defined and used in the literature [15, 17]. The choice of the distance should take into account the followings considerations :

1. the distance should still model the underlying motion content, i.e. the distance should increase as the error in the predicted configuration grows;

2. the random nature of the prediction process in the SMC filtering will rarely produce configurations corresponding to exact matches(this is particularly true when using a small number of samples);

3. particles covering both background and object undergoing different motion should have a low likelihood.

For these purposes, we found out that it was preferable not to use robust norms such as L1 saturated distance or a Haussdorf distance [17]. Additionnaly, we needed to avoid distances which might *a priori* favor patches with specific contents. This is the case for instance of the L2 distance (which corresponds to an additive Gaussian noise model in Eq.(6)), which will generally provide lower scores for patches with large uniform areas. Thus, to avoid this effect, we used the normalized-cross correlation coefficient defined as :

$$\mathrm{d}_c(\tilde{z}_1, \tilde{z}_2) = \frac{\sum_{\mathbf{r} \in R} (\tilde{z}_1(\mathbf{r}) - \bar{\tilde{z}}_1) \cdot (\tilde{z}_2(\mathbf{r}) - \bar{\tilde{z}}_2)}{\sqrt{\mathrm{Var}(\tilde{z}_1)} \sqrt{\mathrm{Var}(\tilde{z}_2)}} \quad (23)$$

where $\bar{\tilde{z}}_1$ represents the mean of $\tilde{z}_1$. Regarding the above equation, it is important to again emphasize that the method is not performing template matching, as in [15]. No object template is learned off-line or defined at the begining of the sequence, and the tracker does not maintain a single template object representation at each instant of the sequence. Thus, the correlation term is not object specific (except through the definition of the reference region R). A particle "lying" on the background would thus receive a high weight if the predicted

motion is in adequation with background motion. Nevertheless, the methodology could be the extended to be more object dependent, by allowing the region R to vary over time (using examplars for instance).

5 Results

To illustrate the method, we have considered three sequences involving head tracking. To differentiate the different elements of the model, we have considered 3 configurations :

- shape tracker M1 : this tracker corresponds to the standard CONDENSATION algorithm [4], with the shape likelihood combined with the same AR model with Gaussian noise for the proposal and the prior.
- shape+implicit correlation tracker M2 : it corresponds to CONDENSATION, with the addition of the implicit motion likelihood term in the likelihood evaluation (i.e now equal to $\mathrm{p}_{sh}.\mathrm{p}_{c2}$). This method does not use explicit motion measurements.
- motion proposal tracker M3 : it is the full model. The samples are drawn from the motion proposal, Eq. 16, and the weight update is performed using Eq. 3. After simplification, he update equation becomes :

$$w_k^i = w_{k-1}^i \ \mathrm{p}_{sh}(z_k|c_k)\mathrm{p}_{c2}(\tilde{z}_{c_k}, \tilde{z}_{c_{k-1}})\mathrm{p}(c_k|c_{k-1}) \tag{24}$$

  For this model, the motion estimation is not performed for all particles since it is robust to variations of the support region. At each time, the particles are clustered into K clusters. The motion is estimated using the mean of each cluster and exploited for all the particles of the cluster. Currently we use max(20,N_s/10) clusters.

Currently, for 200 particles, the shape tracker runs in real time (on a 2.5GHz P IV machine), the shape+implicit correlation at around 20 image/s, and the full model at around 4 image/s. In all experiments, all the common parameters are kept identical.

The first sequence (Fig. 6) illustrates the benefit of the implicit method in the presence of ambiguities. Despite the presence of a highly textured background producing very noisy shape measurements, the camera and head motion, the change of appearence of the head, and partial occlusion, the head is correcly tracked using our methods (on all runs using different seeds). Whatever the number of particles or the noise variance in the dynamical model, the shape tracker alone is unable to perform a correct tracking after time t_{12}.

