Order Matters: A Distributed Sampling Method for Multi-Object Tracking

Kevin Smith and Daniel Gatica-Perez
IDIAP Research Institute, 1920 Martigny, Switzerland

Abstract

Multi-Object tracking (MOT) 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) solves this problem by partitioning the search space, 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.

1 Introduction

With the multitude of applications in surveillance, HCI, etc., MOT constitutes a active area in computer vision. Among the different approaches, Sequential Monte Carlo (SMC) [3] has enjoyed a great deal of success, as witnessed by the published literature [5, 6, 9, 4, 7, 10, 11]. The SMC approach, also known as particle filtering (PF), is a technique for implementing recursive Bayesian filtering by Monte Carlo simulations. Essentially, PFs perform a random search in a configuration space. For a state-space model, PFs recursively approximate the filtering distribution of states given observations using a set of weighted samples (particles), a dynamical model, and random sampling. This is done by predicting candidate configurations and then measuring their likelihood.

The problem of visually tracking a single object is well studied: good object models have been developed and show good performance [12, 7]. Many issues still exist in the case of MOT including multi-object representation [10, 11], handling variable numbers of objects [9, 4], interacting-parts object models [5, 6], and sampling strategies [5, 6].

For our work, an explicit representation of the joint multi-object distribution [5, 4] is used as it constitutes a sound, rigorous implementation of the MOT problem. Because the state space directly represents the joint object configurations, *interactions* can be formally modeled [5], and concepts like the mean configuration retain their physical meaning.

Although formal and conceptually simple, the explicit representation has a strong limitation: the cost to explore a multi-object state space with particles grows (roughly) exponentially with the number of objects [5, 6]. To overcome this problem, complex sampling strategies or strong priors are needed to place particles more effectively. Such sampling strategies attempt to explore the state space in an efficient manner to improve prediction using a suite of statistical techniques. Examples include layered sampling [8], annealed importance sampling [2], hybrid Monte-Carlo [1], and PS [5].

PS tackles the dimensionality problem by separating the parameters into "partitions" by object, and then searching within each objects parameter space by applying dynamic processes and performing weighted resampling. It is an attractive method because it is mathematically rigorous, significantly reduces computation, and tracks robustly. We will show that for the same computational cost, a MOT PF with PS significantly outperforms a traditional MOT PF.

In this paper, we show that, in practice, the sequential weighted resampling steps cause a kind of impoverishment in the particle set which adversely affects the representation and compromises tracking performance. The severity of the impoverishment effect depends *on the order in which the partitions are explored* and increases as more objects are added. A key advantage of PF tracking is that PFs deal well with one of the most difficult problems in tracking, recovering from occlusion. However, we will show that the impoverishment effect in PS severely impedes its ability to recover from occlusion.

Thus, we formulate a solution to this problem in the form of *distributed partitioned sampling* (DPS). The concept behind DPS is to share the burden of impoverishment from PS between the objects by splitting the distribution into a set of mixture components; and choosing a number of different orderings to treat the partitions. We show through experimentation that DPS retains the computational benefit of PS, while exhibiting similar or better tracking performance and reducing the impoverishment effect.

This paper is organized as follows. Section 2 describes particle filters for MOT. Section 3 details PS, outlines a set of tracking performance measures, and uses these measures to explore the limitations of PS on synthetic data. In Section 4, DPS is formulated. Section 5 compares DPS, PS, and Non-PS on several real and synthetic data sequences. In Section 6 conclusions and future work are presented.

