Revisiting Robust Visual Tracking Using Pixel-Wise Posteriors

Falk Schubert[1], Daniele Casaburo[2], Dirk Dickmanns[1]
and Vasileios Belagiannis[3]

[1] Airbus Group Innovations, Germany
falk.schubert@airbus.com, dirk.dickmanns@airbus.com
[2] Technische Universität München, Germany
casaburo@in.tum.de
[3] Chair of Computer Aided Medical Procedures, Technische Universität München, Germany
belagian@in.tum.de

Abstract. In this paper we present an in-depth evaluation of a recently published tracking algorithm [6] which intelligently couples rigid-registration and color-based segmentation using level-sets. The original method did not arouse the deserved interest in the community, most likely due to challenges in reimplementation and the lack of a quantitative evaluation. Therefore, we reimplemented this baseline approach, evaluated it on state-of-the-art datasets (VOT and OOT) and compared it to alternative segmentation-based tracking algorithms. We believe this is a valuable contribution as such a comparison is missing in the literature. The impressive results help promoting segmentation-based tracking algorithms, which are currently under-represented in the visual tracking benchmarks. Furthermore, we present various extensions to the color model, which improve the performance in challenging situations such as confusions between fore- and background. Last, but not least, we discuss implementation details to speed up the computation by using only a sparse set of pixels for the propagation of the contour, which results in tracking speed of up to 200Hz for typical object sizes using a single core of a standard 2.3 GHz CPU.

Keywords: Segmentation, Visual Tracking, Registration, Benchmark, Real-time Tracking

1 Introduction

Visual tracking is a very active research field in which many different, competing methods have been proposed [20]. As many of them focus on similar applications, i.e. generic tracking of manually selected patches, it is important to understand the properties of the type of tracking category each method falls into. Only then it is possible to select the right algorithm from the obscure and overloaded pool of different approaches for a specific task. For instance if one needs to follow the face of a presenter in a lecture hall using a camera, a fast face detector running

as a tracking-by-detection approach might be a better choice than tailoring a generic patch tracking algorithm to that task. In that sense, tracking benchmarks have become increasingly popular to shed some quantitative light into the large pool of many methods. However, often the best performing methods rank almost equally well or their ranking differs depending on the benchmark. The best performing methods on the well-known "Online Object Tracking Benchmark" [28] were learning-based tracking-by-detection methods such as STRUCK [14] and TLD [18]. However, in the same year the "The Visual Object Tracking Challenge" [19] ranked their performance in middle of all methods compared, which was confirmed by the experiments run in the follow-up challenge 2014 [20]. As the selection of algorithms, which were compared and the pool of video sequences used for evaluation had a significant overlap, the conclusions that can be drawn from these results on the general tracking performance are not so obvious for an end-user. After all, these benchmarks only demonstrate that a computer vision algorithm solves a dataset, but not necessarily a target application.

One class of methods which has been under-represented in these state-of-the-art benchmarks are segmentation-based tracking approaches. In our view this is due to four reasons. First, many top-performing methods run usually not in real-time [8, 12]. Second, the overlap computation using bounding boxes penalizes segmentation-based methods as these extract tight object contours, which are usually smaller than the coarse, surrounding groundtruth rectangle. Third, the appearance model is often based on color, which is considered a limitation as apposed to gradient-based descriptors, because color models tend to perform poorly on grayscale and low-contrast sequences. Fourth, segmentation automatically locks onto object boundaries, which prohibits tracking of patches that are not bound by a clear contour, e.g. upper part of a human face. In this paper we want to promote segmentation-based tracking algorithms for generic visual tracking. We demonstrate that the four reasons mentioned above do not always hold true.

On the contrary to the prejudices to segmentation-based tracking, there are quite a few advantages. Highly deformable objects are often impossible to describe with a bounding box. Hence, background information is also included in the foreground area, which misleads learning-based approaches during the online adaption step and can be a major source of drift. A segmentation step, which extracts the contour, could be very valuable to guide the sampling of positive and negative examples for a co-running tracking-by-detection method. Additionally, the contour itself is also a valuable descriptor, which can be used for downstream applications such as pose estimation. For very small objects or those with homogeneous appearance color might be the only valuable descriptor as opposed to the very popular gradient-based ones.

Recently a segmentation-based tracking algorithm was presented [6] which achieves impressive performance in terms of accuracy and run-time. The core power of this approach is the intelligent coupling of rigid registration and fast segmentation via level-sets. This combination allows very fast computation and yet accurate contour estimation. Both together allow the color-based model-

ing to achieve best performance even in very low-contrast scenarios, which are considered intractable for segmentation-based tracking. Due to the challenges in reimplementation and lack of a quantitative evaluation on state-of-the-art datasets, the method did not receive the attention it deserves. Therefore, we present in this paper such a quantitative evaluation using the latest datasets in the tracking community and compare it to the best methods. Furthermore, we extend the method in terms of color and shape model in order to overcome major limitations such as the confusion between fore- and background. Last, but not least, we provide details about an implementation trick using only a subset of the pixels around the contour, which results in tracking speeds of up to 200Hz for typical object sizes using a single core of a standard 2.3 GHz CPU.

