IDIAP RESEARCH REPORT

MUCATAR - DELIVERABLE D4 : MULTI-OBJECT, MULTI-CAMERA TRACKING AND ACTIVITY RECOGNITION

Kevin Smith [1] Sileye Ba [1]
Jean-Marc Odobez [1] Daniel Gatica-Perez [1]

IDIAP–RR

JULY 2004

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

[1] IDIAP, Martigny, Switzerland

IDIAP Research Report

MUCATAR - DELIVERABLE D4 : MULTI-OBJECT, MULTI-CAMERA TRACKING AND ACTIVITY RECOGNITION

Kevin Smith Sileye Ba Jean-Marc Odobez Daniel Gatica-Perez

JULY 2004

Abstract. This document describes the progress on the MUCATAR (MUltiple CAmera Tracking and Activity Recognition) IM2 White Paper Project during its second year. Building on the first year achievments on single-object tracking, the research during the second year moved into two main directions: 1) the investigation of new sampling strategies to improve tracking with particle filters, both for single and multi-object tracking, and 2) the development of mixed-state particle filters for the tasks of joint head tracking and recognition of head poses, and multi-camera multi-modal speaker tracking. Details and results for the developed aproaches are presented and discussed.

1 Introduction

This document describes the progress on the MUCATAR (MUltiple CAmera Tracking and Activity Recognition) IM2 White Paper Project during its second year. Building on achievements of the first year in robust single-object tracking, the research during the second year pursues tracking and activity recognition goals by following two main research directions. The first direction investigates new sampling strategies to improve both single and multi-object tracking with particle filters. Tracking multiple objects is computationally expensive, but improved sampling strategies can reduce the computational cost considerably. The second direction involves the development of particle filters based on mixed-state models which combine discrete and continuous components in a state space representation. Mixed-state representations provide explicit mechanisms to deal with switching models, for instance to jointly perform tracking and recognition, where the (continuous) spatial configuration and the (discrete) activity are represented in the same state model. This framework has been applied to the problems of joint head tracking and head pose estimation, and of multi-camera multi-modal speaker tracking. All of the tasks have been evaluated using data from the multi-sensor meeting room at IDIAP. Brief summaries of the results of these methodologies are presented in this report. The reader is referred to the relevant publications [1, 2, 3, 4] for further details, and to a website www.idiap.ch/mucatar/ where videos with results are available.

The rest of this document is organized as follows. For sake of completeness, Section 2 briefly presents the particle filter formalism. Sections 3 and 4 describe our work on sampling strategies, and Sections 5 and 6 present our work on mixed-state particle filters. In particular, Section 3 describes the embedding of motion estimation between images into the particle filter framework [1]. Section 4 describes a sampling strategy for reducing computational complexity for multiple object tracking [2]. Section 5 describes a probabilistic framework to jointly track and estimate the head pose fusing texture and color features [3]. Section 6 presents our work regarding data fusion for tracking, using both multiple visual features and audio-visual features [4]. We conclude the document, and discuss future work in Section 7 .

2 Tracking with Particle Filters

Sequential Monte Carlo methods or particle filters (PFs) represent a principled methodology for tracking [8, 10]. Given a discriminative object representation and a Markov state-space model, with hidden states $\{\mathbf{x}_t\}$ that represent scene configurations, and observations $\{\mathbf{y}_t\}$ extracted from one or more data streams, a PF recursively approximates the filtering distribution of states given observations $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ by

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

where $\mathbf{y}_{1:t} = \{\mathbf{y}_1, ..., \mathbf{y}_t\}$. The integral in Eq. 1 represents the prediction step, in which the dynamical model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the previous distribution $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ are used to compute a prediction distribution, which is then used as prior for the update step, and multiplied by the likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ to generate the current filtering distribution. Except for a few special cases, exact inference in this model is intractable. SMC methods are usually employed to approximate Eq. 1 for non-linear, non-Gaussian problems, using random sampling by (i) predicting candidate configurations, and (ii) measuring their likelihood, in a process that amounts to random search in a configuration space. The filtering distribution is first defined by a set of weighted samples or particles $\{(\mathbf{x}_t^{(i)}, \pi_t^{(i)}), i = 1, ..., N\}$, where $\mathbf{x}_t^{(i)}$ and $\pi_t^{(i)}$ denote the i-th sample and its importance weight at the current time. The point-mass approximation is given by $\hat{p}_N(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} \pi_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)})$. The prediction step propagates each particle by sampling from a proposal distribution $q(\cdot)$ (the dynamics in the simplest case), and the updating step reweights each particle using the likelihood. In the simplest case, $\pi_t^{(i)} \propto \pi_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$. A resampling step using the new weights is necessary to avoid degradation of the particle set.

The above methodology is general, allowing for a multitude of variants: tracking with multiple sources of observations, tracking in mixed-state configuration spaces composed of continuous and discrete components, or tracking multiple objects can be all defined by the same formulation. Designing a PF therefore involves making choices for specific state-spaces, object representations, sampling mechanisms, dynamical models,

and observation models. MUCATAR has investigated several of these issues, as discussed in the following subsections.

To facilitate reading, each remaining section is organized into identical subsections. The first subsection motivates the specific investigated problem. The second subsection presents our approach to the problem, and is divided into four sub-subsections describing the work in detail (object model, dynamic model, sampling strategy, and observation model). The last subsection describes the experimental setup and discusses results.

3 Improving tracking efficiency with motion proposal

3.1 Motivation

Particle filters have shown to be a successful approach for robust tracking. Among the elements that need to be defined while tracking with particle filters are the dynamical model and the proposal function [18]. The dynamical model characterizes the prior on the object state sequence. A common assumption in particle filtering approaches is to use the dynamic as proposal distribution (the function predicting new state hypotheses). However, this assumption raises some difficulties in the modeling since the proposal should fulfill two contradictory aims. On one hand as a prior, the dynamics should be tight enough to avoid the tracker to be confused by distractor. On the other hand, it should to be broad to cope with abrupt motion changes due to camera motion or object movements. To address this issue we propose a new particle filter method which incorporates explicit inter-frame motion estimates in the proposal distribution to predict new state values. This method improves the sampling efficiency by handling abrupt motion.

3.2 Our Approach

Object Model. An object is represented by a region R subject to some valid geometric transformation, and is characterized by a shape. The chosen transformation comprises a translation $\mathbf{T}$, a scaling factor s, and an aspect ratio e. A state is defined as $\mathbf{x}_t = (\alpha_t \alpha_{t-1})$ where $\alpha = (\mathbf{T}, s, e)$.

Sampling Strategy. We use inter-frame motion estimates to predict the new state values. More precisely, an affine displacement model is estimated using a gradient-based robust and multi-resolution estimation method [17]. From these estimates, we can easily construct an estimate $\widehat{\alpha}_t$ of the variation of the coefficients between the two instants. Denoting the predicted value $\widehat{\alpha}_{t+1} = \alpha_t + \widehat{\alpha}_t$, we define the proposal distribution as :

$$q(\mathbf{x}_{t+1}|\mathbf{x}_{0:t}, \mathbf{y}_{1:t+1}) \propto \mathcal{N}(\alpha_{t+1}; \widehat{\alpha}_{t+1}, \Lambda_{t+1}) \tag{2}$$

where $\mathcal{N}(.; \mu, \Lambda)$ represents a Gaussian distribution with mean μ and variance Λ.