2 Particle Filters for Multi-Object Tracking

Given an object representation and a Markov state-space model, the problem of tracking multiple objects can be approached as the recursive estimation of the hidden states $\{\mathbf{x}_t\}$ belonging to a multi-object configuration $\mathbf{x}_t = (M, \mathbf{x}_t^{1:M}) = (M, \mathbf{x}_t^1, ..., \mathbf{x}_t^M)$ from measurements $\{\mathbf{y}_t\}$ extracted from a video sequence, where M is the number of objects. Bayesian filtering computes the distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ recursively 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 $\mathbf{y}_{1:t} = (\mathbf{y}_1, ..., \mathbf{y}_t)$. Assuming the distribution at time $t-1$ is available, the multi-object 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 $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ (the integral from Equation 1) in the prediction step. The prediction distribution becomes the prior for the update step using Bayes' rule to generate the current filtering distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ where it is multiplied by the multi-object likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$. Usually, exact inference of this model is intractable. SMC methods are employed to approximate Eq. 1 for non-linear, non-Gaussian problems using a set of weighted samples or particles $\{(\mathbf{x}_t^{(i)}, w_t^{(i)}), i = 1, ..., N\}$ by $\hat{p}_N(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i=1}^N w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)})$, where N is the number of particles. The prediction step propagates each particle according to the multi-object dynamics, and the updating step reweights them using their likelihood, $w_t^{(i)} \propto w_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$. Resampling is necessary to avoid degradation of the particle set [3].

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 continuous image coordinates, and α^j is a continuous 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 [4]. 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. Regarding the observation likelihood, we define $p(\mathbf{y}_t|\mathbf{x}_t) = \prod_{j=1}^{M} p(\mathbf{y}_t|\mathbf{x}_t^j)$. The tracker output is given by the mean state of the parameters of the particle set.

3 Improved Search: Partitioned Sampling

Tracking a significant number of objects is a difficult task because as objects are added the search becomes exponentially more complex. MacCormick and Blake offered relief from this problem in the form of PS [5]. 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.

In weighted resampling, importance sampling is performed using importance weights $\rho^{(i),j} = g(x^{(i),j}) / \sum_{k=1}^{N} g(x^{(k),j})$ for the current object, where g is a weighting function, and j is the object index. The weights are then adjusted to preserve the underlying distribution, $w^{(i'),j} = w^{(i),j}/\rho^{(i),j}$. This process is denoted ~g in Figure 1. Weighted resampling 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 1, the single-object dynamic process $p(\mathbf{x}'|\mathbf{x})$ and the weighted resampling step ~g are repeated for each object successively. The final object does not undergo a weighted resampling step, but simply passes the particle set to the measurement likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ before forming the posterior estimate.

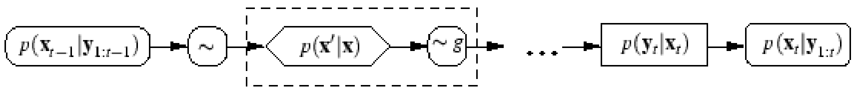

Figure 1: Block Diagram for PS [6]. The block in dashed box is repeated for each object.

3.1 Performance Measures. In this paper, we introduce a number of performance measures which require a labeled ground truth for the dataset, and assume that successful tracking occurs when an estimated object area E has a non-null intersection with a ground truth image area G. Measures are computed over a large number of runs and defined as:

Track State: a binary variable T that indicates the tracking state of an object at each frame, defined as one if $E \cap G \neq \emptyset$, and zero otherwise.

Precision: a percentage defined at each frame as $P = E \cap G/E$.

Recall: a percentage defined at each frame as $R = E \cap G/G$.

Success Rate: the percentage of sequences an object was successfully tracked throughout. Defined as the sum of runs in which $T = 1$ for all t over the number of runs.

Recovery Rate: a percentage which indicates what fraction of sequences an object either successfully tracked throughout or recovered from lost tracking.

In addition, two measures were used to assess specific aspects of the performance in particle filters:

Uniqueness: defined as U/N. The number of distinct configuration hypotheses U normalized by the total number of particles N.

Effective dynamics: defined as the variance in the configuration parameters, estimated at the end of each time step in the particle filter.

3.2 Testing Partitioned Sampling. 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). The objects can be seen in Figure 2.

HSV histograms with spatial components were used as the observation model [7]. 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.

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. It is clear from Table 1 that PS increases the tracking performance with respect to a simple PF. Success rate is increased to 100%, and precision and recall see an absolute increase of 22% and 19.75% over the range $\lambda = 20$ to $\lambda = 40$, respectively.

Figure 2: Left: object templates for synthetic sequence. Right: distracter is similar to Object 1.