The paper is structured as follows. First, we provide an overview of relevant tracking methods and discuss our choice of the baseline segmentation-based tracking. In section 3, we discuss the details of the baseline method. In section 4, we present our extensions to the baseline method. In section 5, we compare all methods against other state-of-the-art approaches using three common benchmark datasets. Last, but not least, we conclude our paper and give an outlook to future research.

2 Related Work

There is vast amount of literature on visual object tracking. For a detailed review we refer to [21, 29]. In this section, we present related work to our method and we mainly focus on approaches that rely on segmentation and non-bounding object representations.

Learning-based object trackers have dominated the field of non-rigid object tracking. The first works on this direction have been published by Avidan [2] and Javed et al. [16], where tracking is defined as a classification problem. In [13] a semi-supervised algorithm was proposed, which learns from unlabeled data during the execution. Another popular learning-based tracker (TLD) was published in [18], which combines online updated random forest and a KLT tracker [4]. Finally, the margin-based trackers[7, 14] have relied on a structured SVM and online updates for tracking, often at the cost of speed. However, all the above approaches are being trained with samples that come from a bounding-box. Consequently, background information can be included into the training process with the threat to drift the tracker. Also the online training inherently limits the run-time speed.

The idea of tracking-by-segmentation has been investigated extensively in the past [24]. In [5], a particle filter has been combined with a graph-based segmentation for sampling observations only from the foreground area. Segmentation-based approaches have also been combined with a Bayesian framework [1], Random walkers [23] and variational approaches [26]. In [8], the object is defined by fragments based on level-sets. Recently, Tsai et al. [25] have proposed an energy minimization scheme within a multi-label Markov Random Field model, but the method does not work online. A new approach based on level-sets was published

in [6]. As opposed to the methods discussed before, the key insight of the authors is, that the contours of deformable objects have a significant rigid part, which can be propagated more efficiently via a registration rather than re-segmentation. The impressive results demonstrated accurate and very fast tracking. However, due to missing quantitative evaluation, the method never raised enough attention in the community to promote segmentation-based tracking. In this paper we adopted this approach and provide such as an evaluation as well as several extensions to it.

To leverage the power of both worlds, learning-based and segmentation-based tracking, approaches have been developed which combine them. In [12] a method called HoughTrack was proposed, merging random forests, hough voting and graph-cut segmentation. The classification output initializes the segmentation serving as a better refinement of the target object. The tracker is relatively slow, but it has produced promising results. PixelTrack [10] also relies on hough voting to detect the object, followed by segmentation between foreground and background. Both are few exceptions of approaches using segmentation that were considered in the two VOT-challenges [19, 20]. In our evaluation we compare both methods against [6] and our extended version.

3 Segmentation-based Tracking

In order to make the paper self-contained and to provide background information for our proposed extensions, we revisit the baseline method [6] which consists of three major steps as illustrated in figure 1. First, the appearance-model is constructed using a probabilistic formulation with a pixel-wise posterior. Second, using this model the level-set segmentation is carried out. Third, the rigid registration and contour propagation (i.e. tracking) is performed. In the following we will discuss each of the steps individually. Last, but not least, we present details on how to significantly speed up the computation by considering only a subset of the pixels as motivated in [6].

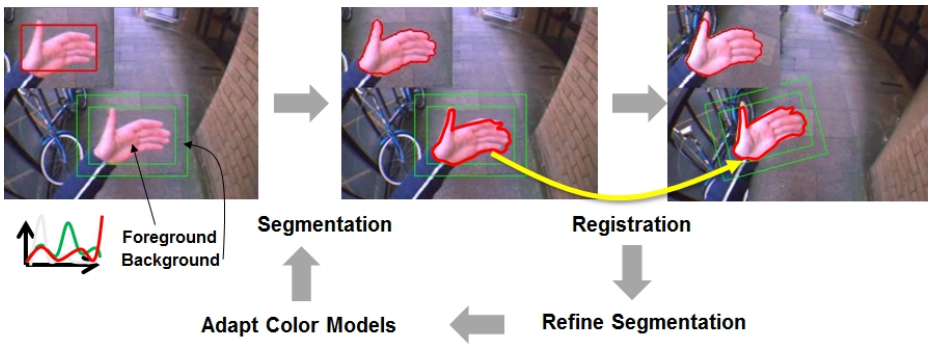

Fig. 1. Overview of the method presented in [6].

