Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Factored Particle Filtering with Dependent and Constrained Partition Dynamics for Tracking Deformable Objects

M. Taner Eskil

Received: / Accepted:

Abstract In particle filtering, dimensionality of the state space can be reduced by tracking control (or feature) points as independent objects, which are traditionally named as *partitions*. Two critical decisions have to be made in implementation of reduced state-space dimensionality. First is how to construct a dynamic (transition) model for partitions that are inherently dependent. Second critical decision is how to filter partition states such that a viable and likely object state is achieved. In this study, we present a correlation based transition model and a proposal function that incorporate partition dependency in particle filtering in a computationally tractable manner. We test our algorithm on challenging examples of occlusion, clutter and drastic changes in relative speeds of partitions. Our successful results with as low as 10 particles per partition indicate that the proposed algorithm is both robust and efficient.

Keywords Particle filtering · condensation · factorized likelihoods · deterministic drift · proposal function

1 Introduction

Stochastic approach to tracking is driven by a state–space model of the tracking object. The state of the object is estimated with a probability density function, which was for convenience assumed to be Gaussian in earlier works. Kalman filter [1], the most popular tracking algorithm with Gaussian probability density assumption, is a provably optimal estimator for linear systems. However, Kalman filter fails in cluttered scenes, occlusions, and sudden changes of state, for which the probability density of state is multimodal.

Particle filtering offers the possibility of propagating arbitrary probability densities through sequential importance sampling (SIS) technique. Introduced as the *condensation* algorithm by Isard and Blake [2], this approach is shown to be capable of representing arbitrary functions by propagating multiple hypotheses. Condensation algorithm models state with a vector of control points which define the contour of the tracking object through B-spline curves. Pitt and Shephard [3] provide an excellent survey of early works in particle filtering, including the condensation algorithm. One disadvantage of consolidating the state of an object (e.g. control points) in vector representation is increased dimensionality of the state space [4], which in turn necessitates exponential increase in the number of hypotheses.

The problem of high dimensionality can be observed both in tracking of multiple similar objects and a single object with complex variations in shape. MacCormick [5] proposed *weighted resampling* and *partitioned sampling* algorithms for tracking multiple similar objects. Weighted resampling algorithm populates more likely regions depending on an importance factor for generating more accurate posterior probability distributions. Partitioned sampling reduces the search space by decomposing the joint dynamics of multiple objects. This study demonstrated the explosion in the number of particles required as the dimensionality of the search space increases.

Tracking of multiple similar objects requires assignment of observations to correct target models. The number of possible matches is combinatorial, which leads to an NP–hard problem [6]. MacCormick and Blake [7] developed a probabilistic exclusion principle in tracking of

M. Taner Eskil
AMF-319, IŞIK University, Şile, Ístanbul, Turkey
Tel.: +90-216-5287155
E-mail: eskil@isikun.edu.tr

two objects. Hue et al. [6] proposed a solution to the association problem of posterior densities through Gibbs sampler.

The principle of dimensionality reduction is applicable to tracking of a single, deformable object. In this approach the object is segmented into non-overlapping *partitions*, which are tracked independently. Its inherent problems are similar to tracking of multiple similar objects, such as the proposal of more likely regions for each partition and selection of a set of hypotheses from partitions to obtain a viable state of the tracking object.

Partitioned sampling was proposed by MacCormick and Isard [8] for tracking of an articulated object. It is an effective method for reducing the computational cost that is incurred by the dimensionality of the state–space. They demonstrated the operation of partitioned sampling via *survival rate*, which is a way to evaluate the reliability of tracking. In this work association of partitions was not addressed to, but mentioned in future works with a proposed tree structure.

Deutscher and Ian [9] proposed a simulated annealing based optimization of a multi-model objective function through *annealed particle filtering.* The approach was demonstrated on an articulated human body tracking problem. They implemented a soft partitioning strategy which allows hierarchical search on partition states with respect to tracking confidence. Wu and Nevatia [10] implemented a boosting based method to learn the shape features of articulated body parts, which is termed by them as *edgelet features.* Responses of part detectors are combined in a joint likelihood function through a greedy algorithm.

Patras and Pantic [11] attempted to reduce the dimensionality of the state space through *particle filtering with factorized likelihoods* (PFFL) algorithm. Feature points on an object are represented with partitions, each of which is independently progressed during the course of tracking. They reduced the number of wasted particles by sampling particles based on their likelihoods after a simulated transition. In the literature, RANSAC based [12] particle sampling procedures were also proposed.

PFFL algorithm treats each partition as an independent object with a state and a reference model. The dependency between the partitions, hence the overall object state is modeled with a *factorized proposal function.* One particle is sampled for each partition, such that the proposal function is maximized. PFFL was shown to be successful in tracking of facial features [11, 13] and gaze tracking [14].

Modular algorithms that work independently on partition level reduce the dimensionality of the problem, offering simplicity and higher speed in tracking. These algorithms also enable tracking of deformable objects through suitable constraints for relative motion within [15] or between [16] objects. Multi-scale systems that propagate information from partition to object and vice versa have also been proposed [17]. Multi-cue systems utilize a voting based scheme to integrate the tracking estimates of object partitions [18–20].

There are two challenges in implementation of particle filtering with reduced state-space dimensionality through partitions. First is to model the transition dynamics of partitions which are inherently dependent. When partitions are allowed to translate independent of each other, their posterior distributions eventually drift away to states that are incompatible with a viable state of the object. On the other hand, partitions chosen on an object indeed move in a pattern, and in varying degrees of dependency with other partitions.In the literature partition dynamics has been modeled as Gaussian noise [11,21] and independent speeds [14]. Graph based models were proposed to incorporate temporal and spatial relations between feature points [22].

We encounter the second difficulty in the final stage of tracking; i.e. sampling a set of particles according to a proposal function. This task is simpler to deal with in the beginning stages of tracking where all partitions exhibit a unimodal distribution of state. However, as tracking progresses the posterior probability distributions of partition states become multimodal, approaching uniform distribution in case of occlusion. In such circumstances, selecting the maximum likelihood particle for each partition would not necessarily output a maximized probability of state, or even a viable state for the object. Since sampling from multimodal distributions is an NP-hard problem, selection of particles becomes significantly harder as the number of partitions increases.

In this paper we tackle the problem of tracking general deformable objects through factored particle filtering. This task differs from tracking of articulated objects in the sense that the partition hierarchy is unknown and cannot be modeled offline, i.e. partition state dependencies have to be extracted dynamically during the progress of tracking. This model is required to be agile and adaptive to the time-varying deformation characteristics of the tracking object.

We present two contributions to particle filtering for tracking deformable objects with multiple partitions. First, we incorporate the dynamic dependencies of partitions in the transition model, with time-dependent correlations. We show that this strategy provides a transition model that is robust to occlusion and clutter. Our second contribution is a proposal function that provides a global estimate of viable states of the object.

For each partition, the proposal function is an exponential function of deviation from its original distances with all other partitions. This smooth function favors the original relative orientation of the partitions, but it also enables elastic deformations in existence of observational evidence.

This paper is organized as follows. In Section 2 we will introduce the original condensation algorithm and its extension with PFFL. We will detail our algorithm in Section 3, including the observation model, transition model and the proposal function. We will present our experimental results in Section 4 and conclude our work in Section 5.

2 Particle Filtering

We aim to track regions chosen on an object in a sequence of image frames. The states of these regions will be represented by *partitions* where the current state of partition i is denoted by x_i. In our implementation each partition has a reference image (template) and a state that consists of the partition's coordinates in the 2D image frame.

Probability density of state for partition i is approximated by a set of *particles*, where each particle k is a tuple of state and its probability $\{s_{ik}, \pi_{ik}\}$. Given an observation sequence $\mathbf{Z^t} = \{\mathbf{z}^0, \cdots, \mathbf{z}^t\}$ up to frame at time t, our goal is to estimate the posterior probability for all partitions $p(x_i^t|\mathbf{Z}^t)$ at time t.

When we model state transition as a Markov chain with independent observations, the posterior probability of state for a particle is defined as:

$$p(x_i^t|\mathbf{Z}^t) \propto p(\mathbf{z}^t|x_i^t)p(x_i^t|\mathbf{Z}^{t-1}) \tag{1}$$

$$p(x_i^t|\mathbf{Z}^{t-1}) = \sum_{x_i^{t-1}} p(x_i^t|x_i^{t-1})p(x_i^{t-1}|\mathbf{Z}^{t-1}) \tag{2}$$

Probability of the assumed state (1) x_i^t is proportional to the probability of reaching this state after observations $\mathbf{Z^{t-1}} = \{\mathbf{z}^0, \cdots, \mathbf{z}^{t-1}\}$ and the probability of the current observation in this state. Probability of transition from the previous probability density to the assumed state (2) is found over the dynamic transition model and the posterior probabilities of this partition's particles from the previous time step. The reader is referred to [2] for a complete derivation of posterior probability of state.