	Success Rate			Precision/Recall		
	Non-PS (%)	PS (%)	DPS (%)	Non-PS (%)	PS (%)	DPS (%)
$\lambda = 20$	97	100	100	69/69	90/87	91/86
$\lambda = 30$	97	100	100	69/68	91/87	90/86
$\lambda = 40$	93	100	100	68/68	91/88	90/87
$\lambda = 50$	96	100	100	67/66	91/88	90/87

Table 1: Success rate, precision, and recall results for Objects 2-7 for several λ values and various sampling methods. Each method was calculated over 50 runs ($50 * 7$ for PS), with $N = 300$ particles on the synthetic data sequence. The performance for Object 1 is treated separately.

3.3 All Objects are Not Created Equal. Because of its ordered nature, PS does not treat objects equally. 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 [3]. 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 [6].

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.

As the number of possible orderings PS can be applied in is factorial without a "natural" ordering (imposed e.g. by an occlusion model), in practice, it becomes a matter of

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 ($T = 1$).

luck if a difficult-to-track object falls in a favorable ordering. This kind of erratic behavior is undesirable and becomes evident over multiple runs. Since it was not feasible to test 7! different orderings we selected a small but representative subset defined by a circular shift. This choice was made so that each object occupies each position in the ordering list only once. Thus, the PS ordering set was defined as $\{1 \to \ldots \to 7\}$, $\{2 \to 3 \to \ldots \to 7 \to 1\}$, ... , $\{7 \to 1 \to \ldots \to 6\}$. 50 runs were performed for Non-PS and each PS ordering.

The synthetic sequence in the previous section was designed to expose the weakness caused by impoverishment. While the other six objects 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 (see Figure 2) which appears in frame 3 over the real object, leads the tracker astray, and disappears in frame 9, as seen in Figure 3. This synthetic scenario is meant to be analogous to occlusion with a similar object or the background followed by re-emergence.

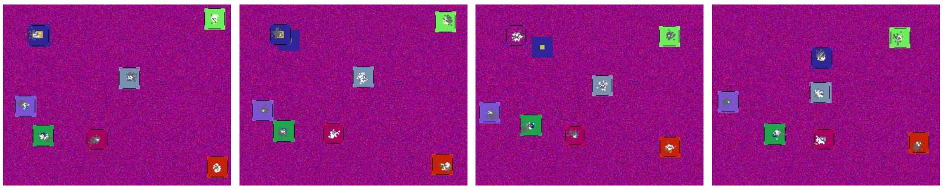

Figure 3: Frames 3, 5, 9, 14 from synthetic test sequence. Seven objects are tracked with a joint PF using PS. Distracter disappears in frame 9. (25 frames at 360x288).

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 4. 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 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.

4 Our Solution: Distributed Partitioned Sampling (DPS)

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. 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

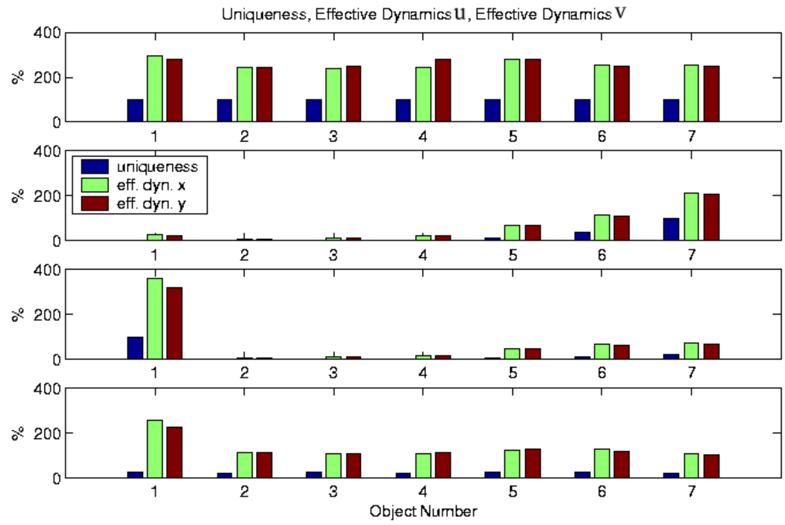

Figure 4: Uniqueness and effective dynamics in u and v for 7 objects in the synthetic sequence (50 runs, $\lambda = 20$, $N = 300$). Top: Non-PS. Second: PS $\{1 \rightarrow \ldots \rightarrow 7\}$. Third: PS $\{2 \rightarrow \ldots \rightarrow 7 \rightarrow 1\}$. Bottom: DPS. Notice the unequal effects 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.