Dynamic Model. We use a standard second order AR model for each of the components of α. However, to account for outliers and reduce the sensitivity of the prior in the tail, we model the noise process with a Cauchy distribution $\rho_c(x, \sigma^2) = \frac{\sigma}{\pi(x^2+\sigma^2)}$.

Observation Model. We modeled the data likelihood as :

$$\mathrm{p}(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t-1}) = \mathrm{p}_{sh}(\mathbf{y}_t^s|\mathbf{x}_t)\mathrm{p}_c(\mathbf{y}_t^g|\mathbf{y}_{t-1}^g, \mathbf{x}_t, \mathbf{x}_{t-1}) \,, \tag{3}$$

where $\mathbf{y}_t = (\mathbf{y}_t^s, \mathbf{y}_t^g)$ and $\mathbf{y}_t^s$ (resp. $\mathbf{y}_t^g$) denotes the shape (resp. the gray-level) measurements, and where p_c models the correlation between the two observations and p_{sh} is a shape likelihood. This choice decouples the model of the dependency between two images, whose implicit goal is to ensure that the object trajectory follows the optical flow field from the object shape model. We assumed that these two terms are independent [18].

• *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 the nearest edge position $\{\hat{\nu}_m^l\}$ relative to a point lying on the contour ν_0^l. With some usual assumptions [8], the shape likelihood $\mathrm{p}_{sh}(\mathbf{y}_t|\mathbf{x}_t)$ can be expressed as

$$\mathrm{p}_{sh}(\mathbf{y}_t|\mathbf{x}_t) \propto \prod_{l=1}^{L} \max\left(K, \exp(-\frac{\|\hat{\nu}_m^l - \nu_0^l\|^2}{2\sigma^2})\right), \tag{4}$$

where K is a constant used when no edges are detected.

• *Image correlation measurement.* We model this term as

$$\mathrm{p}_c(\mathbf{y}_t^g|\mathbf{y}_{t-1}^g, \mathbf{x}_t, \mathbf{x}_{t-1}) \propto \mathrm{p}_{c1}(\widehat{\alpha}_t, \alpha_t)\mathrm{p}_{c2}(\tilde{\mathbf{y}}_{\mathbf{x}_t}^g, \tilde{\mathbf{y}}_{\mathbf{x}_{t-1}}^g) \tag{5}$$

$$\text{with} \qquad \mathrm{p}_{c1}(\widehat{\alpha}_t, \alpha_t) \propto \mathcal{N}(\widehat{\alpha}_t; \alpha_t, \widehat{\Lambda}_t) \tag{6}$$

$$\mathrm{p}_{c2}(\tilde{\mathbf{y}}_{\mathbf{x}_t}^g, \tilde{\mathbf{y}}_{\mathbf{x}_{t-1}}^g) \propto \exp^{-\lambda_c \mathrm{d}_c(\tilde{\mathbf{y}}_{\mathbf{x}_t}^g, \tilde{\mathbf{y}}_{\mathbf{x}_{t-1}}^g)} \tag{7}$$

where d_c denotes a distance between two image patches and $\tilde{\mathbf{y}}_{\mathbf{x}_t}^g$ denotes the patch image casted in a reference frame according to $\mathbf{x}_t$. The first pdf 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. To introduce a non-Gaussian modeling, we use a second term that compares directly the patches around $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. The definition of p_{c2} requires the specification of a patch distance. We use the normalized-cross correlation coefficient defined as :

$$N_c(\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2) = \frac{\sum_{\mathbf{r}\in R}\left(\tilde{\mathbf{y}}_1(\mathbf{r}) - \bar{\tilde{\mathbf{y}}}_1\right)\cdot\left(\tilde{\mathbf{y}}_2(\mathbf{r}) - \bar{\tilde{\mathbf{y}}}_2\right)}{\sqrt{\mathrm{Var}(\tilde{\mathbf{y}}_1)}\sqrt{\mathrm{Var}(\tilde{\mathbf{y}}_2)}} \tag{8}$$

where $\bar{\tilde{\mathbf{y}}}_1$ represents the mean of $\tilde{\mathbf{y}}_1$. The distance between the patch images is defined to be $\mathrm{d}_c = 1 - Nc$.

Tracker	D1	D2	D3	D4	S1	S2
N_s	500				200	100
$\sigma_{\mathbf{T}}$	2	3	5	8	5	
σ_s	0.01			0.02	0.01	
CONDENS.	88	36	2	0	0	0
M2 (Implicit)	100	98	100	94	90	50
M3 (see text)	70	82	92	90	96	80

Table 1: Successful tracking rate (in %, out of 50 trials with different seeds). $\sigma_{\mathbf{T}}$ (resp. σ_s) denotes the dynamics and proposal noise standard deviation of the $\mathbf{T}$ (resp. s) state components.

3.3 Results

To illustrate the method, we consider two sequences involving head tracking. Three configurations of the tracker are considered. The first model (M1) is CONDENSATION [8], which corresponds to the shape likelihood combined with the same AR model with Gaussian noise for the proposal and the prior. The second model (M2) 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. The third model (M3) is the full model. 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. For 200 particles, the M1 tracker runs in real time (on a 2.5GHz machine), M2 at 20 image/s, and M3 at around 4 image/s.

The first example is a 12 s sequence of 330 frames (Fig. 1) extracted from a hand-held home video. Table 1 reports the tracking performance of the three trackers for different dynamics and sampling rates. 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. 1 illustrates a typical failure due to the small size of the head at the beginning of the sequence, the low contrast 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 occurring at different parts of the sequence. Finally, in

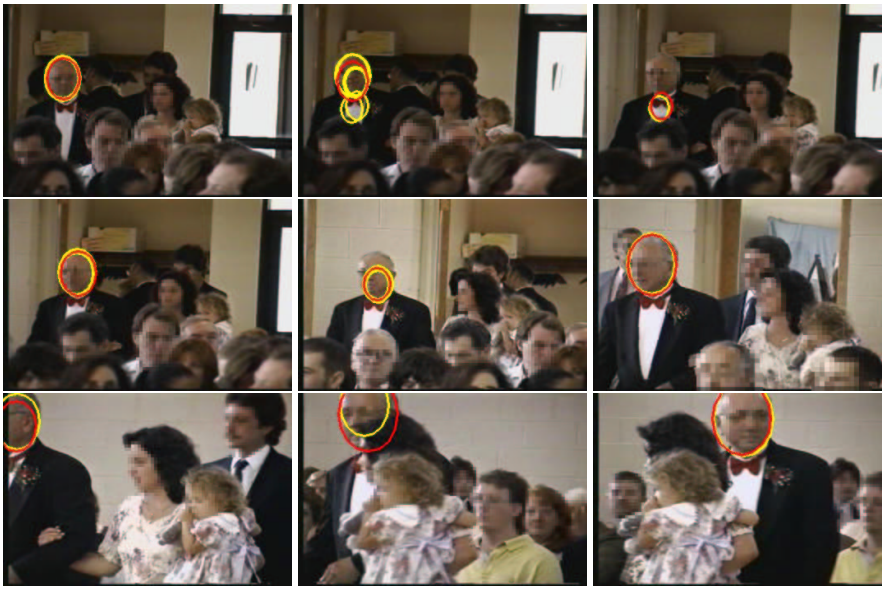

Figure 1: top row : CONDENSATION at time t_1, t_8 and t_{15} (N_s=500). center and bottom : M2 tracker (N_s=200) at time $t_{20}, t_{60}, t_{165}, t_{250}, t_{295}, t_{305}$. In red, mean shape. In yellow, highly likely particles.