Note that probability density of partition x_i^t is approximated with a set of particles and their posterior probabilities. That is, the summation term in (2) incorporates possible state transitions between particles. For sake of implementation simplicity, we allow two particles to be at exactly the same state, and progress particles independently. In that case particle filtering equations (1) and (2) can be combined in one simple expression:

$$p(x_i^t|\mathbf{Z}^t) \propto p(\mathbf{z}^t|x_i^t)p(x_i^t|x_i^{t-1})p(x_i^{t-1}|\mathbf{Z}^{t-1}) \tag{3}$$

2.1 Condensation algorithm

Condensation algorithm [2] extends posterior probability density approximation to a sequence of frames. The iteration capitalizes on Markov chain assumption, i.e. the likelihood of a state depends on the probability density of the previous state, the probability of transition into the current state using the dynamic model and the current observation.

Given the posterior probability density for the previous time step $p(x_i^{t-1}|\mathbf{Z}^{t-1})$, which is represented with a set of particles and their probabilities $\{s_{ik}^{t-1}, \pi_{ik}^{t-1}\}$, $k = \{1, \cdots, M\}$, condensation algorithm proceeds as follows:

1. Sample M particles from $\{s_{ik}^{t-1}, \pi_{ik}^{t-1}\}$.
2. Predict a new state for each particle using the dynamic model $s_{ik}^t = p(x_i^t|x_i^{t-1} = s_{ik}^{t-1})$.
3. Evaluate each particle with respect to the current observation $\pi_{ik}^t \propto p(\mathbf{z}^t|x_i^t = s_{ik}^t)$.

The dynamic model in this algorithm consists of drift and diffusion. Drift is deterministic motion defined with velocity and acceleration, and is common for all particles of a partition. Diffusion on the other hand is Brownian motion; random movements that split particles at the identical state and enable search in the image space through random walk.

2.2 Particle filtering with factorized likelihoods

The original condensation algorithm was demonstrated on tracking control points of splines that define the contours of the tracked objects [2]. The states of these control points are statistically related. Particle Filtering with Factorized Likelihoods (PFFL) algorithm [11] reduces the dimensionality of the state-space by taking this fact into account. Note that a tracking object is segmented into N partitions, and the probability distribution of each partition is estimated by M particles. The original PFFL algorithm estimates the overall state of the tracking object by sampling each partition independently, such that the proposal function is maximized;

$$g(\mathbf{x}^t) = \prod_i p(x_i^t|\mathbf{Z}) \tag{4}$$

Patras and Pantic [11] constrained the transition model using training data of relative positions of facial features. Particles from posterior density are evaluated through Parzen density estimation. Translational invariance is achieved by use of relative positions to a stable feature in face (e.g. the nose tip). Scale invariance is achieved by scaling factors, such as the width of the mouth in the first image frame.

Pogalin [14] used the PFFL method and a 3D facial feature model as prior for gaze direction tracking. The head pose is estimated through tracked eye and mouth corners. Overall, Pogalin tracks 8 points, including the eye balls. Tracking process starts with an initial estimation of 3D facial feature coordinates from a stereo snapshot of the subject's head. Relative positions of the features are used as the reference shape model. Observation likelihood of a particle is decreased as the 3D distance between partition coordinates and the reference shape model increases.

An advantage of the PFFL algorithm is an initial simulation of dynamic transition. This stage enables identification of the most promising particles, i.e. particles that end up in states of high likelihood. The simulation stage is followed by evaluation, based on which the sampling probabilities of more promising particles are increased. This is followed by the stages of the classical condensation algorithm for each iteration. The PFFL algorithm is sketched with the following pseudocode.

Given probability density of partition x_i^{t-1} which is represented by particles $\{s_{ik}^{t-1}, \pi_{ik}^{t-1}\}$,

1. Sample particles from the posterior $p(x_i^{t-1}|\mathbf{Z}^{t-1})$
2. Propagate sampled particles via $p(x_i^t|x_i^{t-1})$ to obtain simulated, temporary particles s'_{ik}. Note that this step may incorporate only drift, or drift and diffusion combined.
3. Evaluate the likelihood of s'_{ik}; $\lambda_{ik} = p(\mathbf{z}^t|x_i^t = s'_{ik})$.
4. Resample M particles from $\{s_{ik}^{t-1}, \lambda_{ik}\pi_{ik}^{t-1}\}$ where M is the number of particles for partition i.
5. Propagate chosen particles via $p(x_i^t|x_i^{t-1})$ through drift and diffusion and obtain new particles s_{ik}^t. This step involves both drift and diffusion. Particles are typically upsampled during diffusion to expand the coverage of random walk.
6. Evaluate and assign weights to new particles.

$$\begin{aligned} w_{ik} &= p(\mathbf{z}^t|s_{ik}^t)\pi_{ik}^{t-1} \\ \pi_{ik}^t &= w_{ik}/\textstyle\sum_k w_{ik} \end{aligned} \tag{5}$$

Steps 1-6 are applied on each partition independently to obtain a particle based representation of posterior probability densities. When completed, we have N posteriors. Assuming that the partitions are initialized on an object, the next task is to choose particles from these densities to define the object's overall state. If the proposal distribution (4) is constructed from individual posteriors, N particles can be independently sampled from $p(x_i^t|\mathbf{Z}^t)$, $i = \{1, \cdots, N\}$.

3 Proposed Method

The PFFL algorithm leverages the dependent behavior of multiple partitions on a single object. In the general sense, the interdependency of partitions can be incorporated either as constraints in the dynamic model $p(x_i^t|x_i^{t-1})$, or in choosing N particles from the posterior of N partitions through proposal function $g(\mathbf{x}^t)$. In this study we first propose a transition model that provides an estimation for dependent drift of partitions. Second, we relieve the factorization constraint in the proposal function $g(\mathbf{x}^t)$ and propose a simple and computationally efficient model for modeling the joint probabilities of partitions.

In the next subsections we will describe our observation model, transition model, proposal function and present the complete algorithm.

3.1 Observation model

Tracking starts with manual initialization of partitions on 2D image plane. Initialization is setting of the states, in our case the coordinates, and the reference templates of partitions. A partition template is an image subregion with fixed window size extracted from the first frame of the video, and is *fixed* throughout tracking. When initialization is complete, all particles of a partition are placed at the starting coordinates of the partition and they share a common reference template.

The reference template $\mathbf{r}_i$ for partition i is a feature vector of RGB values in the chosen subregion of the first video frame. The current observation $\mathbf{o}_{ik}$ is the RGB values within the subregion centered at the coordinates of s_{ik}, the kth particle of partition i. We evaluate this particle by quantifying the difference between $\mathbf{r}_i$ and $\mathbf{o}_{ik}$. We adopt a measure that is invariant to global changes of intensity [11]:

$$d(\mathbf{r}_i, \mathbf{o}_{ik}) = \frac{\mathbf{r}_i}{E[\mathbf{r}_i]} - \frac{\mathbf{o}_{ik}}{E[\mathbf{o}_{ik}]} \tag{6}$$

The distance measure is defined as the L_2 norm of the distance vector $d(\mathbf{r}_i, \mathbf{o}_{ik})$. The likelihood of observation decreases exponentially with increasing distance:

$$p(\mathbf{z}^t|s_{ik}) = e^{-\frac{||d(\mathbf{r}_i,\mathbf{o}_{ik})||^2}{\sigma_o^2}} \quad (7)$$

where σ_o is a fixed scaling factor.

3.2 Transition model

Partitions exhibit correspondence in transition patterns. This is observed whether the partitions are chosen on a single object or on multiple objects. This situation is expected when a single object is being tracked. In the scenario of tracking multiple objects, we see that objects (e.g. people) often ensemble and move in the same direction or in a pattern.

In this study we are interested in tracking a single object with multiple partitions. However, even for a single object, correspondence of partition dynamics may suddenly and drastically change. A change in correspondences may be due to changes in the 3D orientation of the object or deformations within the object. Time dependency of correspondences in partition transitions becomes pronounced in facial feature tracking where 3D orientation change of the head and deformations on the face are simultaneously observed.

Condensation algorithm models transition with drift and diffusion components. Drift is stated to be deterministic, and is common for all particles of a partition. In our implementation drift for a partition is estimated based on correspondences with other partitions, which are represented as correlations of velocities, and dynamically updated in every time step.

Let us assume that we know the coordinates of all partitions in the last τ time steps. The dependency of velocity of partition i on partition j is determined with:

$$\rho_{ij} = \alpha_{ij}\frac{E[(v_i - \mu_i)(v_j - \mu_j)]}{\sigma_i \sigma_j} \quad (8)$$

where v_i and v_j are the velocities of partitions i and j, μ and σ are the mean and standard deviation of velocities in the last τ time steps. Note that the ratio in (8) is the correlation of velocities between partitions i and j. The velocity correlations for partition i are weighted with α_{ij} such that;

$$\sum_{j \neq i} \rho_{ij} = 1 \quad (9)$$

We start dynamic transition with a simulated random walk on a zero mean Gaussian distribution. This step provides an initial, crude estimate of partition velocities in the current time step. Next, drift for partition i is predicted using a weighted sum of correlations ρ_{ij} and estimated velocities v_j for all partitions $j = \{1, \cdots, N\}$, $j \neq i$. Note that drift is same for all particles of a partition.