subset defines its own posterior). A filtering distribution, approximated by a particle set

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}), \tag{2}$$

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

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{c=1}^{C} \pi_{c,t}\ p_c(\mathbf{x}_t|\mathbf{y}_{1:t}), \tag{3}$$

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,

$$p_c(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i \in I_c} \tilde{w}_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}), \tag{4}$$

where the prior and new weights are given by

$$\pi_{c,t} = \sum_{i \in I_c} w_t^{(i)}; \quad \tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\pi_{c,t}}. \tag{5}$$

In the block diagram of the DPS method seen in Figure 5, 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 [6] 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 [11].

For our experiments, we decided to set the PS orderings as a representative subset of possible orderings defined by a circular shift $\{1 \rightarrow \ldots \rightarrow N\}$, ... , $\{N \rightarrow 1 \rightarrow \ldots \rightarrow N-1\}$ in order to balance the effect of PS impoverishment. 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. 3: the weights from each subset are normalized and multiplied

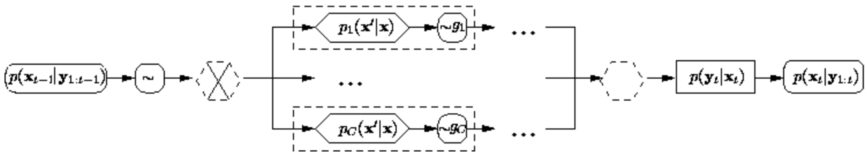

Figure 5: Block Diagram for DPS. The block in dashed box is repeated for each object.

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 5.

5 Results

5.1 Synthetic Data Revisited. Re-examining the synthetic test sequence, the benefits of DPS can be seen. It is evident from Table 1 that for the 6 easy-to-track objects, performance is on par with PS. In Table 2, we can see that for the distracted object, DPS consistently recovers from failure better than PS.

When looking at the quality of tracking (precision/recall) for the distracted object, we find that the worst-case ordering for PS is often worse than for Non-PS. It is clear from Table 2 that DPS offers a higher, more reliable quality of tracking at better success and recovery rates (an absolute increase of 3% and 2.5% for precision and recall, respectively, averaged over all λ values).

Looking again at Figure 4, we can see that DPS indeed more fairly distributes the burden of impoverishment between the different objects. Note that the effective dynamic values for Object 1 in each method appear boosted because of the difficulty in tracking that object (the particles tend to spread out due to ambiguity caused by the distracting object and quicker motion).

A video consisting of the first five runs of Non-PS, the first five runs of PS $\{7 \rightarrow 1 \rightarrow \ldots \rightarrow 6\}$, and the first five runs of DPS on the synthetic data (in that order, separated by a short yellow clip) is provided at www.idiap.ch/∼smith/.

5.2 Real Data. In addition to the synthetic sequences, three 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 6. 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.

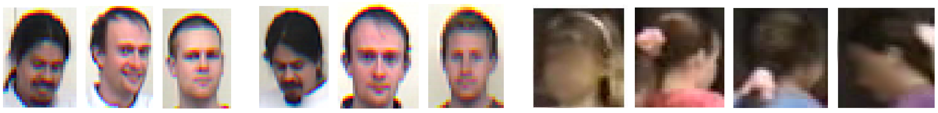

Figure 6: Templates used for the real data sequences. Left: the three objects from *RealSeq1*. Center: the three objects from *RealSeq2*. Right: the four objects from *RealSeq3*. Template sizes vary from 42x32 (*RealSeq1*) to 33x32 (*RealSeq3*) pixels.

In the first sequence, *RealSeq1*, a person is occluded twice as he walks behind two stationary people, seen in Figure 7. 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.

As shown in Figure 8, even when tracking few objects, the uniqueness and effective dynamics are substantially decreased for two of the three stages. The last stage of

Figure 7: Tracking results for *RealSeq1*. Top: PS method fails to recover from occlusion (frames 10,32,41,70 shown). Bottom: DPS method recovers from occlusion (frames 40,91,160,197 shown). (191 frames at 360x288).