The second sequence is a 12 s sequence of 330 frames (Fig. 7) extracted from a hand-held home video. Table 1 reports the tracking performance of the three trackers for different dynamics and sampling rates (all other parameters are left unchanged). A tracking failure is considered when the tracker looses the head and locks on another part of the image. As can be seen, while CONDENSATION performs quite well for tuned dynamics (D1), it breaks down rapidly, even for slight increases of dynamics variances (D2 to D4). Fig. 7 illustrates a typical failure due to the small size of the head at the begining of the sequence, the low

Figure 6: Head tracking 1 : 2 first rows : shape tracker only (CONDENSATION). 2 last rows : shape+implicit correlation (N_s=100). Same performance with the motion proposal tracker.

contrast at the left of the head, and the clutter. On the other hand, the implicit tracker M2 performs well under almost all circumstances, showing its robustness against clutter, partial measurements (around time t_{250} and partial occlusion (end of the sequence). Only when the number of samples is low (100 in S2) does the tracker fail. These failures are occuring at different parts of the sequence. Finally, in all experiments, the M3 tracker produces a correct tracking rate equal to 98%, even with a small number of samples, up to the partial occlusion. At this part of the sequence, as the occlusion reaches 50% of the tracked head, the motion estimation sometimes lock onto the woman's head motion, leading to the reported tracker failures.

The last sequence (Fig. 8) illustrates more clearly the benefit of using the motion proposal. This 24s sequence acquired at 12 frame/s is specially difficult because of the occurence of several head turns[3] and abrupt motion changes (translations, zooms in and out), the large variations of scale, and importantly, the absence of head contours as the head moves

[3]The head turn is indeed a difficult case for the new method, as in the extreme case, the motion inside the head region indicates a right (or left) movement while the head outline remains static.

Tracker	D1	D2	D3	D4	S1	S2
CONDENSATION	88	36	2	0	0	0
M2 (Implicit)	100	98	100	94	90	50
M3 (with proposal)	70	82	92	90	96	80

Table 1: Successful tracking rate (in %, out of 50 trials with different seeds) with different dynamics and sampling. Experiments D1 to D4 correspond to N_s=500, with dynamics D1 (2,0.01), D2 (3,0.01), D3(5,0.01) D4(8,0.02), the 1st (resp. 2nd) number corresponds to the dynamics and proposal noise standard deviation of the $\mathbf{T}$ (resp. s) state component. Experiments S1 and S2 use a (5,0.01) dynamics, with 250 (S1) and 100 (S2) samples.

in front of the bookshelves. Because of these, CONDENSATION is again lost very quickly. On the other hand, the M2 tracker successfuly tracks the head at the beginning, but usually gets lost when the person moves in front of the bookshelves (around frames t_{130}-t_{145}), due to the lack of contour measurements coupled with a large zooming effect. This latter problem is resolved by the motion proposal, which better capture the state variations, and allows a successful track of the head until the end of the sequence (time t_{340}).

6 Conclusion

We presented a methodology to embed data-driven motion into particle filters. This was first achieved by introducing a likelihood term that models the temporal correlation existing between successive images of the same object. This term models the visual motion in an implicit way. Secondly, explicit motion estimates were exploited to predict more precisely the new state values. This data-driven approach allows for designing better proposals that take into account the new image. Altogether, the algorithm allows to better handle unexpected and fast motion changes, to remove tracking ambiguities that arise when using generic shape-based or color-based object models, and to reduce the sensitivity to the different parameters of the prior model. The method is general and could be used to track deformable objects by integrating motion measurements along the shape curve, as described in [9].

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian. *IEEE Trans. Signal Processing*, pages 100–107, 2001.

[2] M. J. Black and A. D. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *European Conf. on Computer Vision*, pages 909–924, Freiburg, Germany, 1998.

