Segmentation Based Particle Filtering for Real-Time 2D Object Tracking

Vasileios Belagiannis[1], Falk Schubert[2], Nassir Navab[1], and Slobodan Ilic[1]

[1] Computer Aided Medical Procedures, Technische Universität München, Germany
{belagian,navab,slobodan.ilic}@in.tum.de
[2] EADS Innovation Works, Germany
falk.schubert@eads.net

Abstract. We address the problem of visual tracking of arbitrary objects that undergo significant scale and appearance changes. The classical tracking methods rely on the bounding box surrounding the target object. Regardless of the tracking approach, the use of bounding box quite often introduces background information. This information propagates in time and its accumulation quite often results in drift and tracking failure. This is particularly the case with the particle filtering approach that is often used for visual tracking. However, it always uses a bounding box around the object to compute features of the particle samples. Since this causes the drift, we propose to use segmentation for sampling. Relying on segmentation and computing the colour and gradient orientation histograms from these segmented particle samples allows the tracker to easily adapt to the object's deformations, occlusions, orientation, scale and appearance changes. We propose two particle sampling strategies based on segmentation. In the first, segmentation is done for every propagated particle sample, while in the second only the strongest particle sample is segmented. Depending on this decision there is obviously a trade-off between speed and performance.
We perform an exhaustive quantitative evaluation on a number of challenging sequences and compare our method with the number of state-of-the-art methods previously evaluated on those sequences. The results we obtain outperform majority of the related work, both in terms of the performance and speed.

1 Introduction

Visual object tracking is one of the major research problems in Computer Vision. It is essential for numerous applications, such as surveillance [1], action recognition [2] or augmented reality [3]. One of the classical approaches for object tracking is particle filtering. It generalizes well to any kind of objects, models well non-Gaussian noise and is able to run in real-time. The observation models that have been used with particle filtering are either colour histograms [4] or histograms of oriented gradients [5]. These observation modes are computed from the bounding boxes surrounding the target object. While using bounding boxes is fast and convenient, they often capture undesirable background information

Fig. 1: Tracking results for some of our evaluation sequences. From top to the bottom row respectively sequences are named: *Mountain-bike*, *Entrance*, *UAV*, *Cliff-dive 1*. The *Entrance* sequence has been captured with a stationary camera while in the other three sequences both the object and camera are moving

as most objects do not fit into a rectangle very well. This information is further propagated to all sample particles and often causes drift. This is particularly true for deformable objects, like humans, where the bounding box sometimes includes very large portions of the background.

The recent trend in visual tracking is related to learning the object's appearance. The tracking then becomes a classification problem where the goal is to discriminate the object of interest from the background [6]. The appearance of the object can be learned offline or online. These approaches are traditionally called *tracking-by-detection* or online learning approaches and have performed very well on demanding tracking scenarios, e.g. sport activities, pedestrian tracking or vehicle tracking. Although they are often robust against occlusions, deformations, orientation, scale and appearance changes, their computational cost makes most of them inefficient for real-time applications. In addition, the presence of false positive detections causes drifting. The drifting is closely related to the area from which the object's features are extracted. It is usually determined from a rectangular bounding box. However, the object does not usually fit perfectly inside the box, so the additional background information is included in the extracted features. For instance, this results in learning background in the trackers based on the online learning of the object appearance. Again, the presence of the background information becomes more critical for deformable objects where

the bounding box always includes that type of noise. To overcome this problem Godec et al. [7] recently proposed an approach that removes a bounding box constraint and combines segmentation and online learning. However, due to the very expensive learning procedure based on Hough forests the efficiency of this tracker is far from real time.

Our objective is to overcome majority of these limitations and provide a general purpose tracker that can track arbitrary objects whose initial shape is not a priori known in challenging sequences in real-time. These sequences contain clutter, partial occlusions, rapid motion, significant viewpoint and appearance changes (Fig. 1). We propose to use the standard particle filter approach based on colour and gradient histograms and incorporate the object shape into the state vector. Since the classical particle filter based on bounding box surrounding particle samples drifts due to sometime abrupt amount of captured background, we propose to use segmentation at the particle sample locations propagated by a basic dynamic motion model. This allows having particle samples of arbitrary shapes and collecting more relevant regions features than when the bounding box is used. Consequently the object state vector strongly depends on the object's shape. Relying on segmentation allows the tracker to easily adapt to the object's deformations, occlusions, orientation, scale and appearance changes. We propose two particle sampling strategies based on segmentations. In one case the segmentation is done for every propagated particle sample and therefore is more robust to large displacements, scales and deformations, but it is more time consuming. The other strategy is to do the segmentation on the particle sample with the highest importance weight and propagate its shape to all other samples. This is definitely less robust and more critical in difficult sequences where object shape and position change dramatically from frame to frame, but in all other sequences, where this is not the case, is sufficient and comes with the great computational complexity reduction leading to very fast runtime of up to 50 fps. Depending on this decision, there is obviously a trade-off between speed and performance.