Drift based on correlations provides us an estimate of partition states depending on their history of relative movements. We need to search further to account for inaccuracy of this estimation and to allow further deformations, i.e. relative motion of partitions within the object. We refine the state of a partition through diffusion, which is modeled with zero mean Gaussian distribution $G(0, \sigma_t)$. Overall, our transition model is defined as;

$$s_{ik}^t = s_{ik}^{t-1} + \sum_{j \neq i} \rho_{ij} v_j + G(0, \sigma_t) \quad (10)$$

In words, transition of kth particle of partition i from $t-1$ to t is a weighted combination of estimated partition velocities and a random search component. This approach perturbs the posterior distribution of partitions to comply better with the proposal function. Thus, partitions are constrained in their drift to maximize the number of particles of high likelihood in the proposal distribution.

3.3 Proposal function

Transition model progresses the particle clouds of each partition. Utilizing the observation model (6,7), we evaluate each particle and assign posterior probabilities. At this point, the probability density of state for each partition is approximated with its particle cloud.

One of the harder challenges in object tracking with multiple partitions is how to determine the state of the object from posterior probability approximations of its multiple partitions. The factorized likelihoods approach constructs a proposal distribution from individual posteriors (4) and attempts to sample particles that would maximize this function.

We encounter two pitfalls in practical implementation of the factorized likelihoods approach. First, particle distributions typically become multimodal as tracking progresses. The combinatorial nature of the problem makes it difficult to make the right choices for the partition states, especially when there are more than a few partitions to be tracked. Second, the proposal distribution is dependent on reference point(s) in reality, shifting in the state space with respect to their coordinates. Pogalin [14] chose partitions one by one, using the selected partitions as reference points. Patras [11] chose a simpler approach and exploited the nose tip as the reference point for proposal distribution in facial feature tracking. This approach requires an initialization to the subject's face and estimation of head orientation in each frame, as the relative distances of face features to nose tip alter with the head model and pose.

The state of each partition is represented by a cloud of particles. We can use this approximation to deal with the pitfalls of factorized likelihoods. We construct the proposal function dynamically in each frame, using only the L_2 norms from the partition coordinates for the given frame.

$$g(x_i^t) = \prod_{j \neq i} e^{-\frac{(||\mathbf{x}_i^t - \mathbf{x}_j^t|| - d_{ij})^2}{\sigma_p^2}} \tag{11}$$

When partition i is initialized, its distances to all other partitions d_{ij} are kept as reference distances. The probability of state of the partition decreases as its distance to other partitions deviates from the reference. This enables us to avoid the combinatorial explosion problem and the requirement of selecting a particular reference point. We evaluate the proposal function independently for partition i and choose particles from the distribution $g(\mathbf{x}_i^t)\pi_{ik}^t$, favoring the particles that conform to the shape topology of the object and with high posterior probability.

Figure (1) presents a 2D plot of the magnitude of the proposal function for the partition at the mouth of a gourami. When the scaling factor σ_p is low, relative motion between the partitions are more restricted and the object is considered to be more rigid. The partitions move more freely against each other as the scaling factor increases.

3.4 The algorithm

Given the parameters for the observation model, transition model and the proposal function, we can sketch the complete algorithm for tracking deformable objects with the following pseudocode.

Given the states of partitions x_i^{t-1} as approximations with particle clouds $\{s_{ik}^{t-1}, \pi_{ik}^{t-1}\}$;

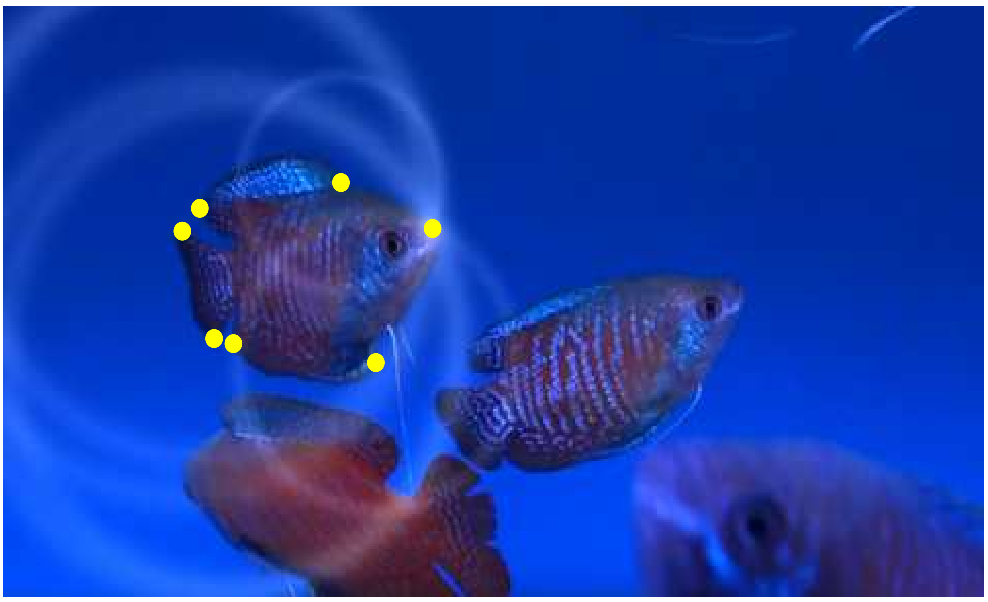
1. Simulate transition of all partitions with diffusion; $s'_{ik} = s_{ik}^{t-1} + G(0, \sigma_t)$.
2. Evaluate the likelihood of particles s'_{ik}; $\lambda_{ik} = p(\mathbf{z}^t | x_i^t = s'_{ik})$ using the observation model (7).
3. Estimate simulated states $x'_i = E[s'_{ik}]$ and velocities of partitions $v'_i = x'_i - x_i^{t-1}$.
4. Apply correlation based drift and diffusion (10) to particles s_{ik}^{t-1} and obtain new partitions s_{ik}^t. During diffusion, particles are upsampled to expand the coverage of random walk.
5. Use the observation model (7) to evaluate the conditional probability of observation $p(\mathbf{z}^t | s_{ik}^t)$ and estimate posterior probabilities π_{ik}^t (5).
6. Calculate the proposal function for each particle (11) and update their posterior probabilities with $g(x_i^t)\pi_{ik}^t$.
7. Sample M particles from $\{s_{ik}^t, \pi_{ik}^t\}$ where M is the number of particles for partition i.
8. Estimate the state of each partition as expected value of particles $x_i^t = E[s_{ik}^t]$.
9. Update velocity statistics μ_i and σ_i.

4 Experiments And Results

We tested the proposed algorithm against two other tracking algorithms. The first algorithm we chose for comparison was PFFL as it is a factored particle filtering method. We also tested our algorithm against Structure Preserving Object Tracking (SPOT) [23]. SPOT is a very recent tracking algorithm that is not based on particle filtering. It is a model-free tracker that leverages histogram-of-gradient (HoG) features as observation model, relative locations of objects as a proposal function and sliding-window exhaustive search to determine transition between frames.

In the first two tracking scenarios we test the capabilities of these three algorithms (the proposed algorithm, PFFL and SPOT) to deal with occlusion and clutter. In both scenarios the task is to track an object in a crowded scene of similar objects. As shown in Table (1), *platty* is a publicly available high-definition video of a crowded school of fish. *Skydiving* is a challenging video as it is low resolution and the skydivers' jumpsuits are very similar. In the third scenario we assess the tracking algorithms on sudden changes in transition dependencies of partitions. In this scenario we use a face video from the MMI facial expression database [24]. This video is carefully selected to exhibit relative motion and speed reversals (e.g. blinking, neutral-smile-neutral) together with rigid body motion (e.g. head movements) of the features.

Table (2) presents the values for all parameters that are used in our experiments. Note that we used 10 particles per partition for our tracking algorithm in all experiments. This figure is an order of magnitude lower than 100 particles used with the condensation algorithm [2] and significantly fewer than minimum of 50 particles suggested for tracking humans in open field [26, 27].