Method	*RealSeq1* Suc. (%)	Rcv. (%)	Pre. (%)	Rcl. (%)	*RealSeq2* Suc. (%)	Rcv. (%)	Pre. (%)	Rcl. (%)
Non-PS	32	78	59	98	10	32	72	86
PS ($1 \to 2 \to 3$)	18	18	60	98	0	8	55	91
PS ($2 \to 3 \to 1$)	40	96	58	98	12	40	72	86
PS ($3 \to 1 \to 2$)	34	52	62	99	0	8	59	91
DPS	100	100	62	100	78	88	58	98

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

PS always has Non-PS-like values (Non-PS values not shown) because there is no final weighted resampling step. Again, DPS shares impoverishment evenly between the three objects.

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 32%. DPS, on the other hand, successfully tracked the occluded object in all experiments. The successful PS ordering was $\{2\to3\to1\}$, as expected, as the occluded object was last in order and thus never underwent a weighted resampling step. In Figure 8 it is shown that for this PS ordering alone, were the effective dynamics and uniqueness on par with that of DPS. 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 $\{1\to2\to3\}$, and the first five runs of DPS (in that order, separated by a short yellow clip) is provided at www.idiap.ch/∼smith/.

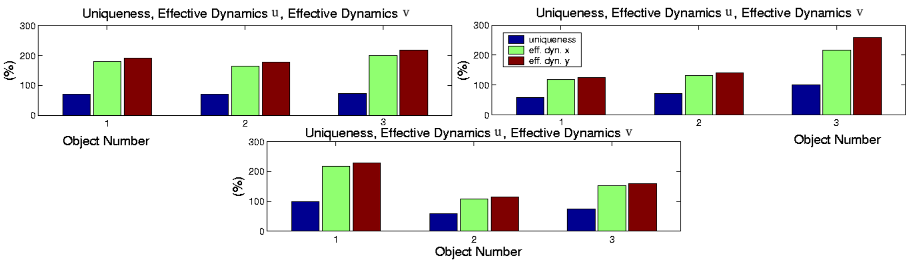

Figure 8: Uniqueness, effective dynamics in u, and effective dynamics in v for 3 objects in *RealSeq1* over 50 runs ($\lambda = 20$, $N = 200$). Top-left: DPS. Top-right: PS order ($1\to2\to3$). Bottom: PS order ($2\to3\to1$). PS impoverishment causes reduced effective dynamics and uniqueness, resulting in diminished ability to recover from occlusion.

The second real data sequence (*RealSeq2* seen in Figure 9) produced similar results. In this sequence, the target object (Object 1) is occluded as he first walks behind another person, then passes in front of a second. This scenario again tests the ability of the dynamic process to recover from occlusion, but more so as the occlusion lasts longer (6 frames of significant occlusion in the 1^{st} case, 10 frames in 2^{nd}).

Figure 9: Tracking results for *RealSeq2* . PS ($1 \rightarrow 2 \rightarrow 3$) fails to recover from occlusion (frames 2,20,40,120 shown) success and recovery rate: 8%. Particle (u, v) positions are superimposed on the image. Brighter pixels indicate higher likelihoods (139 frames at 360x288). On the contrary, DPS has higher success and recovery rates: 60% and 88% respectively.

The results for *RealSeq2*, presented in Table 3, confirm that the success and recovery rates for DPS are consistently better than PS and Non-PS. In this case, DPS outperforms all three orderings of PS. It is important to point out that *precision* and *recall* are calculated over successfully tracked frames only, so a successful tracker may have lower *precision/recall* scores for a difficult scenario.

A limitation to DPS is exhibited in the third real data sequence, *RealSeq3*. This sequence captures four dancing girls, as seen in Figure 10. As in previous sequences, each object tracked a girl and one (Object 4) was tested for its ability to recover from occlusion.

In each previous example, DPS has performed better than or equal to the mean of PS orderings (it outperformed all orderings, in fact). Thus, for a randomly selected PS ordering, DPS would out-perform PS. For this sequence, however, this is not the case. Figure 4 shows DPS success recovery rates (36%/38%) lower than the mean PS rates (65.5%/68.5%). DPS still performs better than the PS worst-case-scenario and Non-PS. This reduced performance of DPS potentially points out a limitation of our method.