We tested our method on a number of available sequences used by the recent state-of-the-art methods. We demonstrated the advantage of our method over normal particle filtering based on bounding box and made a comparison with many state-of-the-art trackers. This analysis showed that in many cases our method outperforms related approaches both in terms of speed and performance.

1.1 Related Work

There is notable literature on visual object tracking. Given the limited space, we focus on work mostly related to particle filtering and learning-based approaches as well as methods that do not rely on rectangular bounding boxes. Starting from the probabilistic methods, Isard and Blake [8] introduced the particle filter, namely condensation algorithm, for tracking curves. Later on, the method was also applied to colour based tracking [9]. Similarly, Pérez et al. [4] proposed a colour histogram based particle filtering approach. However, the colour distribution fails to describe an object in situations where the object is of a similar

colour as the background. For that reason, Lu et al. [5] incorporated a gradient orientation histogram in the particle filter. The most common particle filtering algorithm, the bootstrap filter [10], has been combined with a classifier [11] in order to be created an advanced motion model. All these methods rely on bounding boxes for sampling and therefore are sensible to the particle samples erroneously taken from the background. A more recent approach combines an off-line and an online classifier in the bootstrap filter's importance weight estimation [1]. In all cases, incorporating a classifier into the particle filter has an important impact on the runtime.

In the domain of a unified tracking and segmentation, the object is presented from a segmented area instead of a bounding box. Particularly impressive is the probabilistic approach of Bibby and Reid [12]. They have combined the bag-of-pixels image representation with a level-set framework, where the likelihood term has been replaced from the posterior term. Even though this approach adapts the model online and is not based on the bounding box, it is susceptible to the background clutter and occlusions. Chockalingam et al. [13] divided the object into fragments based on level-sets as well. Recently, Tsai et al. [14] have proposed a multi-label Markov Random Field framework for segmenting the image data by minimizing an energy function, but the method works only offline. The complexity of all these methods increases their computational cost significantly. In addition to the object segmentation, Nejhum et al. [15] have used a block configuration for describing the object. Each block corresponds to an intensity histogram and all together share a common configuration. This representation forms the searching window which is iteratively updated. Nevertheless, the bounding box representation is still present but in a small scale.

The first work on learning-based approaches was published by Avidan [6] and Javed et al. [16], where tracking is defined as a binary classification problem. A set of weak classifiers is trained online and afterwards boosted to discriminate the foreground object from the background. The idea of online training has been continued by Grabner et al. [17] for achieving a real-time performance in a semi-supervised learning framework. In this approach, the samples from the initialization frame are considered as positive for online training and during the runtime the classifier is updated with unlabelled data. Babenko et. al [18] have proposed a multiple instance learning (MIL) approach for dealing with the incorrectly labelled data during the training process. The MIL classifier is trained with bags of positive and negative data, where a positive bag contains at least one positive instance. More recently, Kalal et al. [19] have combined the KLT tracker [20] with an online updated randomized forest classifier for learning the appearance of the foreground object. The tracker updates the classifier and the classifier reinitializes it in case of a drift. Similarly in [21], the appearance model of the tracker evolves during time. All the above approaches present mechanisms for preventing the drifting effect in some form. However, they are all trained with data extracted from a bounding box. As a result, background information is highly probable to penetrate into the training process which will eventually lead to drift assuming arbitrarily shaped objects.

Godec et al. [7] have gone a step further into online learning by removing the rectangular bounding box representation. They have employed the Hough Forests [22] classification framework for online learning. In this approach, the classification output initializes a segmentation algorithm for getting a more accurate shape of the object. The approach is relatively slow, but it delivers promising results on demanding tracking sequences. In the proposed work, we similarly make use of the segmentation concept as well but we incorporate this into a much faster particle filter tracker instead of using a non-bounding box classification approach.