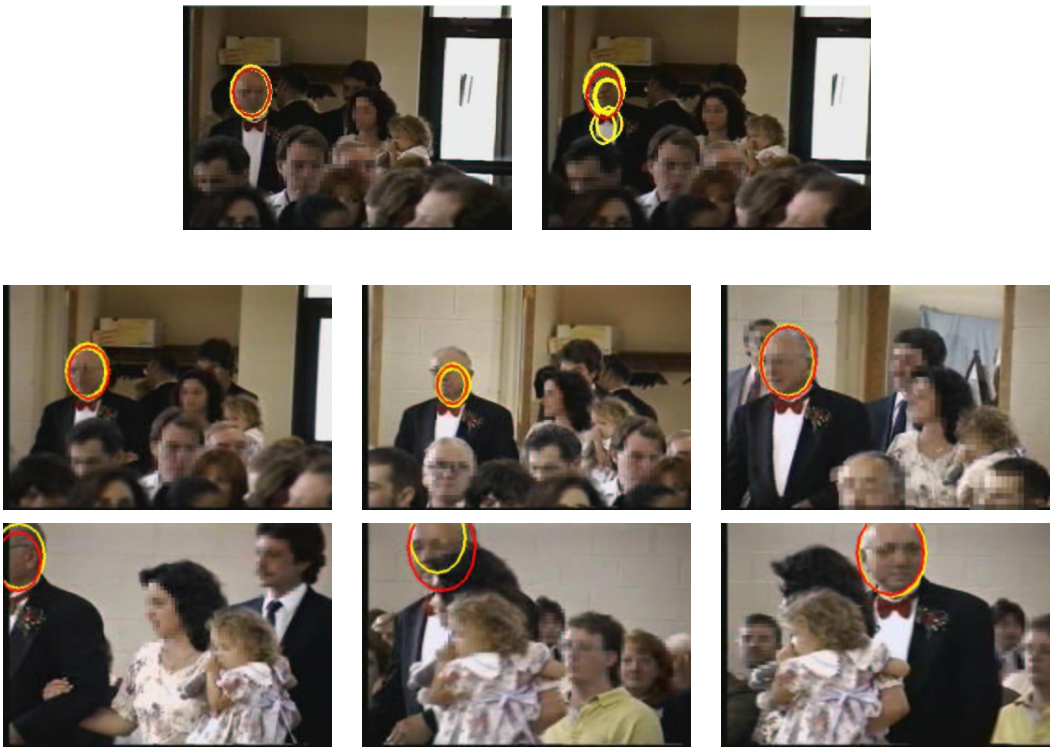

Figure 7: Head tracking 2 : top row : shape tracker only (CONDENSATION) at initial time t_{861} and t_{868} (N_s=500). After a few frame, the tracker diverge. Two last rows : shape+implicit correlation (N_s=200) at time t_{880}, t_{920}, t_{1025}, t_{1110}, t_{1155}, t_{1165}. In red, mean shape. In yellow, highly likely particles. Same performance with the motion proposal tracker.

[3] D. Gatica-Perez, G. Lathoud, I. McCowan and J.-M. Odobez A Mixed-State I-Particle Filter for Multi-Camera Speaker Tracking In *IEEE WOMTEC*, Nice, France, 2003.

[4] Andrew Blake and Michael Isard. *Active Contours*. Springer, 1998.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR*, pp 142–151, 2000.

[6] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[7] M. Isard and A. Blake. ICONDENSATION : Unifying low-level and high-level tracking in a stochastic framework In *5th ECCV*, pp 893-908, 1998.

[8] M. Isard and A. Blake. A mixed-state CONDENSATION tracker with automatic model-switching. In *ICCV*, pp 107–112, 1998.

[9] C. Kervrann, F. Heitz and P. Pérez Statistical model-based estimation and tracking of non-rigid motion In *13th Int. Conf. Pattern Recognition*, pp 244-248, 1996.