Since DPS splits the distribution into mixture components, each component consists of a subset of the original particle set. Cases can exist (such as *RealSeq3*) when the number of particles in the mixture component is not sufficient to perform proper tracking from one frame to the next. This can happen due to poorly tuned models, tough dynamics, etc., and is made worse by the impoverishment effect. In *RealSeq3*, four objects were tracked using a total of 200 particles so each mixture component was forced to track with just 50 particles. In the synthetic test case this was enough, but for this real sequence it was not. Further research is needed in this area.

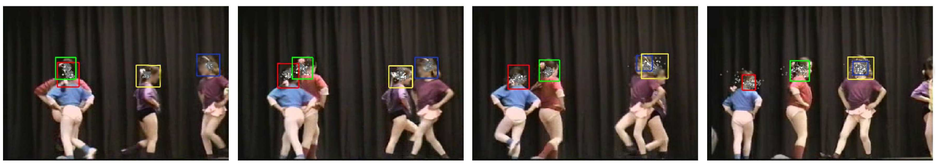

Figure 10: Tracking results for *RealSeq3*. DPS fails to recover from occlusion for Object 4 - the girl with the black pants (frames 1,7,14,32 shown). (32 frames at 352x240).

Method	Success Rate (%)	Recovery Rate (%)	Precision (%)	Recall (%)
Non-P.S.	4	8	41	70
P.S. ($1 \to 2 \to 3 \to 4$)	26	28	46	77
P.S. ($2 \to 3 \to 4 \to 1$)	82	86	52	87
P.S. ($3 \to 4 \to 1 \to 2$)	84	88	51	87
P.S. ($4 \to 1 \to 2 \to 3$)	70	72	51	86
Distributed	36	38	51	79

Table 4: Precision, Recall, Success and Recovery Rate for occluded object (Object 4) over various sampling methods on *RealSeq3* (50 runs each at $\lambda = 20$, $N = 200$ particles).

6 Conclusion and Future Work

In this paper we have shown the benefits of using PS over the traditional PF through systematic experimentation over a synthetic data set. We defined a set of performance measures for evaluation of tracking results. With these performance measures, we have shown that choosing a good ordering is very important when using PS. Because it is usually impossible to know which orderings will perform best, we used a mixture representation to formulate a sampling method that retains the formal rigor and the computational advantages of PS while reducing the effects of impoverishment. Finally, we verified our hypotheses on several sequences of real and synthetic data over many runs.

The implementation of DPS presented in this paper only assigns mixture components in the most simple possible manner. Future work will explore other methods for creating mixture components to help deal with cases of low particle numbers, as in *RealSeq3*.

Acknowledgments. We would like to thank the Eastman Kodak Company for providing *RealSeq2* and Jean-Marc Odobez for his helpful discussions. This work has been supported by the IM2.MUCATAR and IST-M4 projects.

References

[1] K. Choo and D. Fleet. People tracking with Hybrid Monte Carlo. In *Proc. IEEE ICCV*, 2001.

[2] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. IEEE CVPR*, 2000.

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

[4] M. Isard and J. MacCormick. BRAMBLE: A Bayesian multi-blob tracker. In *Proc. IEEE ICCV*, Vancouver, Jul. 2001.

[5] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. IEEE ICCV*, pages 572–578, 1999.

[6] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. ECCV*, pages 2:3–19, 2000.

[7] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based Probabilistic Tracking. In *Proc. ECCV*, Copenhagen, May 2002.

[8] J. Sullivan, A. Blake, M. Isard, and J. MacCormick. Object localization by Bayesian correlation. In *Proc. IEEE ICCV*, pages 1068–1075, 1999.

[9] H. Tao, H. Sawhney, and R. Kumar. A Sampling Algorithm for Detecting and Tracking Multiple Objects. In *Proc. IEEE ICCV Workshop on Vision Algorithms*, 1999.

[10] D. Tweed and A. Calway. Tracking Many Objects using Subordinate Condensation. In *Proc. BMVC*, Oct.. 2002.

[11] J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. In *Proc. IEEE ICCV*, Nice, Oct. 2003.

[12] Y. Wu and T. Huang. A co-inference approach for robust visual tracking. In *Proc. IEEE ICCV*, Vancouver, July 2001.