2 Particle Filter Based Visual Object Tracking

The particle filter has shown to be a robust tracking algorithm for deformable objects with non-linear motion [8]. The tracking problem is defined as a Bayesian filter that recursively calculates the probability of the state $\mathbf{x}_t$ at time t, given the observations $\mathbf{z}_{1:t}$ up to time t. This requires the computation of the (probability density function) pdf $p(\mathbf{x}_t \mid \mathbf{z}_{1:t})$. It is assumed that the initial pdf $p(\mathbf{x}_0 \mid \mathbf{z}_0) = p(\mathbf{x}_0)$ of the state vector, also known as the prior, is available. $\mathbf{z}_0$ is an empty set indicating that there is no observation. In our problem the state consists of the object's shape S and 2D position of the shape's centre of mass x_c, y_c and is defined as $\mathbf{x}_t = [x_c, y_c, S]^T$. The prior distribution is estimated from the initial object shape. The initial shape can be either manually drawn or estimated from segmenting a bounding box which surrounds the object. Finally, the pdf $p(\mathbf{x}_t \mid \mathbf{z}_{1:t})$ can be computed from the Bayesian recursion, consisting of two phases called prediction and update. Assuming that the pdf $p(\mathbf{x}_{t-1} \mid \mathbf{z}_{1:t-1})$ is available and the object state evolves from a transition model $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{v})$, where $\mathbf{v}$ is a noise model, then in the prediction phase the prior pdf $p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1})$ at time t can be computed using the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1} \tag{1}$$

The probabilistic model of the state evolution $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ is defined by the transition model. When at time t an observation $\mathbf{z}_t$ becomes available, the prior can be updated via Bayes' rule:

$$\begin{aligned} p(\mathbf{x}_t \mid \mathbf{z}_{1:t}) &= \frac{p(\mathbf{z}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1})} = \\ &= \frac{p(\mathbf{z}_t \mid \mathbf{x}_t) \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{z}_{t-1}) \mathrm{d}\mathbf{x}_{t-1}}{\int p(\mathbf{z}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}) \mathrm{d}\mathbf{x}_t} \end{aligned} \tag{2}$$

where the likelihood $p(\mathbf{z}_t \mid \mathbf{x}_t)$ is defined by the observation model $\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{n}_t)$ with known statistics $\mathbf{n}_t$. In the update phase, the observation $\mathbf{z}_t$ is used to update the prior density in order to obtain the desirable posterior of the current state. The observation in our method comes from colour $p(\mathbf{z}_t^{col} \mid \mathbf{x}_t)$ and gradient orientation $p(\mathbf{z}_t^{or} \mid \mathbf{x}_t)$ histograms.

Since posterior density cannot be computed analytically, it is represented by a set of random particle samples $\{\mathbf{x}_i^t\}_{i=1\cdots N_s}$ with associated weights $\{\mathbf{w}_i^t\}_{i=1\cdots N_s}$. The most standard particle filter algorithm is Sequential Importance Sampling (SIS). Theoretically, when the number of samples becomes very large, this so called Monte Carlo sampling becomes an equivalent representation to the usual analytical description of the posterior pdf. Each particle sample represents a hypothetical object state and it is associated with an importance weight. The calculation of the weight is based on the observation likelihood and weight from the previous time step.

However, a common problem with the SIS particle filter algorithm is the degeneracy phenomenon. This means that after a few iterations the majority of particles will have negligible weight. To overcome this problem the bootstrap filter, which is based on the Sampling-Importance-Resampling (SIR) technique, aims to remove low importance samples from the posterior distribution. When the number of particle samples with high importance weight drops under a constant threshold, the resampling step is executed. There, every sample contributes to the posterior with proportion to its importance weight. The weight estimation is given by:

$$\mathbf{w}_t^{(i)} = \mathbf{w}_{t-1}^{(i)} \cdot p(\mathbf{z}_t \mid \mathbf{x}_t^{(i)}) = \mathbf{w}_{t-1}^{(i)} \cdot p(\mathbf{z}_t^{col} \mid \mathbf{x}_t^{(i)}) p(\mathbf{z}_t^{or} \mid \mathbf{x}_t^{(i)}), \sum_{i=1}^{N_s} \mathbf{w}_t^{(i)} = 1 \quad (3)$$

After the resampling step, the samples are equally weighted with $\mathbf{w}_{t-1}^{(i)}$ being constant (i.e. $1/N_s$). The importance weight calculation cost is increased linearly with the number of the the particle samples. Detailed description and discussion of particle filtering can be found in [10]. Next, we detail elements of our particle filtering approach including observation and transition model as well as the segmentation of the particle samples.