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. Table 1 shows correct tracking rates for the methods depending for specified tracking parameters.

The second sequence (Fig. 2) illustrates more clearly the benefit of using the motion proposal. This 24s sequence acquired at 12 frame/s is specially difficult because of the occurrence of several head turns and abrupt motion changes (translations, zooms) the large variations of scale, and importantly, the absence of head contours as the head moves in front of the bookshelves. Because of these, CONDENSATION is again lost very quickly. On the other hand, the M2 tracker successfully 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}).

4 Sampling strategies for Multi-Object Tracking

4.1 Motivation

Multi-Object tracking (MOT) [12, 11] is an important problem in a number of vision applications. For particle filter (PF) tracking, as the number of objects tracked increases, the search space for random sampling explodes in dimension. Partitioned sampling (PS) [12] solves this problem by partitioning the parameter set, then searching each partition sequentially. However, sequential weighted resampling steps cause an impoverishment effect that increases with the number of objects. This effect depends on the specific order in which the partitions are explored, creating an erratic and undesirable performance. We propose a method to search the state space that fairly distributes these impoverishment effects between the objects by defining a set of mixture components and performing PS in each of these components using one of a small set of representative object orderings. Using synthetic and real data, we show that our method retains the overall performance and reduced computational cost of PS, while improving performance in scenes where the impoverishment effect is significant.

Tracking a significant number of objects is a difficult task because as objects are added the search becomes

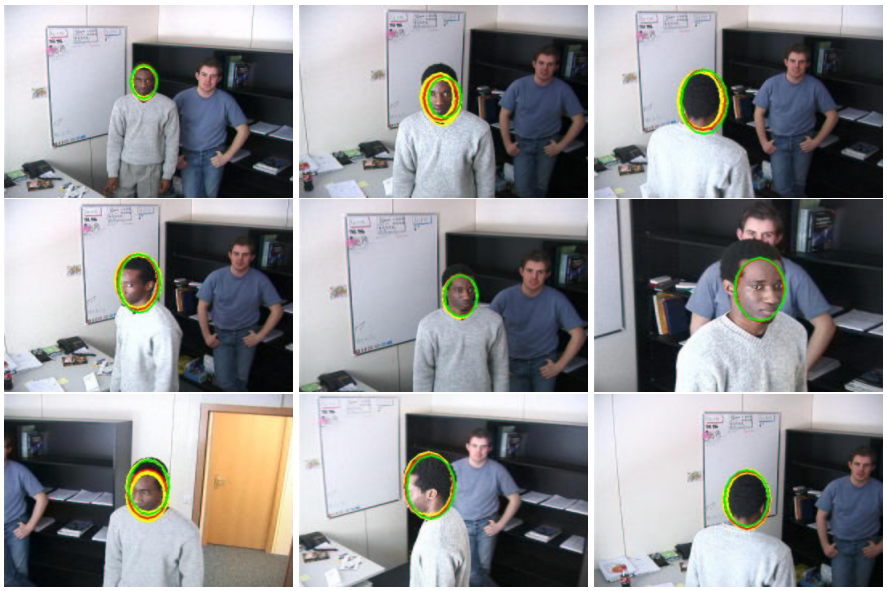

Figure 2: Tracker with motion proposal (N_s=1000) at time t_2, t_{40}, t_{85}, t_{100}, t_{130}, t_{145}, t_{170}, t_{195}, and t_{210} . In red, mean shape; in green, mode shape; in yellow, likely particles.

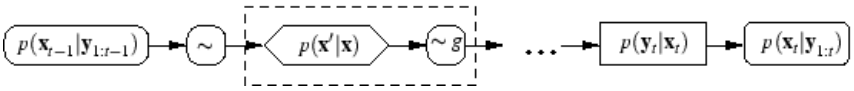

Figure 3: Block Diagram for PS [13]. The block in dashed box is repeated for each object.

exponentially more complex. MacCormick and Blake offered relief from this problem in the form of PS [12]. PS is a sampling strategy that reduces the dimensionality problem by handling one object at a time. Applied to multi-object tracking, PS divides the state space into M sub-space "partitions" and sequentially applies dynamics and performs weighted resampling in each of them. Weighted resampling, denoted $\sim$g in Figure 3, effectively reduces the computational cost by by exploring the state space individually, not jointly, and passing information from one object to the next. In Figure 3, the single-object dynamic process $p(\mathbf{x}'|\mathbf{x})$ and the weighted resampling step $\sim$g are repeated for each object successively.

PS does not treat objects equally because of its ordered nature. Weighted resampling in successive stages adversely affects the representation by impoverishing objects placed at early stages, very much in the same way a PF without resampling impoverishes a particle representation over time [10]. This impoverishment is caused by importance sampling and weight re-assignment in the weighted resampling step. As PS proceeds through the stages, the number of unique candidates from past objects is reduced with each weighted resampling step. This process might not be so evident for few objects, and was not, in fact, discussed by [13].

Because of this impoverishment, the objects in the first stage tend to have only a few remaining "good" candidates while the objects in the last stage will tend to have more, but potentially biased, candidates. Indeed, the weight set passed to the final stage will be heavily biased toward objects from previous stages.

This distortion of the representation can have disastrous effects on tracking performance. It can kill the ability to (a) maintain multi-modality; (b) adjust to new good, yet distant observations; (c) react to sudden fast motion in the presence of visual clutter. While the distortion will be less noticeable for smooth tracking conditions, it undermines some of the principle advantages of PFs.

4.2 Our Approach

We propose a method (Distributed Partitioned Sampling, or DPS) to search the state space that fairly distributes the impoverishment effects between the objects by defining a set of mixture components and performing PS in each of these components using one of a small set of representative object orderings.

Object Model. A bounding box divided into parts was used as the object model, where each part appearance is modeled by a color histogram [5]. In our implementation, the state for each object is defined as a continuous vector $\mathbf{x}^j = (u^j, v^j, \alpha^j)$ where (u^j, v^j) are image coordinates, j is the object index, and α^j is a scale parameter. For our work, M remains fixed but the ideas presented in this paper can be extended to deal with a variable number of objects [11].

Dynamic Model. The multi-object dynamic process consists of M AR2 processes defined for each object as $\mathbf{x}_t^j = \mathbf{x}_{t-1}^j + 0.5(\mathbf{x}_{t-1}^j - \mathbf{x}_{t-2}^j) + \sigma\omega_t^j$ where σ is a diagonal matrix of diagonal $(\sigma_u\ \sigma_v\ \sigma_\alpha)$, and ω_t is a 3D noise process of zero mean and unit variance.