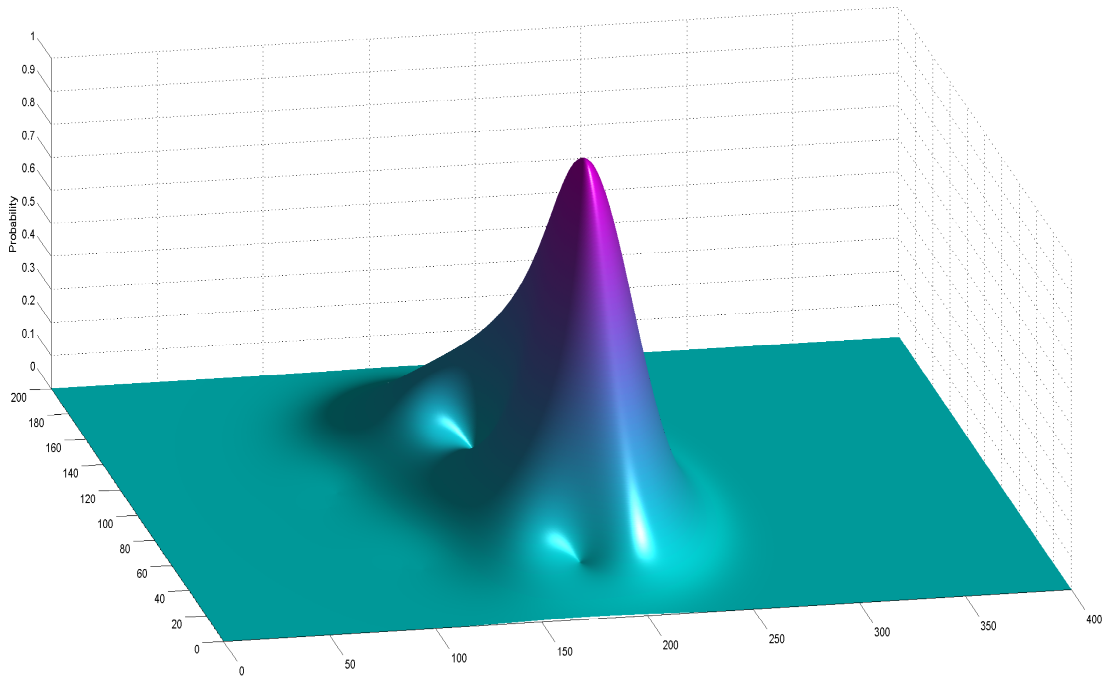


Fig. 1: Probability field for the partition at the mouth of a gourami. Top: The magnitude of the proposal function is depicted with shades of gray. For demonstration purposes, the scaling factor is chosen higher than the actual implementation. Bottom: 3D plot of the probability field.

Video	Resolution	Frames
Platty [25]	1080×1920	110
Skydiving [23]	480×656	300
Face [24]	576×720	127

Table 1 We evaluated tracking algorithms on three scenarios; platty, skydiving and face. The length of segment that is used in experiments is indicated by the number of frames.

Parameter	Sym.	Platty	Skydiv.	Face
Number of partitions	N	12	7	18
Number of particles	M	10	10	10
Obs. window size		21x21	21x21	21x21
Scaling factor (obs.)	σ_o	0.1	0.1	0.1
Scaling factor (trans.)	σ_t	10	5	5
Scaling factor (prop.)	σ_p	0.01	0.01	0.01

Table 2 Parameters for the proposed algorithm. Note that the same parameters are used for the PFFL algorithm with the exception of number of particles which was raised to 50.

These parameters were determined by trial and error on various videos. We observe that the set of parameters are very similar for best tracking performance, regardless of the nature of occlusion, clutter and motion of the object. The only parameter that is sensitive to the tracking task is observed to be σ_t, which is the scaling factor for transition. Note that this parameter governs the neighborhood for random walk and needs to be increased for tracking fast motion or tracking in high resolution, as is the case for the platty video.

4.1 Occlusion and clutter

Occlusion and clutter could easily lead to catastrophic results in tracking. Divergent drift due to occlusion is more pronounced when derivatives of error are exploited, such as in optical flow. In particle filtering, partition posterior probabilities resemble uniform distribution during occlusion. When other objects with similar characteristics are present in the image, particles may cling on nearest similar object and drift away from their true locations.

Figure (2) presents the results of tracking platty using the three algorithms. Each yellow dot on the platty represents a partition, whose posterior is estimated with 10 and 50 particles with the proposed algorithm and PFFL, respectively. Partitions on the nose and eye of the fish become occluded on frame 14 and stay occluded until frame 45. Occlusion of dorsal fin starts in frame 32 and continues until frame 57. Out of 12 partitions as many as 7 partitions are occluded in the progress of this video.

With the proposed algorithm all occluded partitions move with the rest of the partitions, resisting divergent drift. Once the occlusion is over, all partitions converge to their true positions. Convergence of partitions can be clearly observed on the nose and eye of platty between frames 60 and 90. All partitions are at their true coordinates after 90 frames, with slight error on the dorsal fin.

Divergent drift of partitions becomes noticeable as early as frame 15 with the PFFL algorithm. The partition on the dorsal fin attaches to the dorsal fin of a passing fish and drifts away. Similarly, the eye partition starts drifting away after frame 30. Only 7 out of 12 partitions are near their true coordinates at the end of the video. Most occluded partitions could not be recovered after the occlusion is over.

The benefit of exploiting relative locations of objects in the proposal function becomes apparent in the third experiment, in which we use the SPOT algorithm. This algorithm tracks platty until the end of the video despite occasional jumps produced by the sliding-window exhaustive search.

Figure (3) depicts the relative error for the proposed algorithm on selected features. We define error as the 2D distance between the tracking point and the true feature coordinates. True feature coordinates are marked manually by visual inspection. Error in pixels is divided by the approximate length of the fish to obtain unitless relative error. Increase in tracking error on the nose, eye and dorsal fin during occlusion are evident (frames 5-70 for the nose, 20-65 for the eye, frames 35-60 for the dorsal fin). Due to drift based on correlations, these features converge back to their true positions and are stable at the end of the experiment.

Figure (4) compares tracking performances of the proposed algorithm, PFFL and SPOT. In this figure, average relative error over 12 tracking points is plotted against the frame numbers. Tracking error in PFFL parts from the other two algorithms starting from frame 20. At the end of the experiment, average relative error with PFFL is about an order of magnitude higher than the other two algorithms. SPOT occasionally drifts due to the imprecision of its observation model. Both proposed algorithm and SPOT highly benefit from the configuration based proposal function and manage to keep the general structure of platty until the end of the experiment.

In the second experiment, we tested the three tracking algorithms on a cluttered scene and under low resolution. Figure (5) presents our results with the skydiving video, which also constitutes our longest experiment with 300 frames. Figure (6) depicts relative error for the head, left and right hand of the skydiver. For the great majority of 300 frames, the relative error for all three partitions is below 0.2, which corresponds to approximately 15 pixels.

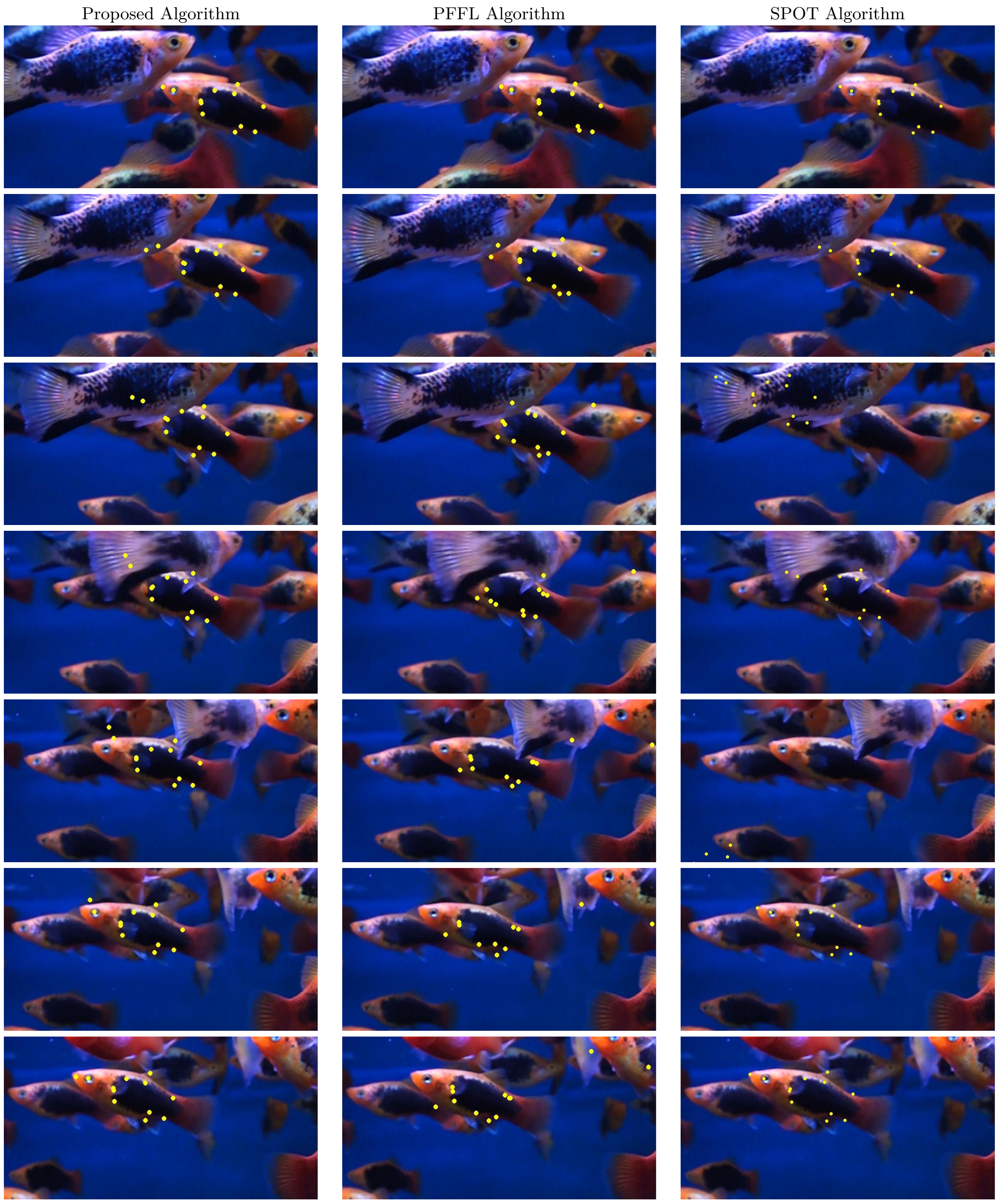


Fig. 2: Tracking a fish in a school and under occlusion. Left column: Proposed algorithm (10 particles per partition). Middle column: The PFFL algorithm (50 particles per partition). Right column: SPOT algorithm. From top to bottom: Frames 0, 15, 30, 45, 60, 75 and 90.