2.1 Observation Model

Our observation model relies on two components, the colour and gradient orientation histograms. Concerning the colour information, we use the HSV space similar to [4] since it is less sensitive to illumination changes. The colour distribution is invariant to rotation, scale changes and partial occlusion. For the gradient orientation histogram, we compute the histogram of oriented gradients (HOG) descriptor [23]. The strong normalization of the descriptor makes it invariant to illumination changes.

The likelihood of the observation model $p(\mathbf{z}_t \mid \mathbf{x}_t^{(i)})$ for each particle sample $i = 1 \dots N_s$ is calculated from the similarity between the current $\mathbf{q}(\mathbf{x}_{t-1}) = \{q_n(\mathbf{x}_{t-1})\}_{n=1,\dots,N_c}$ and the predicted state $\mathbf{q}(\mathbf{x}_t) = \{q_n(\mathbf{x}_t)\}_{n=1,\dots,N_c}$ distributions represented by colour histograms, where N_c is the number of colour bins. The state distribution of the gradient orientation histogram is formulated in the same way. We use the Bhattacharyya coefficient $\rho[\mathbf{q}(\mathbf{x}_{t-1}), \mathbf{q}(\mathbf{x}_t)] = \sum_{i=1}^{N_c} \sqrt{q_i(\mathbf{x}_{t-1}) q_i(\mathbf{x}_t)}$ for measuring the similarity of two distributions. As a

result, the distance measure is equal to $d = \sqrt{1 - \rho[\mathbf{q}(\mathbf{x}_{t-1}), \mathbf{q}(\mathbf{x}_t)]}$. In the proposed method, likelihoods of both colour and gradient orientation histograms are estimated using the Bhattacharyya coefficient and an exponential distribution, resulting in $p(\mathbf{z}_t^{col} \mid \mathbf{x}_t^{(i)}) = e^{-\lambda d_{col}}$ being the colour likelihood and $p(\mathbf{z}_t^{or} \mid \mathbf{x}_t^{(i)}) = e^{-\lambda d_{or}}$ being the gradient orientation likelihood. The final importance weight is consequently given by:

$$\mathbf{w}_t^{(i)} = p(\mathbf{z}_t^{col} \mid \mathbf{x}_t^{(i)}) p(\mathbf{z}_t^{or} \mid \mathbf{x}_t^{(i)}) = e^{-\lambda d_{col}} e^{-\lambda d_{or}} \tag{4}$$

where λ is a scaling factor. While d_{col} and d_{or} are the distances of the colour and orientation histogram respectively.

2.2 Transition Model

The transition model of the particle filter has the same importance as the observation model for achieving an accurate forward inference. The variance and/or non-linearity of the motion of different objects do not allow to use a simplified motion model, like the constant velocity in [1]. In our work, the transition model of the particle filter is based on a learnt second order autoregressive model. The Burg method [24] is used for deriving two second order autoregressive functions, independently for the x and y direction. The last term of the object's state, the shape, is represented by a constant term in state space, which is estimated from the segmentation.

2.3 Segmentation of the Particle Samples

The particle filter algorithm treats the uncertainty of the object's state by estimating the state's distribution. In the state model we introduce the shape term S for discriminating the foreground object from the background information during sampling. The shape term is assumed to be known while a segmentation algorithm is employed for estimating it. Finally, the sample's observation is free of background during the likelihood $p(\mathbf{z}_t \mid \mathbf{x}_t^{(i)})$ estimation.

In the current work, the choice of the segmentation algorithm is important. We require that the segmentation algorithm is fast, generic and provides two-class segmentation output. Therefore, we chose the *GrabCut* algorithm, a graph-cut segmentation approach [25]. The algorithm is incorporated with the particle filter for refining the shape of the particle samples.

The area to be segmented is always slightly larger than the area of the sample's shape. Based on the current shape, an initial bounding box is specified where everything outside of it is considered as background and the interior area is considered uncertain. With such input, *GrabCut* segments the foreground object inside the rectangular area occupied by the particle.

The computational cost of the *GrabCut* algorithm scales with the size of the area which has to be segmented. Even though the speed of the GrabCut is appropriate for small regions of interests like our particle samples, the overall computational complexity grows with the number of particle samples. For that reason we have implemented two different sampling strategies.