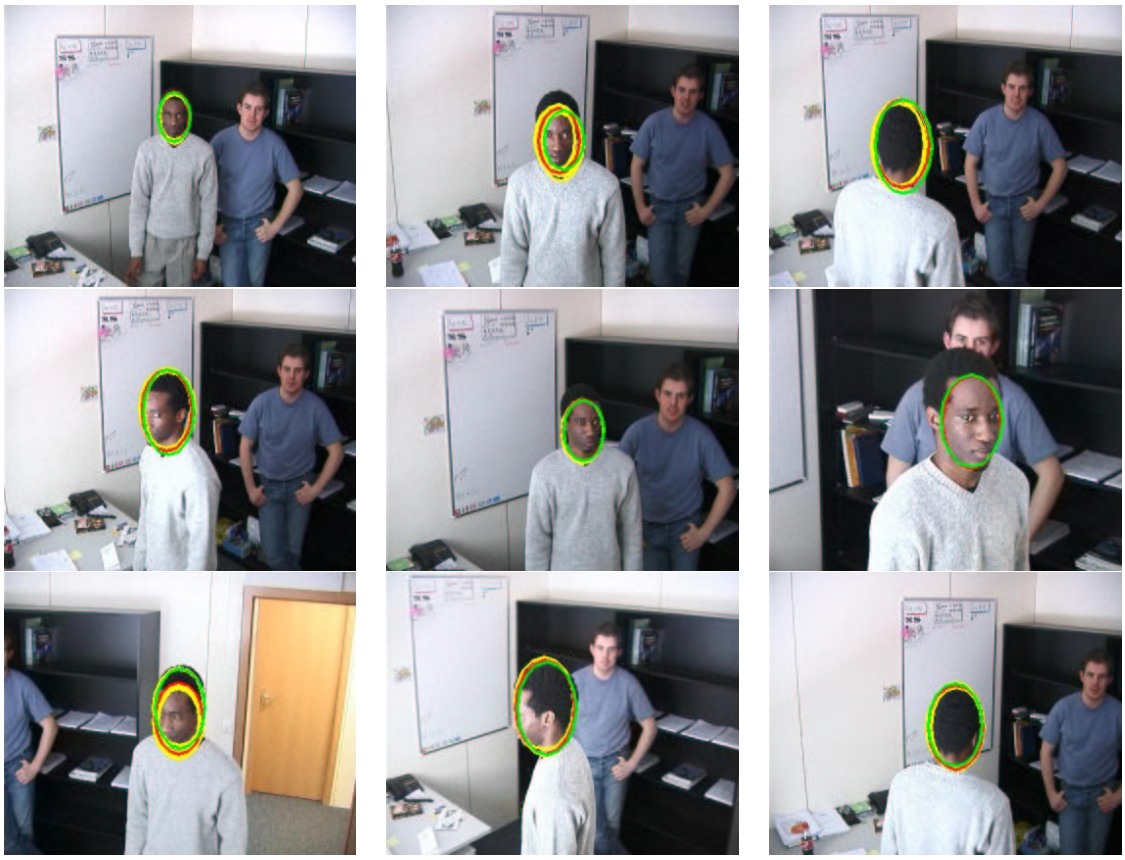

Figure 8: Tracker with motion proposal (N_s=1000) at time t_2, t_{40}, t_{85}, t_{100}, t_{130}, t_{145}, t_{170}, t_{195}, and t_{210} . In red, mean shape; in green, mode shape; in yellow, likely particles.

[10] J.-M. Odobez and P. Bouthemy Robust multiresolution estimation of parametric motion models In *Jl of Visual Com. and Image Representation*, vol 6, num. 4, pp 348-365, 1995.

[11] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Eur. Conf. on Computer Vision, ECCV'2002, LNCS 2350*, pp 661–675, June 2002.

[12] M.K. Pitt and N. Shephard. Filtering via Simulation: Auxiliary Particle Filters In *Journal of the American Statistical Association*, pp 590–599, vol. 94, num. 446, 1999.

[13] Y. Raja, S. McKenna, and S. Gong. Colour model selection and adaptation in dynamic scenes. In *5th European Conference on Computer Vision*, pp 460–474, 1998.

[14] Y. Rui and Y. Chen. Better proposal distribution: object tracking using unscented particle filter In *CVPR*, pp 486–793, dec. 2001.

[15] J. Sullivan and Rittscher J. Guiding random particles by deterministic search. In *ICCV*, pp 323–330, 2001.

[16] Hai Tao, Harpreet S. Sawhney, and Rakesh Kumar. Object tracking with bayesian estimation of dynamic layer representations. *IEEE PAMI*, 24(1):75–89, 2001.

[17] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Proc.* 8^{th} *Int. Conf. Computer Vision*, 2001.

[18] Y. Wu and T. Huang. A co-inference approach for robust visual tracking. In *Proc.* 8^{th} *Int. Conf. Computer Vision*, 2001.