Sampling Strategy. We can expect some PS orderings will fail and some will succeed, but without an explicit interaction model, we have no guess as to which orderings will do well and thus should be preferred. Distributed Partitioned Sampling (DPS) handles this by redefining the distribution as a mixture, composed of subsets of particles, on each of which we can perform PS in a different ordering (as each subset defines its own posterior). A filtering distribution, approximated by a particle set

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}), \tag{9}$$

with $\sum_{(i)=1}^{N} w_t^{(i)} = 1$, can always be re-expressed as a mixture model,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{c=1}^{C} \pi_{c,t}\ p_c(\mathbf{x}_t|\mathbf{y}_{1:t}), \tag{10}$$

where the mixture prior sums up to one $\sum_{c=1}^{C} \pi_{c,t} = 1$, and each mixture component p_c denotes a proper particle distribution defined over a subset I_c of particles,

$$p_c(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i \in I_c} \tilde{w}_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}), \tag{11}$$

where the prior and new weights are given by

$$\pi_{c,t} = \sum_{i \in I_c} w_t^{(i)}; \quad \tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\pi_{c,t}}. \tag{12}$$

In the block diagram of the DPS method seen in Figure 4, the mixture component creation step is denoted by a hexagon with a large X. We can define many mechanisms to associate particles to a specific mixture component, "branched" partitioned sampling [13] is a particular example. In our work, we randomly divide the particles into $C = M$ sets of N/M particles by sampling without replacement. Note, however, that other assignment strategies could be used to attempt to maintain multi-modality [14].

For our experiments, we decided to set the PS orderings as a representative subset of possible orderings in order to balance the effect of PS impoverishment defined by a circular shift $\{1 \rightarrow ... \rightarrow N\}, ..., \{N \rightarrow 1 \rightarrow ... \rightarrow N-1\}$. Under this formulation, experiments will show that DPS performs at the same level as PS for simple tracking, while suffering less from the effects of impoverishment.

After PS is completed for each mixture component, the subsets must be reassembled by simply applying Eq. 10: the weights from each subset are normalized and multiplied by the prior factor $\pi_{C,t}$ to ensure a fair representation in the distribution. The reassembly step is denoted by a dashed hexagon in Figure 4.

Observation Model. HSV histograms with spatial components were used as the observation model [5]. The object likelihood is defined as $p(\mathbf{y}_t|\mathbf{x}_t^j) \propto e^{-\lambda d^2}$ where λ is a hyper-parameter and d is the Bhattacharya distance between the specific observation and the template histograms.

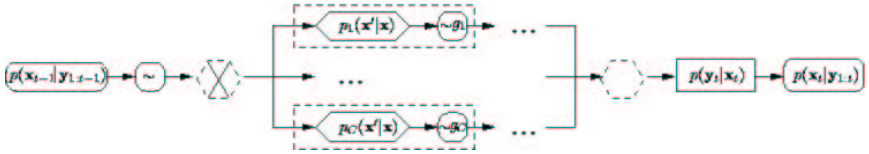

Figure 4: Block Diagram for DPS. The block in dashed box is repeated for each object.

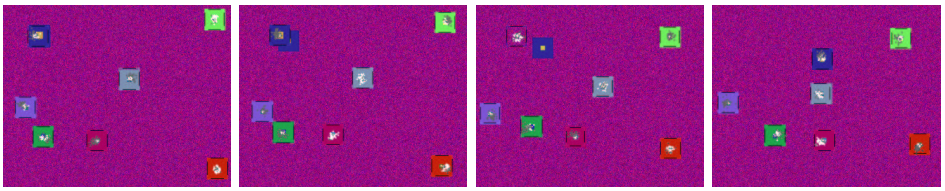

Figure 5: Frames 3, 5, 9, 14 from synthetic test sequence. Seven objects are tracked with a joint PF using PS. Distracter disappears in frame 9. (25 frames at 360x288).

4.3 Results

To see the benefits of partitioned sampling, a short synthetic test sequence was generated. This sequence consists of seven objects moving about a scene with a noise-filled background. Object dynamics were defined by an AR2 process with $\sigma_u = \sigma_v = 3$ and no size variation (σ_α).

Experiments were run with $N = 300$ particles over a range of likelihood hyper-parameter values (λ). An AR2 process with $\sigma_u = \sigma_v = 2$ and $\sigma_\alpha = 0.001$ was used for the dynamic process in tracking. Several performance measures are used to evaluate the performance over large numbers of experiments. These include: *Track State, Precision, Recall, Success Rate, Recovery Rate, Uniqueness,* and *Effective Dynamics.* Details can be found in the [2]. Results indicate that PS increases the tracking performance with respect to a simple PF.

While the Objects 2-7 are relatively simple to track, we induced tracking failure for Object 1 to test PS's ability to recover from tracking loss. The histogram model was learned from a "distracter" object which appears in frame 3 over the real object, leads the tracker astray, and disappears in frame 9, as seen in Figure 5. This synthetic scenario is meant to be analogous to occlusion with a similar object or the background followed by re-emergence.

Table 2 compares PS and Non-PS recovery rates for the distracted object (Object 1) on the synthetic test sequence. While the mean recovery rate for PS is greater than Non-PS, we can see that the worst-case-scenario is often barely so. The disparity between different orderings becomes even more apparent when comparing the best and worst case orderings (best: 38% recovery rate, worst: 8% recovery rate, for $\lambda = 30$).

The impoverishment effects become apparent when considering the uniqueness and the effective dynamics seen in Figure 6. Particle parameters from objects in early stages are successively resampled until only a few hypotheses remain. The PS impoverishment effects are so severe that even for the case of a not-so peaked

Hyper-parameter	Recovery Rate				Precision/Recall			
	Non-PS (%)	PS (mean %)	PS (worst %)	DPS (%)	Non-PS (%)	PS (mean %)	PS (worst %)	DPS (%)
$\lambda = 20$	2	19	10	42	27/28	27/26	17/16	41/36
$\lambda = 30$	2	18	8	34	21/21	24/24	16/16	36/36
$\lambda = 40$	0	21	16	38	23/22	30/29	18/17	21/22
$\lambda = 50$	2	21	12	36	24/23	28/28	25/24	23/23

Table 2: Recovery rate, precision, and recall for Object 1 for several λ values and various sampling methods. Each method was calculated over 50 runs ($50 * 7$ for P.S.) for $N = 300$ on the synthetic data sequence. Note: precision and recall are reported only for successful frames ($T = 1$).