2.4 Sampling Strategies

To investigate the approximation of the state distribution, we propose two sampling strategies based on the segmentation output. In the first strategy each particle sample is segmented in every iteration in order to refine its shape. We name this sampling strategy a multiple particle filter samples segmentation (*Multi-PaFiSS*) strategy and use this name in our experimental evaluation. The second sampling strategy that we call the single particle filter samples segmentation (*Single-PaFiSS*) strategy is based on segmentation of the sample with the highest importance weight and then propagating its shape to the rest particle samples.

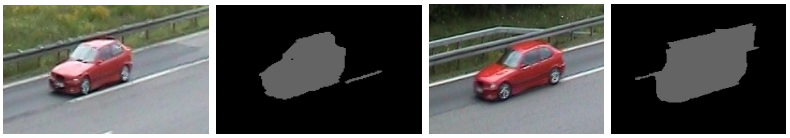
The first sampling strategy is more robust and better adapts to the object's large deformations and scale from frame to frame. However, it comes at the price of increased computational complexity. On the contrary, the second strategy is not that robust to large appearance and scale changes, but it is extremely fast and in many situations also performs well as our experimental validation indicates.

2.5 Segmentation Artifacts and Failure

The proposed algorithm is dependent on the segmentation output for refining the shape of the particle samples. Subsequently, a segmentation failure could obstruct the algorithm's pipeline. We identify two possible failure modes. In one case, the segmentation delivers more than one segmented areas of the same class (Fig. 2b). In the second case, the segmentation explodes by including almost the whole area to a single class or segments everything as background. These two common problems can occur when the *GrabCut* algorithm is used.

The first failure mode provides a successful segmentation output. However, there are some small isolated areas, which we call artifacts, that are often present in the output (Fig. 2b). In our experiments, it never happened to have artifacts with an area larger than 5% of the segmented area. By applying a two-pass connected component labelling, we locate the shape with the largest area and exclude the smaller artifacts.

The second failure mode is more critical because we cannot recover a meaningful segmentation (Fig. 2d). The reason for the failure of the *GrabCut* algorithm is poor quality of the image, failure of the edge extraction and when the colour of the object is not discriminative enough from the background color. Hopefully, this type of failure is easily identified in our algorithm by comparing the current output with the segmentation of the particle sample in the previous time instant based on a threshold. The overlap of the two areas is being compared. In the case of a segmentation failure, the shape of the particle samples becomes rectangular until a new shape is estimated. Thereby, the algorithm continues the tracking task without segmentation refining.



(a) (b) (c) (d)

Fig. 2: Segmentation artifacts and failure: The figures (a) and (c) show input images. (b) The red car is correctly segmented, but there are two connected components. One is a car and the other is a line marking that is an artifact. We eliminate it by keeping the largest connected component. (d) The segmentation algorithm failed to segment (c) and labeled background as foreground object. In this case the shape of the particle samples becomes rectangular until a new shape is estimated

3 Experiments

In order to demonstrate the advantages of the proposed algorithm, we evaluate it on standard tracking sequences used in other related work and we also offer five new challenging sequences[1]. For evaluating our algorithm, we have implemented two versions of our method according to the sampling strategy. The evaluation dataset includes videos with objects of different classes that undergo deformations, occlusions, scale and appearance changes. The test video sequences come from the following datasets: *ETH Walking Pedestrians (EWAP)* [26], *Pedestrian dataset* [27], *Comets project* [28] and the *Aerial Action Dataset* [29]. In total, we used 13 sequences for evaluation. The comparison is done with the standard particle filter and three recent approaches. We compare the two versions of our method with the *TLD* [19], *MIL* [18] and *HoughTrack* [7] algorithms.

The evaluation dataset includes the ground-truth annotations in which the target object is outlined by a bounding box in every frame. We use this type of annotation for all test sequences. This type of annotation is not the appropriate way to describe complex objects (e.g. articulated), but it is the standard annotation method. Therefore, our ground-truth are bounding box representations centered on the centre of mass of the segmented area. *HoughTrack* [7] segmentation based tracking algorithm produces bounding boxes for evaluation in the same way. *TLD* [19] and *MIL* [18] have already a bounding box output and they do not require any modification. Then the overlap between the tracker's bounding box and the annotated one is calculated, based on the *PASCAL VOC* challenge [30] overlap criterion. In all experiments, we set the overlap threshold to 50%. Additionally, we evaluate the computational cost of each method by estimating the average number of tracked frames per second (fps) for every sequence.