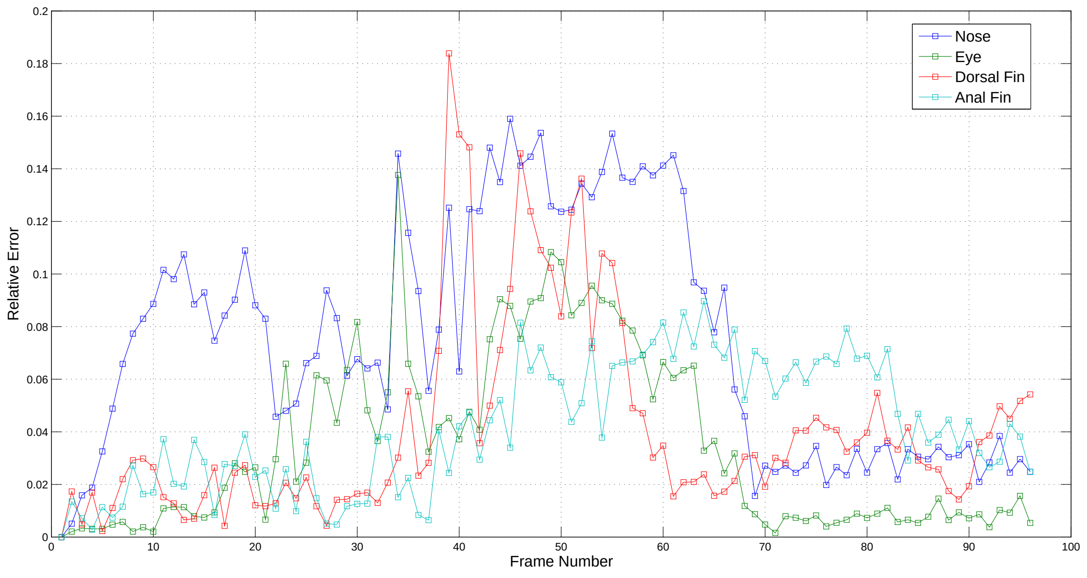


Fig. 3: Tracking error for significant features on platty with the proposed algorithm. The pixel error for each feature is divided by the approximate length of the fish in pixels.

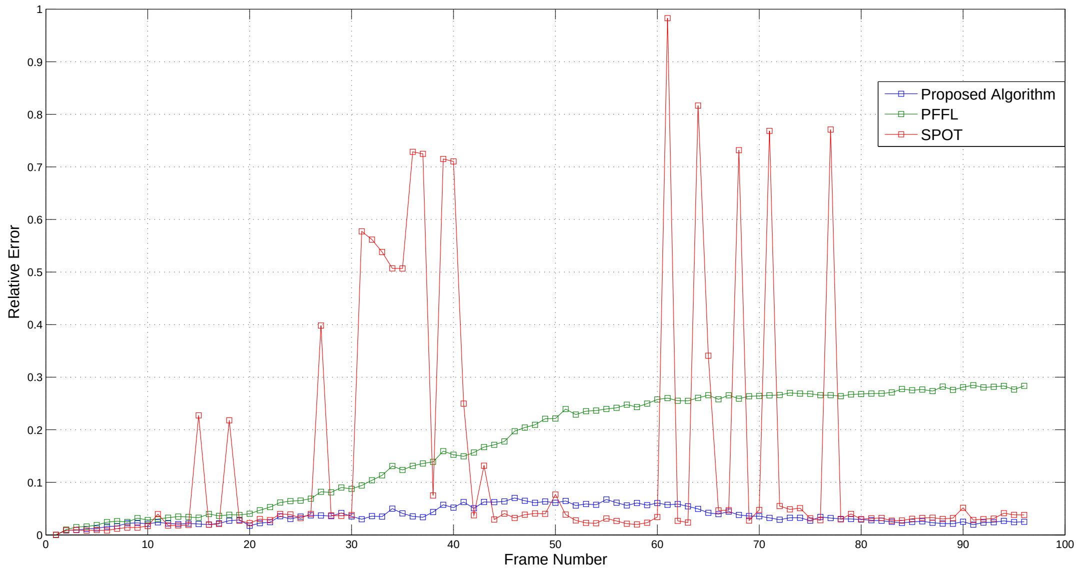


Fig. 4: Comparison of average error on platty for the proposed algorithm (10 particles), PFFL (50 particles) and SPOT. Data markers depict the average relative error for 12 partitions for each frame.

Figure (7) presents relative error comparison for the three tracking algorithms. With the proposed algorithm, we observe increase in the relative error when the background is cluttered (frames 50-80) and when the object scale changes (after frame 200). However, average relative error does not exceed 0.25 (one fourth of the skydiver body length) for the duration of the experiment. PFFL algorithm diverges after frame 50. SPOT algorithm cannot continue tracking beyond frame 75, where the skydiver moves past the white cloud in the background.

The proposed algorithm drives its robustness and capability to deal with occlusion and clutter from two contributions; drift with respect to correlated partition velocities and the proposal function. Drift of a partition is computed through its correlation of velocities with all other partitions. For that reason, under partial occlusion or clutter the partition moves smoothly, together with the other partitions. Under full occlusion of one or multiple partitions, the only implication for the location of the occluded partitions comes from the proposal function. The proposal function is modeled to bring forth the relative distances of partitions when no other evidence is available. Therefore under full occlusion of a partition, diffusion becomes ineffective and indifferent with approximately uniform distribution of likelihoods and the proposal function takes over tracking.

4.2 Drastic changes in relative speeds

Facial expression databases constitute great test beds for tracking algorithms. Relative motion of facial features are markedly dissimilar during smiling, frowning, surprise, anger, blinking, talking and so forth. We tested our algorithm on a video from the MMI facial expression database [24]. This video is carefully selected to exhibit relative motion and speed reversals (e.g. blinking, neutral-smile-neutral) together with rigid body motion (e.g. head movements) of the features.

Figure (8) demonstrates tracking of 18 facial features on a smiling video. Rows present the snapshots of tracking taken at frames 0, 30, 60, 90 and 120. Each frame of the video captures both the front and profile views of the subject. This situation makes the video a particularly tough test bed since groups of features may move in opposite directions, as observed with the tilt of the head from frame 30 to frame 60. Note that in our algorithm drift for each partition is determined through correlations with other partitions. As such, the statistics have to quickly adopt to the changes in partition dynamics.

	Proposed	PFFL	SPOT
Platty	3.01 ± 0.02	3.79 ± 0.04	0.16 ± 0.01
Skydiving	3.89 ± 0.02	4.47 ± 0.03	1.54 ± 0.01
Face	2.34 ± 0.02	3.39 ± 0.02	0.59 ± 0.01
Average	3.08 ± 0.64	3.88 ± 0.45	0.76 ± 0.59

Table 3 Frames per second for the proposed algorihtm, PFFL and SPOT.

Drift with respect to correlated partition speeds and distance-based proposal function handles drastic changes in relative speeds of partitions extremely well. Particles may momentarily drift away from their true coordinates as seen on lip corners in frame 60. However the probability densities of particles quickly converge to their true states and tracking continues stably until the end of the video. With the PFFL algorithm (middle column), partitions on the left lip corner and the upper lip drift away in frame 90 and do not converge back to their true states. With SPOT (right column), eye corner in the profile view diverges towards the end of the experiment.

The performance of the proposed algorithm can be observed more clearly in Figure (9). Top and bottom plots in this figure depict the relative error for selected frontal and profile features, respectively. On the profile of the subject (bottom), the eye corner, upper lip and lower lip features are tracked precisely throughout the experiment. On the frontal view, right lip corner is not tracked accurately during smile (frames 56-95). This is primarily due to the fact that its relative motion is not correlated with any particular feature on the front or profile of the subject. However, this error is handled by the proposal function and the partition converges to its true position towards the end of the smile. We observed that three eye blinks were successfully handled by the proposed algorithm.

Figure (10) compares the overall performance of the proposed algorithm with the PFFL and SPOT algorithms. Mean relative error is calculated over all 18 partitions for each frame. As observed in the figure, the performance of the proposed algorithm parts from the PFFL algorithm after frame 85. Our tracking results are consistently more accurate than both PFFL and SPOT until the end of the experiment.

4.3 Time requirements

We measured running times of the three tracking algorithms in each experiment. Table (3) presents approximate speeds on a single core 2.2 GHz processor. The implementations of the tracking algorithms were done in MATLAB.