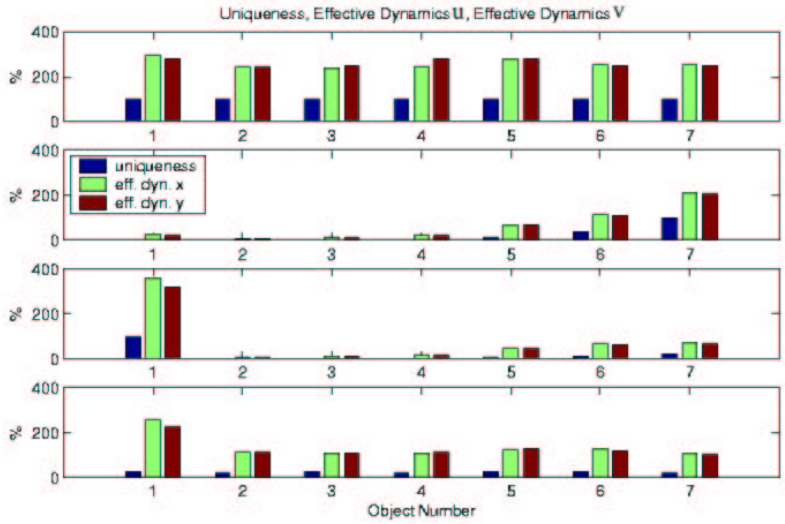

Figure 6: Uniqueness and effective dynamics in u and v for 7 objects in the synthetic sequence (50 runs, $\lambda = 20$, $N = 300$). Top: Non-PS. Second: PS $\{1 \rightarrow \ldots \rightarrow 7\}$. Third: PS $\{2 \rightarrow \ldots \rightarrow 7 \rightarrow 1\}$. Bottom: DPS. Notice the unequal effects of impoverishment on PS. Since these measures are normalized to N, σ_u, and σ_v resp., they are expressed here are percentages. Effective dynamics can exceed 100% because particles, in practice, try to follow the observations.

likelihood ($\lambda = 20$) the mean uniqueness (over all orderings) for the object in the first stage of PS is 1.99. Impoverishment will only become more pronounced as λ increases or more objects are added.

In addition to the synthetic sequences, real data sequences were considered. Tracking for the real sequences was done using color histograms as in the synthetic sequence. Templates for the trackers are shown in Figure 7. For each real sequence, 50 runs were performed for each of the following methods: Non-PS, PS with circular shifted ordering, and DPS. Unless otherwise specified, $N = 200$, $\lambda = 20$, $\sigma_u = \sigma_v = 2$, and $\sigma_\alpha = 0.001$. Histogram templates are initialized in the first frame.

In the first sequence, *RealSeq1*, a person is occluded twice as he walks behind two stationary people, seen in Figure 8. The occlusion process is relatively slow, and there are several frames of total occlusion. This sequence is another example of the adverse effects of impoverishment. In order for the occluded object (Object 1) to track properly, its dynamics must be sufficiently spread so that particles can cross the distracting face. Due to PS's reduced effective dynamics, these conditions do not exist.

Impoverishment severely affected PS in this sequence. It is shown in Table 3 that PS was significantly outperformed by Non-PS in two of the three cases. In one case PS only managed a success rate of 18%, while baseline Non-PS was 72%. DPS, on the other hand, successfully tracked the occluded object in all experiments. The successful PS ordering was $\{2 \rightarrow 3 \rightarrow 1\}$, as expected, as the occluded object was last in order and thus never underwent a weighted resampling step. Clearly, in PS the ordering of the objects matters. A video of the first five runs of Non-PS, the first five runs of PS $\{1 \rightarrow 2 \rightarrow 3\}$, and the first five runs of DPS (in that order, separated by a short yellow clip) is provided at www.idiap.ch/∼smith/.

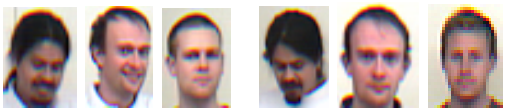

Figure 7: Templates used for the real data sequences. Left: the three objects from *RealSeq1*. Right: the three objects from *RealSeq2*. Template sizes vary from 42x32 to 33x32 pixels.

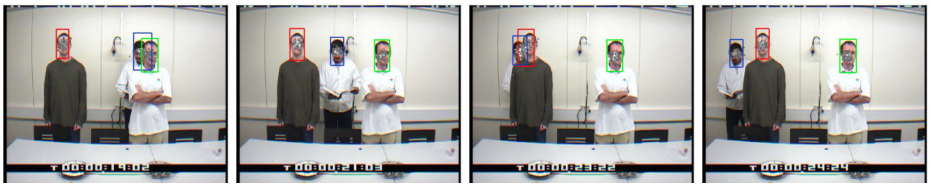

Figure 8: Tracking results for *RealSeq1*. DPS method recovers from occlusion (frames 40,91,160,197 shown). (191 frames at 360x288).

Method	*RealSeq1*		*RealSeq2*	
	Success Rate (%)	Recovery Rate (%)	Success Rate (%)	Recovery Rate (%)
Non-PS	72	74	22	32
PS ($1 \rightarrow 2 \rightarrow 3$)	18	18	8	8
PS ($2 \rightarrow 3 \rightarrow 1$)	96	96	28	28
PS ($3 \rightarrow 1 \rightarrow 2$)	52	52	8	8
Distributed	100	100	60	88

Table 3: Success and Recovery Rate results for occluded object (Object 1) over various sampling methods on *RealSeq1* and *RealSeq2* (50 runs each at $\lambda = 20$, $N = 200$ particles).

5 Mixed-State Models: Joint Head Tracking and Head Pose Estimation

5.1 Motivation

Head detection and tracking are essential components in video applications related to human behavior understanding. It is commonly used as a first step before applying algorithms for higher level task such as facial expression recognition, expression recognition or visual focus of attention estimation.

We have developed a generic method to perform joint tracking and activity recognition using mixed state model in a particle filter framework [3]. The approach combines in a joint state space model a continuous valued motion parameter (location, size, aspect ratio of people in the image) with a discrete labels (index of a set of head pose appearance).

In [3] the discrete latent variable describes an appearance exemplar of a given head pose based on textures features. Texture is very efficient to model head poses but it is distracted by cluttered background. In these cases, skin color might be very helpful. Textures are robust to changes in illumination conditions but are sensitive to background clutter. Color is robust to clutter but permit only gross head pose localization. These two visual cues are then complementary and using them together provide better head pose exemplars.

5.2 Our Approach

To jointly track the head and estimate pose, we use the mixed state particle filter framework. This framework allow to jointly represent in the state model $\mathbf{x} = (S, k)$ a continuous variable S representing the spatial configuration of the head (translation, rotation, scaling) and a discrete variable k specifying a head pose model.

Object Model. The head pose are defined by a pan angle ranging from -90 to 90 degrees. Allowed values are discretized with a 22.5 degrees step. Training data patches are extracted from head images by locating a tight bounding box around he head. These patch are resized to to the same 64×64 resolution. For each head pose a binary mask specifying the head location is extracted (Figure 9). These masks are used to focus only on the informative parts of a head image and to avoid building background dependent models.

• *Head Pose Texture Model:* To compute the texture exemplars of a given head pose, the training images of the head pose are converted to gray level image, and then preprocessed by histogram equalization. Four filters are then applied to the region specified by the head mask of the pose to give the texture training features. Then the texture features vectors are clustered using a K-means algorithm. The K centers of of clusters e_k^θ, $k =$

Figure 9: Example of head Pose discretization and pose dependent head masks

Figure 10: Texture features for a frontal head pose

$1, ..., K$ are taken to be the models of the head pose θ. The covariance matrix $\Sigma_k^\theta,\ k = 1, ..., K$ resulting from the clustering is kept for likelihood computation.