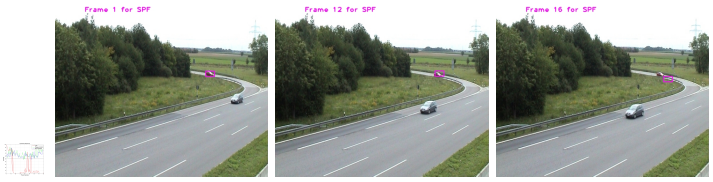
[1] The evaluation dataset can be found at http://campar.in.tum.de/Chair/PaFiSS

3.1 System Setup

Both versions of our method have fixed parameters for all sequences. There are two parameters which affect the performance of the system: the number of particle samples and the threshold indicating the segmentation failure. Since we do not depend on the bounding box, we found out experimentally that the performance of our method does not increase with the number of the samples. Hence, the number of samples is set to 50 and the segmentation failure threshold to the 40% overlap between two successful consecutive segmentation. All methods have been downloaded from the web and executed with their default settings. All experiments are carried out on a standard Intel i7 3.20 GHz desktop machine.

3.2 Comparison to the Standard Particle Filter

The proposed method is compared to the standard particle filter (*SPF*) to prove the superiority of the non-rectangular sampling. For comparison, we implemented the standard bootstrap particle filter [10]. We tested it on all of our sequences but choose the *Entrance* sequence for comparison, since it nicely demonstrates that the amount of background, captured with the bounding box, causes drift. Based on the 50% overlap criterion of the *PASCAL VOC* challenge [30], the standard way of sampling totally fails (Fig 3). Since we also noticed that the increase of the number of samples does not increase the performance of *SPF*, we also set it to 50. In contrast, the proposed method excludes the background information from the likelihood estimation and keeps tracking the object until the end of the sequence.



(a) (b) (c) (d)

Fig. 3: Failure of the Standard Particle Filter (*SPF*). (a): The overlap over time plot, based on the *PASCAL VOC* challenge [30] criterion, shows the performance of the *SPF* and the two versions of our method. Other images: *SPF* tracker gradually drifts due to collecting background information

3.3 Comparison to the state-of-the-art

The comparison to the latest online learning methods aims to show the outstanding performance of the computationally inexpensive single sampling *Single-PaFiSS* strategy and the more accurate multiple segmentation *Multi-PaFiSS*

strategy of our method. Table 1 shows that both strategies of our method outperform the other approaches. While Table 2 shows that Single-PaFiSS is considerably faster than the other approaches.

We introduce the sequences *Entrance*, *Exit 1*, *Exit 2* and *Bridge* for evaluation of occlusions, scale and appearance changes. All of them come from outdoor and dynamic environments where the illumination varies. Furthermore, the main characteristic of the sequences is the simultaneous motion and deformations of the target objects.

There is a number of sequences where we have achieved better results than the other methods. For instance, in *Actions 2* sequence both of our sampling versions outperform the other methods because of the adaption to the scale changes.

In *Exit 1* and *Exit 2* sequences, both versions of our method and *HoughTrack* give similar results, while *TLD* partially drifts. *MIL* succeeds in *Exit 2* but it does not scale in *Exit 1* sequence. Next, in the *Skiing* sequence the abrupt motion leads *TLD* and *MIL* to complete failure while only *HoughTrack* tracks partially the object until the end. In our algorithm, the segmentation fails to refine the object's shape after some time and the algorithm completely drifts.

In general, we face the segmentation failure problem when the quality of the image data is low, like in the *Pedestrian 1* sequence. As long as the tracker is dependent on the segmentation output for getting the object's shape, a possible failure can cause drift. However, our algorithm continues tracking the object by fitting a bounding box to the most recent object shape and sampling using the bounding box, up to small scale changes. This behavior can be observed in the *Single-PaFiSS* sampling strategy while in *Multi-PaFiSS*, it rarely occurs.

Another segmentation failure can be observed in *Cliff-dive 1* sequence where there is an articulated object in low qualitative image data. Consequently there is high probability that the segmentation can provide incorrect information about the shape of the object. For that reason *Single-PaFiSS* performs better than *Multi-PaFiSS* where there are multiple segmentations per frame. In *Bridge* sequence, our algorithm failed to track the object because there is full occlusion. It is a situation which we do not treat with the current framework. The same failure result occurred with the other approaches.