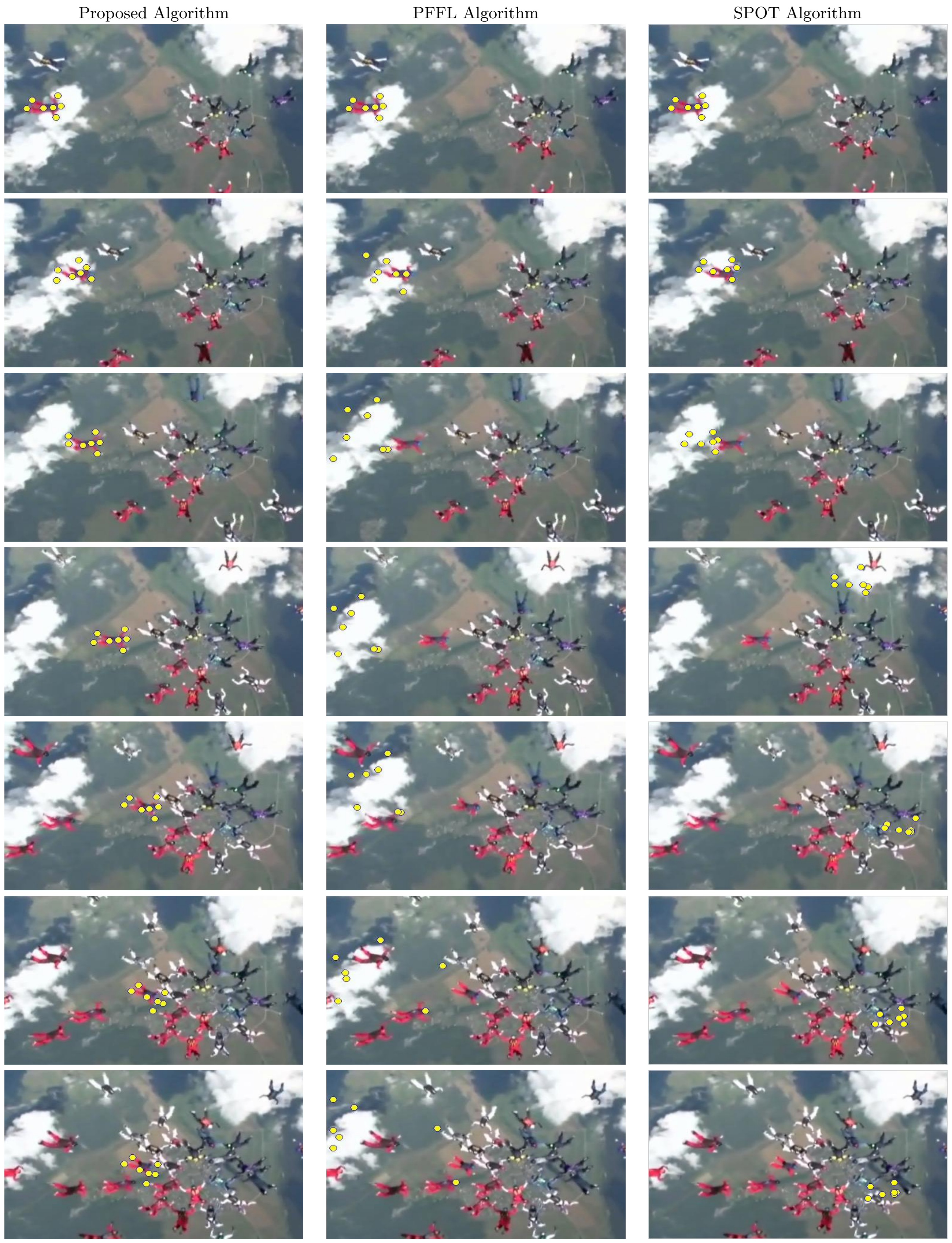


Fig. 5: Tracking a skydiver. Left column: Proposed algorithm (10 particles per partition). Middle column: The PFFL algorithm (50 particles per partition). Right column: SPOT algorithm. From top to bottom: Frames 0, 50, 100, 150, 200, 250 and 300.

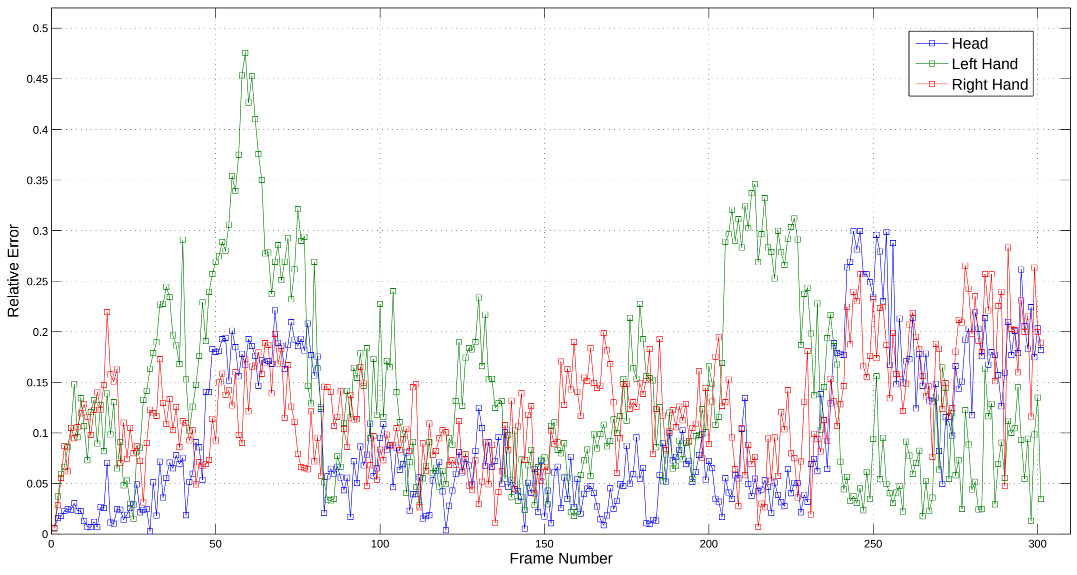


Fig. 6: Tracking error for head, left hand and right hand for the skydiver. The pixel error for each feature is divided by skydiver's approximate length in pixels.

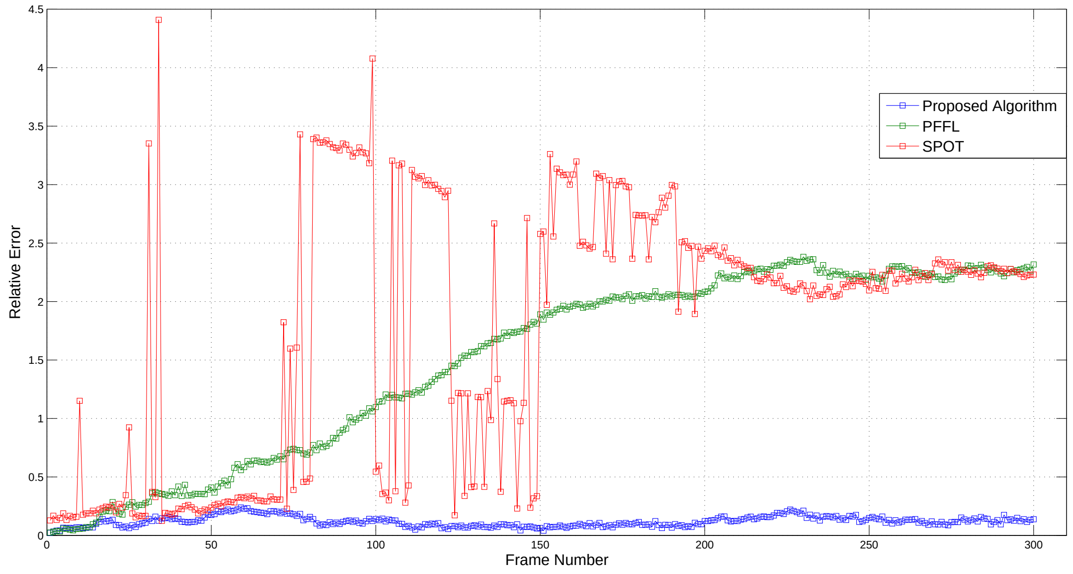


Fig. 7: Comparison of average error on skydiver for the proposed algorithm (10 particles), PFFL (50 particles) and SPOT. Data markers depict the average relative error for 7 partitions for each frame.