• *Head Pose Color Model:* Previous work about skin color modeling ([19]) has showed that human skin color are clustered in the color space and skin color differences among people can be reduced by intensity normalization. Also skin color distribution can be characterized by a single multivariate Gaussian distribution in the normalized color space .

Skin color is well defined in normalized rg color space. Using hand labeled data of face skin color pixels in training images we build a world skin color distribution represented by a Gaussian in the 2d color space with diagonal covariance matrix. The pose color model M_k^F is a binary mask of the face skin color region within the corresponding head mask. To build the color model corresponding to an exemplar e_k^θ, the training images belonging to the cluster of the head pose exemplar are converted into the normalized color space. The skin color distribution is used to classify each training image pixel into skin or non skin to give a binary mask for each training image. The exemplar color model is then the average of the masks of the training images of the pose.

Dynamic Model. We assume that the evolution of the state followed a second order process $\mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_{t-2})$. The skin color distribution parameters are estimated by MAP adaptation. This adaptation is given by :

$$m_t = (1 - \alpha)m_{t-1} + \alpha m(\hat{\mathbf{x}}_{t-1}) \tag{13}$$

where the adaptation parameter α is taken proportional to the color likelihood of the most likely state $\hat{\mathbf{x}}_{t-1}$, and $m(\hat{\mathbf{x}}_{t-1})$ is the skin color parameters estimated over the region of the most likely state.

Sampling Strategy. There are two hidden state in the model : the object state $\mathbf{x}$ and the skin color distribution model m. The object state is sampled according the sequential Monte Carlo principle using the dynamical model to take care of quick changes while the skin color distribution is adapted by MAP estimation because its

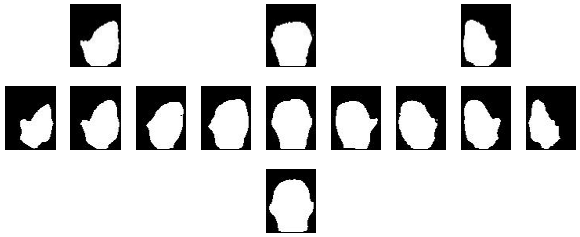

Figure 11: Exemplar dependent face masks

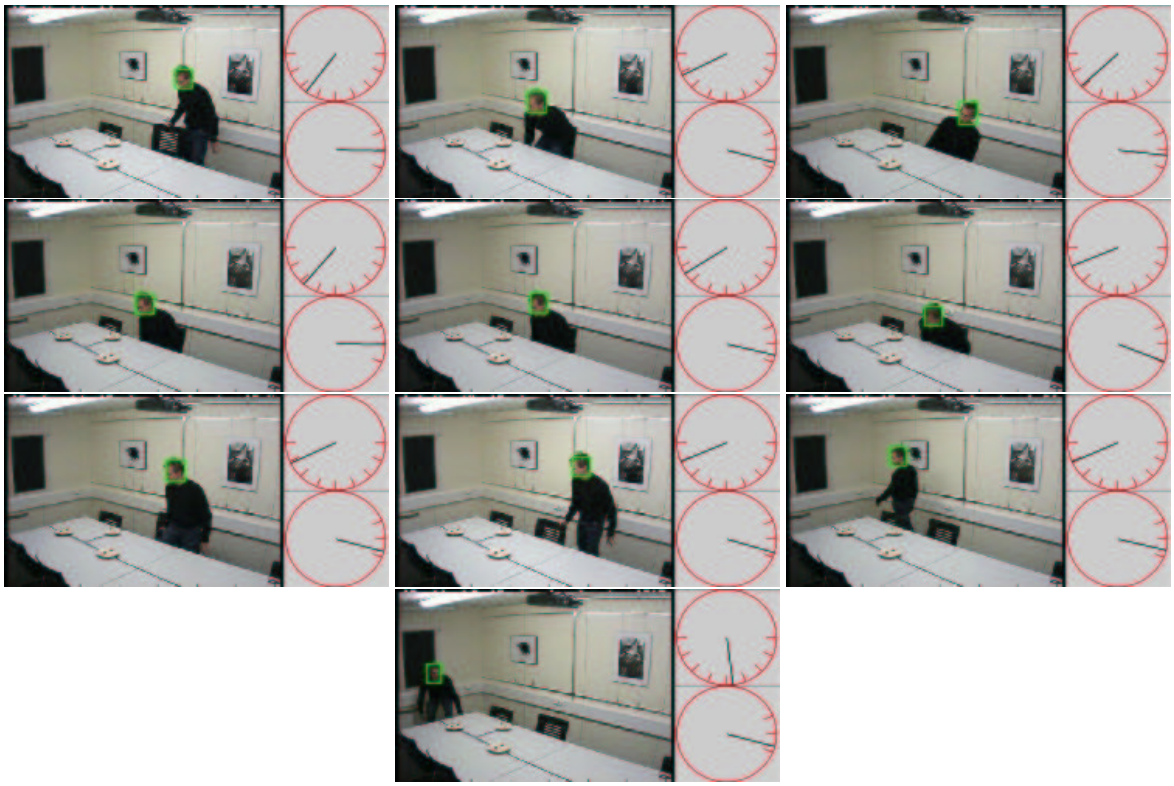

Figure 12: Meeting room tracking

variations are smoother.

Observation Model. To compute the likelihood of an input patch image located to the position S with respect to a head pose exemplar, first the image follow the same preprocessing and filtering steps to obtain the feature vector $\mathbf{y}^T(S)$. Then the feature vector is compared to the exemplar by:

$$p(\mathbf{y}^T|k) \propto \frac{1}{2\pi|\Sigma_k^\theta|} \exp -\frac{1}{2}(\mathbf{y}^T(S) - e_k^\theta)' \Sigma_k^{\theta -1} (\mathbf{y}^T(S) - e_k^\theta) \tag{14}$$

The likelihood of the image with respect to a pose color exemplar is computed by first resizing the image, and then computing the binary mask of skin color pixel $\mathbf{y}^C(S)$. The measured mask and the exemplar are compared by:

$$p(\mathbf{y}^C(S)|k) \propto \exp -\lambda||\mathbf{y}^C(S) - M_k^F||_1 \tag{15}$$

If we assume that texture and color measurement are independent, the joint likelihood is the product of the likelihood of each cue.

5.3 Results

We ran experiments to test the robustness of the method. Two video sequences were used, one is about a person in a meeting room and the other is about a person in an office. Tracking results are displayed in Figure 12 and 13. These tracking and head pose estimation examples are showing that the method is very robust even when heads are at very low resolution (Figure 12).