Taking into consideration the evaluation results, one can conclude that the idea of using a probabilist searching method with the combination of shape based sampling produces a robust tracker. The two evaluated implementations of our method give similar results but *Single-PaFiSS* is considerably faster than all the other methods. Fig. 1 and 4 show some of our results for selected frames.

4 Conclusion

We have presented a simple yet effective method for tracking deformable generic objects that undergo a wide range of transformations. The proposed method relies on tracking using a non-rectangular object description. This is achieved by integrating a segmentation step into a bootstrap particle filter for sampling

Table 1: Results for 13 sequences: Percentage of correct tracked frames based on the overlap criterion ($> 50\%$) of the *PASCAL VOC* challenge [30]. The average perentage follows in the end

Sequence	Frames	Single-PaFiSS	Multi-PaFiSS	TLD [19]	MIL [18]	HT [7]
Actions 2 [29]	2113	82.30	**89.87**	8.18	8.42	8.61
Entrance	196	96.42	**98.46**	35.20	35.20	64.79
Exit 1	186	**100**	**100**	74.19	17.74	**100**
Exit 2	172	96.51	98.83	59.88	95.93	**100**
Skiing [7]	81	13.50	**48.14**	6.17	8.64	46.91
UAV [28]	716	64.26	**88.68**	47.90	58.10	73.46
Bridge	55	10.90	10.90	10.9	**12.72**	12.65
Pedestrian 1 [27]	379	1.84	11.60	**66.22**	56.20	12.40
Pedestrian 2 [26]	352	83.23	94.73	**98.57**	89.20	96.30
Cliff-dive 1 [7]	76	**100**	94.73	55.26	63.15	56.57
Mountain-bike [7]	228	18.85	40.35	36.84	**82.89**	39.03
Motocross 2 [7]	23	**95.65**	69.56	73.91	60.86	91.65
Head	231	82.68	**84.41**	77.05	33.34	61.47
Average		65.53	**70.92**	49.88	47.87	58.73

Table 2: Speed results for 13 sequences: Average frames per second (fps) for every sequence. The total average fps follows in the end

Sequence	Frames	Single-PaFiSS	Multi-PaFiSS	TLD [19]	MIL [18]	HT [7]
Actions 2 [29]	2113	6.07	0.50	3.76	**19.09**	1.35
Entrance	196	**51.17**	5.79	5.44	20.60	1.75
Exit 1	186	**39.73**	4.17	5.29	21.10	1.83
Exit 2	172	**21.07**	1.92	4.57	17.79	1.57
Skiing [7]	81	**83.67**	4.71	4.25	24.65	2.93
UAV [28]	716	**36.50**	4.30	6.50	27.3	4.58
Bridge	55	**22.17**	1.46	4.38	19.4	1.67
Pedestrian 1 [27]	379	18.82	2.51	5.87	**24.43**	1.56
Pedestrian 2 [26]	352	**29.46**	3.14	2.73	18.72	1.73
Cliff-dive 1 [7]	76	6.46	0.55	8.97	**30.24**	2.48
Mountain-bike [7]	228	**37.79**	3.22	4.53	26.53	2.81
Motocross 2 [7]	23	10.05	1.45	3.95	**23.28**	1.78
Head	231	7.50	0.76	9.74	**34.40**	7.51
Average		**28.23**	2.65	5.38	23.64	2.20

based on shapes. We investigated two sampling strategies which allow a great trade-off between performance and speed. In the first version, we have reached a better performance by segmenting every particle sample while in the second, we have a less accurate but significantly faster algorithm. During the evaluation on a wide variety of different sequences, our method outperforms recent state-of-the-art object tracking approaches on most sequences or performs at least on