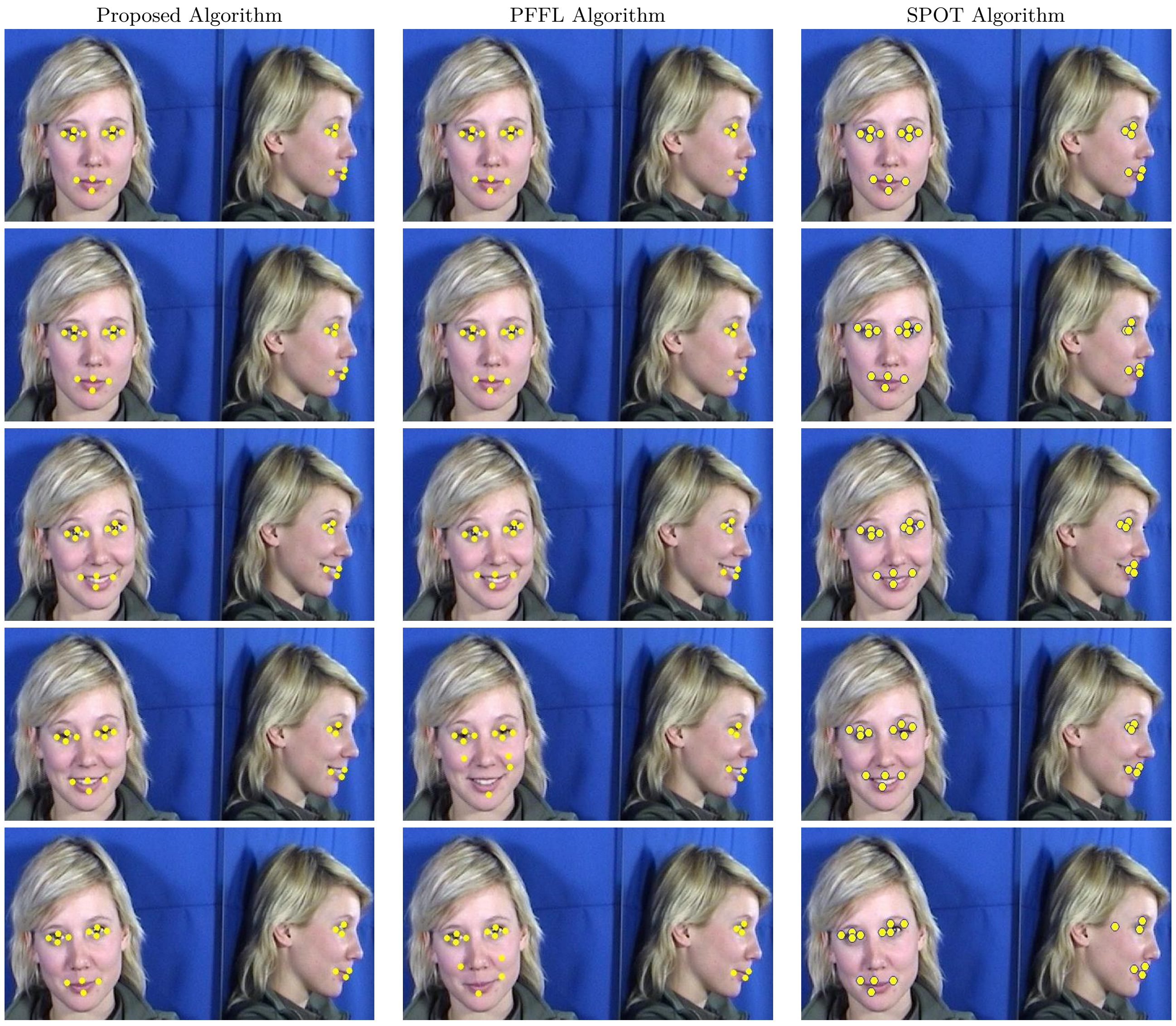


Fig. 8: Tracking facial features. Left column: Proposed algorithm (10 particles per partition). Middle column: The PFFL algorithm (50 particles per partition). Right column: SPOT algorithm. From top to bottom: Frames 0, 30, 60, 90 and 120.

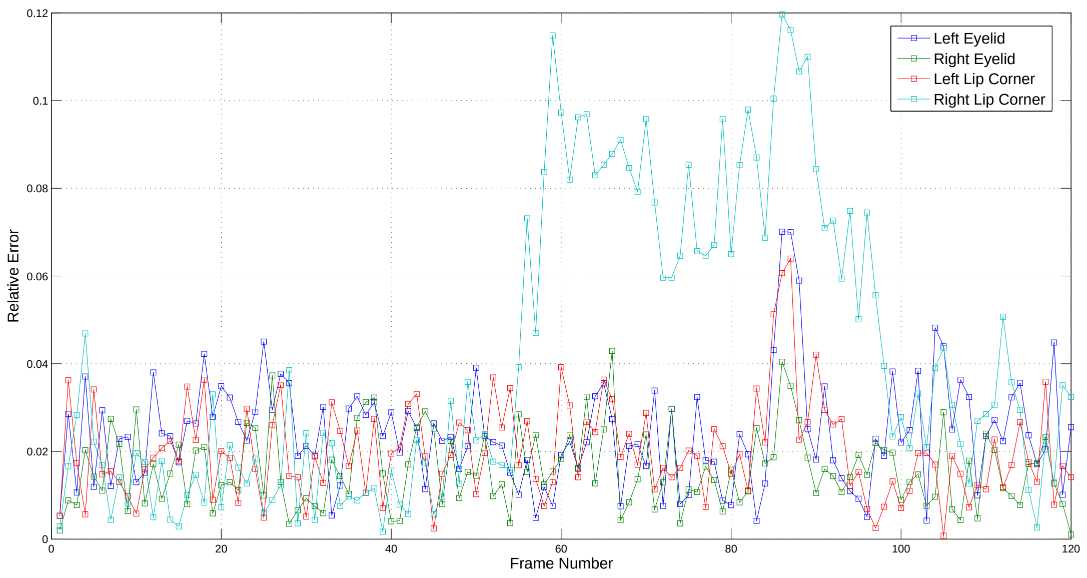
The speed of the proposed algorithm varies between 2.3 and 3.6 fps. The major factor in speed variations among experiments is the changing number of partitions. Note that the number of particles per partition is fixed (10 for the proposed algorithm, 50 for PFFL) in all three experiments. The time cost of tracking a partition of 10 particles is approximately 30 milliseconds for the proposed algorithm.

The SPOT algorithm is considerably slower than the other two algorithms. Running time of SPOT depends mostly on the resolution of the video as it calculates HoG features for the entire frame. For that reason, we see that SPOT is significantly slow on the high resolution platty video.

We observe that the proposed algorithm is considerably faster than SPOT. However, the cost of calculation of speed correlations, correlation based drift, and distance based proposal function makes our algorithm up to 30% slower than PFFL.

5 Conclusion

In this paper we present two contributions to tracking multiple partitions of a deformable object with particle filtering. Our first contribution is in *drift*; the deterministic motion of a partition with respect to other partitions. The second contribution is in selection of



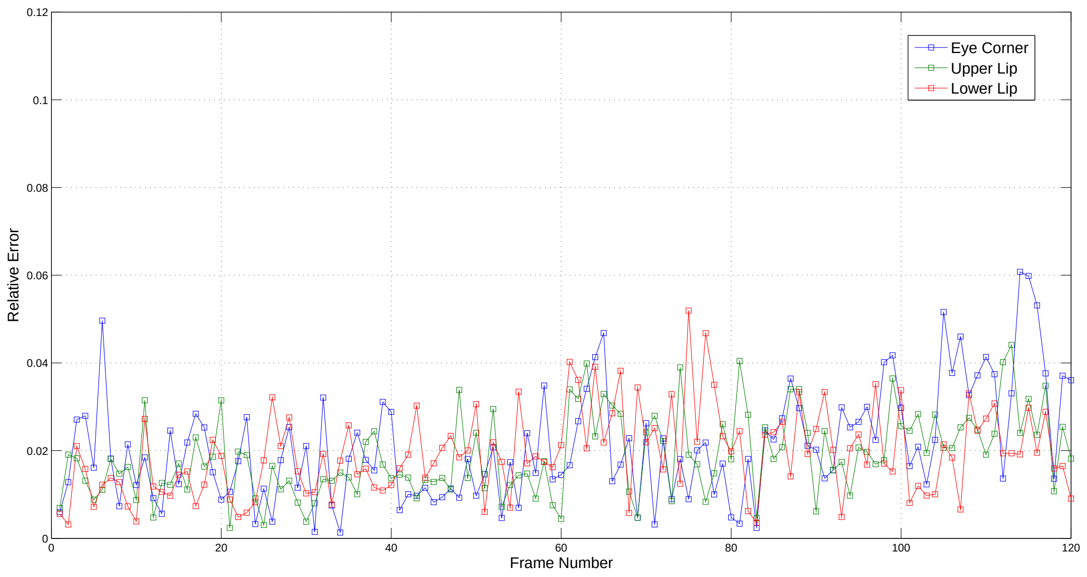


Fig. 9: Tracking error for significant facial features. The pixel error for each feature is divided by the approximate width of the face in pixels. Top: Frontal features. Bottom: Profile features.

particles from probability densities of partitions such that the set of particles represent a viable and likely current state of the object.

We propose an algorithm that extends particle filtering to tracking of objects that deform or change 3D orientation during their courses of motion. An iteration of filtering starts with a simulated random walk followed by an evaluation through observation model. This stage is not meant to be precise or even close to the actual solution, but only to provide an estimation of velocity of the partitions. The outcome of the simulation stage is used in estimating the interdependent motion (drift) of partitions.