Figure 13: Office tracking

6 Mixed-State Models: Multi-Camera Multi-Modal Speaker Tracking

6.1 Motivation

Tracking speakers in multi-party conversations represents an important step towards automatic analysis of meetings. We have developed a probabilistic method for audio-visual (AV) speaker tracking in a multi-sensor meeting room. The algorithm fuses information coming from three uncalibrated cameras and a microphone array via a mixed-state importance particle filter, allowing for the integration of AV streams to exploit the complementary features of each modality. Our method relies on several principles. First, a mixed state space formulation is used to define a generative model for camera switching. Second, AV localization information is used to define an importance sampling function, which guides the search process of a particle filter towards regions of the configuration space likely to contain the true configuration (a speaker). Finally, the measurement process integrates shape, color, and audio observations. We show that the principled combination of imperfect modalities results in an algorithm that automatically initializes and tracks speakers engaged in real conversations, reliably switching across cameras and between participants.

6.2 Our Approach

A PF for AV speaker tracking involves the definition of the state-space, the speaker model, the dynamical process, the sampling strategy, and the observation models. We extend the previous use of I-PFs [15] to multi-modal fusion. Audio tends to be imprecise for localization, due to discontinuities during periods of non-speech, as well as effects of reverberation and other noise. Audio does have some important advantages however, such as the ability to provide instantaneous localization at reasonable computational expense. Additionally, even though audio can be inaccurate, it can still provide reasonable proposals that could be enriched by the use of extra visual localization information, and integrated in an additional importance sampling (IS) function. We propose an asymmetrical use of modalities, where audio and skin color are used for localization via sampling (as part of the IS function and the reinitialization prior), and shape, color and audio are further used as observations in the measurement process.

Object Model. We define a mixed-state model in which (i) human heads in the image plane are modeled as elements of a template-space, allowing for the description of a template and a set of valid transformations [8], and (ii) cameras depicting people are indexed by a discrete variable. Specifically, a state is defined by

$$X_t = (k_t, x_t), k \in \{0, .., N_K - 1\}, x_t \in \mathbb{R}^{N_x},$$

where k_t is a discrete N_K-valued camera index, and x_t is a continuous vector in the space of transformations $\mathbb{R}^{N_x}$. Speaker heads are represented by their silhouettes (contours) in the image plane [8]. In particular, we

used a parameterized vertical ellipse to represent the basic shape. We use three cameras ($N_K = 3$), and the space $\mathbb{R}^{N_x}$ is a subspace of the affine transformations comprising translation T^x, T^y and scaling θ, i.e., $x_t = (T_t^x, T_t^y, \theta_t)$.

Dynamic Model. The dynamical model can be factorized as follows,

$$p(X_t|X_{t-1}) = p(k_t|X_{t-1})p(x_t|k_t, X_{t-1}). \tag{16}$$

The first factor in the right side of Eq. 16 constitutes a generative model for switching cameras: for any given geometric transformation at the previous time, $p(k_t = n|k_{t-1} = m, x_{t-1}) = \mathbf{T}_{mn}(x_{t-1})$ represents a transition probability matrix (TPM) to switch between cameras. The second factor, $p(x_t|x_{t-1}, k_{t-1} = m, k_t = n) = p_{mn}(x_t|x_{t-1})$ denotes elements of a set of N_K^2 continuous dynamical models, one for each possible camera transition.

Sampling Strategy. The additional importance sampling function I_t in the I-PF formulation is defined by audio and color information. Our audio speaker localization approach consists of two steps: finding candidate source locations $\hat{Z}_t$, and classifying them as speech or non-speech. Details are presented in [4]. For each frame, each of the audio estimates in 3-D is mapped onto their corresponding image planes and 2-D locations. The mapping was learned from training data of people speaking and moving in typical regions in the meeting room. In the used camera setup, one camera has overlapping fields of view (FOV) with the other two, while these two cameras do not share FOVs with each other. We used a majority rule to keep all the proposals $C(Z_t) = (k_t, T_t^x, T^y)$ for the specific k_t^* that has the largest number of audio estimates N_t^a. The IS function is then defined as a Gaussian Mixture Model on the image plane for the k_t^* camera.

Observation Model. We propose to combine shape and localization (audio and color) information in the measurement process. The sole usage of shape is clearly limited to discriminate between two different human heads. In presence of multiple people or visual clutter, the shape likelihood is multi-modal, and particles with large weights would be generated for each person, and likely remain there even after a speaker turn. Fusing shape and localization information in the observation process would solve the above ambiguity, tracking speaker turns with lower latency, and locking only onto the current speaker. Modalities are fused by defining

$$p(\mathbf{y}_t|\mathbf{x}_t) = p(\mathbf{y}_t^{sh}|\mathbf{x}_t)p(\mathbf{y}_t^{loc}|\mathbf{x}_t), \tag{17}$$

where $p(\mathbf{y}_t^{sh}|\mathbf{x}_t)$ denotes a shape-based observation likelihood defined in Equation 4, and $p(\mathbf{y}_t^{loc}|\mathbf{x}_t)$ represents a localization likelihood, that uses audio and color information, and that in this work is defined using the IS function,

$$p(\mathbf{y}_t^{loc}|\mathbf{x}_t) \propto I_t(\mathbf{x}_t), \tag{18}$$

in case there is audio, and it is a fixed constant otherwise.

6.3 Results

This section presents an evaluation of both parts of the audio speaker localization system. Experiments were performed on a test case, an audio source localization system, a speech/non-speech classification system, and the global audio performance. Results for AV tracking are described here. For details on other experiments, see [4].

Parameters in the model (dynamics and observations) were hand-specified based on intuition, and kept fixed for all experiments. Parameter estimation is an issue that has to be addressed in future work.

Fig. 14(a-d) shows the results of tracking four speakers engaged in a two-minute conversation in the meeting room (3000 frames), using 500 particles (weighted mean of the posterior in red and standard deviation from the mean in yellow). Speakers talk at a natural pace, and one of the participants stands up and addresses the others from the projection screen and white board areas (see sequence `demo-test-seq1.avi`) found at www.idiap.ch/mucatar/. The tracker is automatically instantiated when a person starts talking, and remains for the most part in accurate track across participants for the rest of the sequence with small latency. In case of overlapping speech, the tracker locks onto only one speaker.

Table 4 presents an objective evaluation of the results, using a semi-automatically generated ground-truth (GT) of speaker segments, which consists of the camera index and the approximate speaker's head centroid in the corresponding image plane for each speaker segment.

We define two performance measures, and present results averaged over ten runs of the particle filter. The first measure is the error on the estimated camera indexes ϵ_k (with range [0,1]). Globally, the camera indexes were correctly estimated in 88.73% of the frames labeled in the GT. The second performance measure is the median over time of the error in the image plane $\epsilon_{(T^x,T^y)}$, between the GT and estimated mean 2-D positions, computed over all those frames for which the estimated camera index was correct.

error type	modality	cam_1	cam_2	cam_3	global
$\epsilon_k\,(\times 10^{-2})$	AV	1.91	0.31	25.00	11.27
		(0.09)	(0.09)	(0.39)	(0.18)
$\epsilon_{(T^x,T^y)}$	AV	1.88	1.69	0.40	1.00
		(0.08)	(0.18)	(0.01)	(0.03)
	A	11.39	11.86	10.60	11.20
	V	4.57	4.88	2.19	3.52

Table 4: AV tracking results. The std of each measure is shown in parenthesis; the units of $\epsilon_{(T^x,T^y)}$ are pixels.