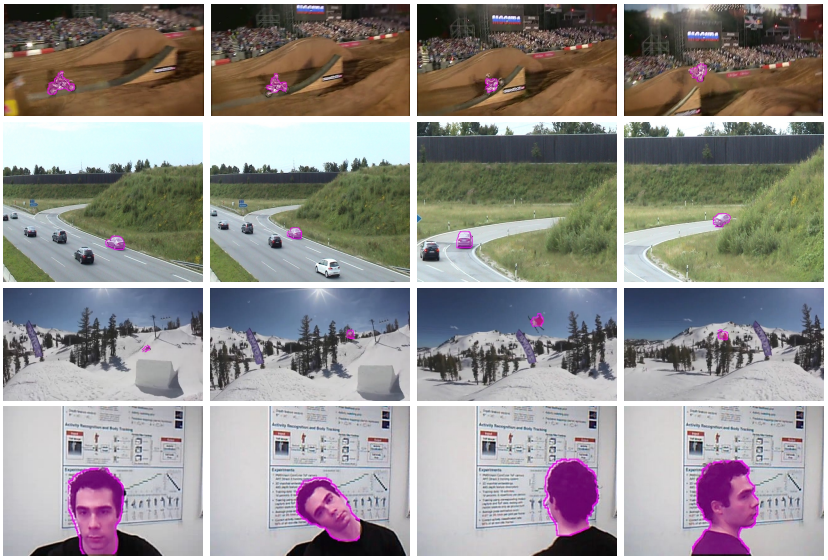


Fig. 4: Additional tracking results (first row: *Motocross 2*, second row: *Exit 2*, third row: *Skiing*, fourth row: *Head*). The *Exit 2* and *Head* sequences have been captured with a stationary camera while in the other two sequences both the object and camera are moving

par. In future work, we will increase the robustness of the segmentation (e.g. by using spatio-temporal information) and speed by parallelizing our method.

References

1. Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Online multiperson tracking-by-detection from a single, uncalibrated camera. IEEE Trans on PAMI (2011)
2. Ikizler, N., Forsyth, D.: Searching video for complex activities with finite state models. In: CVPR. (2007)
3. Wagner, D., Langlotz, T., Schmalstieg, D.: Robust and unobtrusive marker tracking on mobile phones. In: ISMAR. (2008)
4. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. ECCV (2002)
5. Lu, W., Okuma, K., Little, J.: Tracking and recognizing actions of multiple hockey players using the boosted particle filter. Image and Vision Computing (2009)
6. Avidan, S.: Ensemble tracking. In: CVPR. (2005)
7. Godec, M., Roth, P., Bischof, H.: Hough-based tracking of non-rigid objects. In: ICCV. (2011)
8. Isard, M., Blake, A.: Condensation-conditional density propagation for visual tracking. IJCV (1998)

9. Nummiaro, K., Koller-Meier, E., Van Gool, L.: An adaptive color-based particle filter. Image and Vision Computing (2003)
10. Doucet, A., De Freitas, N., Gordon, N.: Sequential Monte Carlo methods in practice. Springer Verlag (2001)
11. Okuma, K., Taleghani, A., Freitas, N., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. ECCV (2004)
12. Bibby, C., Reid, I.: Robust real-time visual tracking using pixel-wise posteriors. ECCV (2008)
13. Chockalingam, P., Pradeep, N., Birchfield, S.: Adaptive fragments-based tracking of non-rigid objects using level sets. In: ICCV. (2009)
14. Tsai, D., Flagg, M., Rehg, J.: Motion coherent tracking with multi-label mrf optimization. Algorithms (2010)
15. Shahed Nejhum, S., Ho, J., Yang, M.: Visual tracking with histograms and articulating blocks. In: CVPR. (2008)
16. Javed, O., Ali, S., Shah, M.: Online detection and classification of moving objects using progressively improving detectors. In: CVPR. (2005)
17. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. ECCV (2008)
18. Babenko, B., Yang, M., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR. (2009)
19. Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: CVPR. (2010)
20. Lucas, B., Kanade, T.: with an application to stereo vision. Proceedings DARPA Image Understanding Workrhop (1998)
21. Kwon, J., Lee, K.: Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In: CVPR. (2009)
22. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: CVPR. (2009)
23. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)
24. Stoica, P., Moses, R.: Introduction to spectral analysis. Volume 51. Prentice Hall Upper Saddle River, NJ (1997)
25. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: ACM Transactions on Graphics (TOG). (2004)
26. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: ICCV. (2009)
27. Leibe, B., Schindler, K., Van Gool, L.: Coupled detection and trajectory estimation for multi-object tracking. In: ICCV. (2007)
28. Ollero, A., Lacroix, S., Merino, L., Gancet, J., Wiklund, J., Remuss, V., Perez, I., Gutierrez, L., Viegas, D., Benitez, M., et al.: Multiple eyes in the skies: architecture and perception issues in the comets unmanned air vehicles project. Robotics & Automation Magazine, IEEE (2005)
29. Lockheed-Martin: Ucf lockheed-martin uav dataset. `http://vision.eecs.ucf.edu/aerial/index.html` (2009)
30. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)