The dependency of partition drift on deformable objects vary in space and time. We propose an online algorithm that extracts this interdependency using correlation statistics. The estimated velocities of partitions in the last k frames are used for deriving the correlation of

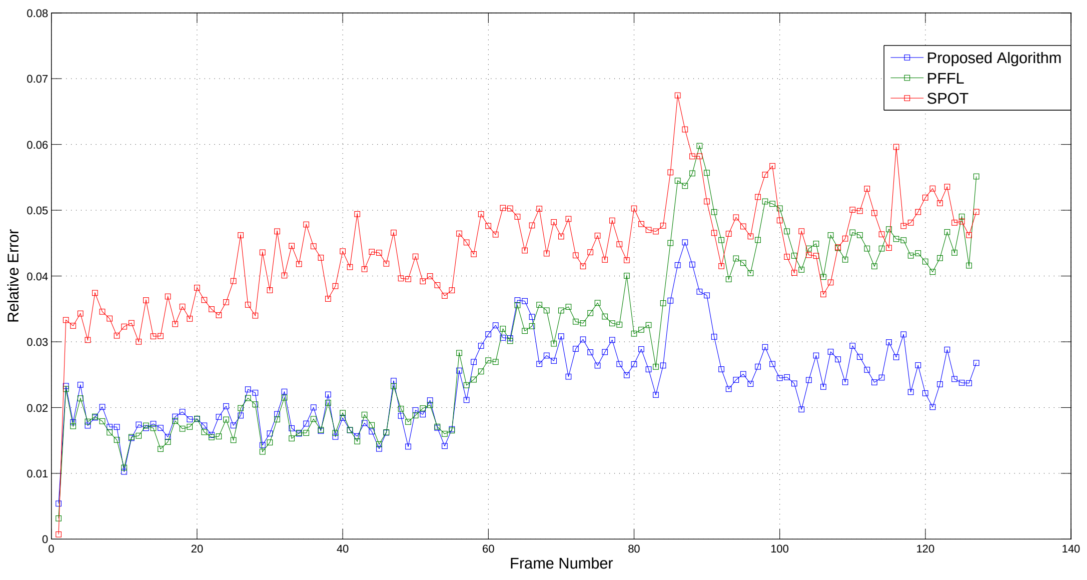


Fig. 10: Comparison of average error on facial features for the proposed algorithm (10 particles), PFFL (50 particles) and SPOT. Data markers depict the average relative error for 18 partitions for each frame.

velocities between each pair of partitions. For one partition, its normalized velocity correlations with all other partitions are used as weights in determining its drift. The dot product of the weights and the initial estimate of partition velocities obtained through the simulation stage provides the deterministic drift for the partition.

In modular or partitioned sampling, the state of the tracking object can be estimated by selecting the particles with highest likelihood for each partition. It is easy to visually verify this assumption in the beginning stages of tracking, where all partitions exhibit a unimodal distribution of state. As the tracking proceeds, probability densities of partitions become multimodal, approaching uniform distribution when there is occlusion. In this stage, selecting a set of particles that constructs a viable object state becomes a combinatorial and NP-hard problem. This challenge is elevated as the number of partitions increases and combinatorial explosion becomes a concern.

Here we propose a computationally simple proposal function that provides an estimate of viable states of the object. The probability distribution of proposal for each partition is modeled as an exponential function of deviation of distance with all other partitions. This smooth proposal function operates like a spring, allowing non-rigid motion of partitions or the deformation of the object when there is observational evidence, meanwhile favoring the original relative orientation of the partitions.

The proposed algorithm displayed excellent performance under occlusion, clutter and non-rigid deformations. Correlation based drift estimation provides robustness and efficiency when occlusion and clutter confuses the tracker. Relative distance based proposal function handles deformations robustly, favoring the original orientations of partitions meanwhile enabling non-rigid motion in existence of observational evidence.

Acknowledgements

This research is part of project “Expression Recognition based on Facial Anatomy”, grant number 109E061, supported by The Support Programme for Scientific and Technological Research Projects of The Scientific and Technological Research Council of Turkey (TÜBİTAK).

In comparative evaluation of the tracking algorithms we utilized the SPOT tracking code that was made publicly available by researchers Lu Zhang and Laurens van der Maaten.

A special thanks to *Fish Species* who generously provided the high definition aquarium videos used in our experiments (http://www.fish-species.org.uk).

References

1. R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Basic Eng-T ASME*, no. 82 (Series

D), pp. 35–45, 1960.

2. M. Isard and A. Blake, “Condensation - conditional density propagation for visual tracking,” *Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, 1998.
3. M. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filtering,” *J. Amer. Statist. Assoc.*, vol. 94, pp. 590–599, 1999.
4. A. Blake, R. Curwen, and A. Zisserman, “A framework for spatio-temporal control in the tracking of visual contours,” *Int. J. Comput. Vision*, vol. 11, no. 2, pp. 127–145, 1993.
5. J. MacCormick, “Probabilistic modeling and stochastic algorithms for visual localisation and tracking,” Ph.D. dissertation, University of Oxford, 2000.
6. C. Hue, J.-P. Le Cadre, and P. Perez, “Tracking multiple objects with particle filtering,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, no. 3, pp. 791–812, 2002.
7. J. MacCormick and A. Blake, “A probabilistic exclusion principle for tracking multiple objects,” *Int. J. Comput. Vision*, vol. 39, no. 1, pp. 57–71, Aug. 2000.
8. J. MacCormick and M. Isard, “Partitioned sampling, articulated objects, and interface-quality hand tracking,” in *Proceedings of the 6th European Conference on Computer Vision-Part II*, ser. ECCV ’00. London, UK, UK: Springer-Verlag, 2000, pp. 3–19.
9. J. Deutscher and I. Reid, “Articulated body motion capture by stochastic search,” *Int. J. Comput. Vision*, vol. 61, no. 2, pp. 185–205, Feb. 2005.
10. B. Wu and R. Nevatia, “Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors,” *Int. J. Comput. Vision*, vol. 75, no. 2, pp. 247–266, Nov. 2007.
11. I. Patras and M. Pantic, “Particle filtering with factorized likelihoods for tracking facial features,” in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, may 2004, pp. 97–102.
12. X. Dai and G. D. Hager, “Efficient particle filtering using ransac with application to 3d face tracking,” *Image Vision Comput.*, vol. 24, pp. 581–592, 2006.
13. M. Valstar and M. Pantic, “Fully automatic facial action unit detection and temporal analysis,” in *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, ser. CVPRW ’06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 149–154. [Online]. Available: http://dx.doi.org/10.1109/CVPRW.2006.85
14. E. Pogalin, A. Redert, I. Patras, and E. Hendriks, “Gaze tracking by using factorized likelihoods particle filtering and stereo vision,” in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, june 2006, pp. 57–64.
15. V. Lepetit, J. Pilet, and P. Fua, “Point matching as a classification problem for fast and robust object pose estimation,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, 2004, pp. II–244 – II–250 Vol.2.
16. N. M. Oliver, B. Rosario, and A. P. Pentland, “A bayesian computer vision system for modeling human interactions,” *IEEE T. Pattern Anal.*, vol. 22, no. 8, pp. 831–843, 2000.
17. L. Taycher, J. W. Fisher, III, and T. Darrell, “Combining object and feature dynamics in probabilistic tracking,” *Comput. Vis. Image Underst.*, vol. 108, no. 3, pp. 243–260, Dec. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2006.11.022
18. J. Triesch and C. Von Der Malsburg, “Democratic integration: Self-organized integration of adaptive cues,” *Neural Comput.*, vol. 13, no. 9, pp. 2049–2074, Sep. 2001.
19. L. Bréthes, F. Lerasle, P. Danés, and M. Fontmarty, “Particle filtering strategies for data fusion dedicated to visual tracking from a mobile robot,” *Mach. Vision Appl.*, vol. 21, no. 4, pp. 427–448, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1007/s00138-008-0174-7
20. E. Erdem, S. Dubuisson, and I. Bloch, “Fragments based tracking with adaptive cue integration,” *Comput. Vis. Image Underst.*, vol. 116, no. 7, pp. 827–841, Jul. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2012.03.005
21. S. J. McKenna and H. Nait-Charif, “Tracking human motion using auxiliary particle filters and iterated likelihood weighting,” *Image Vision Comput.*, vol. 25, no. 6, pp. 852–862, Jun. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2006.06.003
22. S. Coşar and M. Çetin, “A graphical model based solution to the facial feature point tracking problem,” *Image Vision Comput.*, vol. 29, no. 5, pp. 335–350, Apr. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2010.12.001
23. L. Zhang and L. van der Maaten, “Structure preserving object tracking,” in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR ’13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 1838–1845. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2013.240
24. M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat, “Web-based database for facial expression analysis,” in *Proceedings of IEEE Int’l Conf. Multimedia and Expo (ICME’05)*, Amsterdam, The Netherlands, July 2005, pp. 317–321.
25. “Fish species,” http://http://fish-species.org.uk/, accessed: 2014-04-17.
26. U. Filters, “Tracking football player movement from a single moving camera,” in *Visual Media Production, 2006. CVMP 2006. 3rd European Conference on*, 2006, pp. 29 –37.
27. W. Hassan, N. Bangalore, P. Birch, R. C. D. Young, and C. R. Chatwin, “An adaptive sample count particle filter.” *Comput. Vis. Image Underst.*, vol. 116, no. 12, pp. 1208–1222, 2012.