For comparison, the results of audio-only localization are also shown in Table 4. Figures have been computed only taking into account frames with detected speech. The errors reported correspond to the median over time of the minimum error between the GT and all the audio estimates available at each frame. Such errors are the combined effect of 3-D localization and AV calibration. We have also included the results obtained by a visual-only, histogram-based tracker [5] initialized by hand at each speaker turn. Errors are slightly higher, although visually the performance is similar. For this sequence, AV fusion has shown better performance than each independent modality.

The benefit of using color and audio information compared to audio-only in the IS function and in the measurement process (Eq. 17) can be appreciated in the sequence `video2.avi` (images not shown here). In this video, the GMM defining the IS function consists only of audio estimates. It can be observed that tracking is less accurate than the observed in the previous example, due to the inherent limitations of single microphone-array estimates and the errors introduced by 3-D-to-2-D mapping that are not being improved by the use of color, as in `demo-test-seq1.avi` found at www.idiap.ch/mucatar.

Performance of the method on a cluttered background is shown in Fig. 14(e-i), and in `demo-test-seq2.avi` (1200 frames), and `demo-test-seq3.avi` (800 frames) (both found at www.idiap.ch/mucatar). The sequences display a four-party conversation with a fifth person walking in the room, and creating visual distractions by approaching the speakers. The tracker can get momentarily distracted by the walking person, or by the background visual clutter, but recovers in all cases. Although not shown here, work for a single-camera version of the system has shown that our formulation can also handle reinitialization in cases of total AV occlusion.

7 Conclusion and Future Work

In the second year of the project, we investigated sampling strategies to improve single and multi-object tracking with particle filters, and mixed-state representations to perform tracking in switching models. More precisely, we addressed the following issues:

Single-object tracking. We developed a model that handles abrupt motion changes and filters out visual distractors when tracking objects with models based on shape representations.

Multi-object tracking. We developed an improved sampling method to improve the performance and reduce the computational complexity of multi-object trackers.

Joint tracking and pose estimation. We developed a framework to perform simultaneous tracking and head pose estimation. With this algorithm we can achieve a basic activity recognition by detecting head turning in meetings.

Figure 14: (a-d) Tracking speakers in the meeting room. Frames 100, 1100, 1900, and 2700. (e-i) Tracking with visual distractions. Frames 313, 564, 640, 645, and 731.

Multi-camera speaker tracking. We developed a framework to track speakers across multiple cameras by fusing visual and audio features.

For the period July-December 2004, we plan to extend the work on several areas:

Multi-object tracking. We expect to work on three aspects. First, sampling strategies based on Markov Chain Monte Carlo (MCMC) will be investigated and compared with our current work. Second, interaction models between multiple objects will also be studied. Third, we will extend the work on speaker tracking to a multi-person tracker setup.

Activity recognition. The work done in head pose recognition will be expanded to more complex activities (e.g. head gestures). The work will also include the collection of a data set of head gestures in the meeting room, and the development of automatic tracking initialization mechanisms from multiple visual cues.

IM2.SA initiative for evaluation of tracking algorithms. We will collaborate with IM2.SA initiative to define a data set and standard performance evaluation procedures for multi-object tracking.

References

[1] J.M. Odobez, D. Gatica-Perez, "Embedding Motion in Model-Based Stochastic Tracking", in *Proc. Int. Conference on Pattern Recognition (ICPR)*, Cambridge, Aug. 2004.

[2] K. Smith, D. Gatica-Perez, "Order Matters: A Distributed Sampling Method for Multi-Object Tracking", in *Proc. British Machine Vision Conference (BMVC)*, London, Sep. 2004.

[3] S. Ba, J.M. Odobez, "A Probabilistic Framework for Joint Head Tracking and Pose Estimation", in *Proc. Int. Conference on Pattern Recognition (ICPR)*, Cambridge, Aug. 2004.

[4] D. Gatica-Perez, G. Lathoud, I. McCowan, J.M. Odobez, and D. Moore, "A Mixed-State I-Particle Filter for Multi-Camera Speaker Tracking", in *Int. Workshop on Multimedia Technologies in E-Learning and Collaboration*, Nice, Oct. 2003.

[5] P. Perez, C. Hue, J. Vermaak, and M. Gagnet, "Color-Based Probabilistic Tracking",in *Proc. European Conference on Computer Vision (ECCV)*, May 2002.

[6] J. Vermaak, M. Gagnet, A. Blake, and P. Perez, "Sequential Monte Carlo Fusion of Sound and Vision for Speaker Tracking", in *IEEE Proc. Int. Conference on Computer Vision (ICCV)*, Vancouver, July 2001.

[7] M. Beal, H. Attias, and N. Jojic, "Audio-Video Sensor Fusion with Probabilistic Graphical Models", in *Proc. European Conference on Computer Vision (ECCV)*, May 2002.

[8] A. Blake and M. Isard, "Active Contours", Springer-Verlag, 1998.

[9] I. Isard and A. Blake, "A Mixed-State CONDENSATION Tracker with Automatic Model-Switching", in *IEEE Proc. Int. Conference on Computer Vision (ICCV)*, 1998.

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

[11] M. Isard and J. MacCormick, "BRAMBLE: A Bayesian Multi-Blob Tracker", in *IEEE Proc. Int. Conference on Computer Vision (ICCV)*, Vancouver, Jul. 2001.

[12] J. MacCormick and A. Blake, "A Probabilistic Exclusion Principle for Tracking Multiple Objects", in *IEEE Proc. Int. Conference on Computer Vision (ICCV)*, pages 572–578, 1999.

[13] J. MacCormick and M. Isard, "Partitioned Sampling, Articulated Objects, and Interface-Quality Hand Tracking", in *Proc. European Conference on Computer Vision (ECCV)*, 2000.

[14] J. Vermaak, A. Doucet, and P. Perez, “Maintaining Multi-Modality through Mixture Tracking”, in *IEEE Proc. Int. Conference on Computer Vision (ICCV)*, Nice, Oct. 2003.

[15] M. Isard and A. Blake, “ICONDENSATION: Unifying Low-level and High-level in a Stochastic Framework”, in *Proc. European Conference on Computer Vision (ECCV)*, 1998.

[16] J.M. Odobez, S.O. BA D. Gatica-Perez and K. Smith, “Mucatar Project Deliverable D3: Towards Joint Tracking And Activity Recognition”, in *Idiap Research Report*, Jan 2004.

[17] J.M. Odobez and P. Bouthemy, “Robust Multireolution Estimation of Parametric Motion Model”, in *Jl of Visual Com. and Image Representation*, 1995.

[18] J.M. Odobez and D. Gatica-Perez, “Embedding Motion in Model-Based Stochastic Tracking” in *Idiap Research Report*, 2003.

[19] J. Yang, W. Lu and A. Waibel, “Skin-Color Modeling and Adaptation”, in *Proc. of Asian Conference on Computer Vision (ACCV)*, Hong Kong, 1998.