Appearance-Model The segmentation and tracking procedures are derived from a probabilistic framework in which the representation of the object is obtained by modeling the information coming from its location (defined by a warp $W(x, \mathbf{p})$ within the image and the shape of its contour C (the zero-level set $C = \{x|\Phi(x) = 0\}$, extracted from the embedding function $\Phi(x)$). The latter enable to define the foreground and background areas $\Omega_i, i = \{f, b\}$ from which the colors $\mathbf{y}$ at the pixels locations $\mathbf{x}$ are used to build the color histograms $M_i, i = \{f, b\}$ which serve as the appearance models $P(\mathbf{y}|M_i), i = \{f, b\}$. Within this framework, it is possible to infer the embedding function $\Phi(x)$ and the position $\mathbf{p}$ of the object by expressing the pixel-wise joint probability distribution:

$$P(\mathbf{x}, \mathbf{y}, \Phi, \mathbf{p}, M) = P(\mathbf{x}|\Phi, \mathbf{p}, M)P(\mathbf{y}|M)P(M)P(\Phi)P(\mathbf{p}) \tag{1}$$

Conditioning on $\mathbf{x}$ and $\mathbf{y}$ and marginalizing over M yields the pixel-wise posterior [6]:

$$P(\Phi, \mathbf{p}|\mathbf{x}, \mathbf{y}) = \frac{1}{P(\mathbf{x})} \sum_{i=\{f,b\}} \big\{P(\mathbf{x}|\Phi, \mathbf{p}, M_i)P(M_i|\mathbf{y})\big\}P(\Phi)P(\mathbf{p}). \tag{2}$$

For each pixel $\mathbf{x}$, the posteriors are merged with a logarithmic opinion pool assuming pixel-wise independence:

$$P(\Phi, \mathbf{p}|\Omega) = \prod_{i=1}^{N} \sum_{j=\{f,b\}} \big\{P(\mathbf{x}_i|\Phi, \mathbf{p}, M_j)P(M_j|\mathbf{y}_i)\big\}P(\Phi)P(\mathbf{p}) \tag{3}$$

In order to remove the need to re-initialize the level set during its evolution, Φ is forced to be as close as possible to a signed distance function by introducing a geometric prior:

$$P(\Phi) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(|\nabla\Phi(\mathbf{x}_i)| - 1)^2}{2\sigma^2}. \tag{4}$$

By taking the log of Eq.3, we obtain the objective function for the pixel-wise log posterior:

$$\log\left(P(\Phi, \mathbf{p}|\Omega)\right) \propto \sum_{i=1}^{N} \left\{\log\left(P\left(\mathbf{x}_i|\Phi, \mathbf{p}, \mathbf{y}_i\right)\right) - \tfrac{(|\nabla\Phi(\mathbf{x}_i)|-1)^2}{2\sigma^2}\right\} + N\log\left(\tfrac{1}{\sigma\sqrt{2\pi}}\right) + \log\left(P(\mathbf{p})\right) \tag{5}$$

Segmentation The segmentation of the object foreground from its background is achieved by iteratively optimizing Eq.5 w.r.t. Φ under the assumption of constant $\mathbf{p}$ using the following derivative of Eq.5:

$$\frac{\partial P(\Phi, \mathbf{p}|\Omega)}{\partial\Phi} = \frac{\delta_\epsilon(\Phi)(P_f - P_b)}{P\left(\mathbf{x}|\Phi, \mathbf{p}, \mathbf{y}\right)} - \frac{1}{\sigma^2}\left[\nabla^2\Phi - \operatorname{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)\right], \tag{6}$$

in which $\delta_\epsilon(\Phi)$ is the derivative of the blurred Heaviside step function H_ϵ and ∇^2 is the Laplacian. The probabilities $P_i, i = \{f, b\}$ indicate how strong a color

value $\mathbf{y}$ belongs to foreground/background and is defined as follows:

$$P_i = \frac{P(\mathbf{y}|M_i)}{\eta_f P(\mathbf{y}|M_f) + \eta_b P(\mathbf{y}|M_b)}, i = \{f, b\} \tag{7}$$

Note, that the use of normalized histrogram look-ups ($P(\mathbf{y}|M_i), i = \{f, b\}$) is crucial. We normalized via dividing the histrograms by the number of fore-/background pixels and using discrete voting for the construction of them.

Tracking Following the inverse compositional approach [4], the position of the object in a sequence is described by the warp transformation $\mathbf{W(x,}\Delta\mathbf{p)}$, where $\Delta\mathbf{p}$ represents the incremental warp and is calculated by optimizing Eq.5 w.r.t. the parameter vector $\mathbf{p}$ while keeping the embedding function Φ constant. The resulting $\Delta\mathbf{p}$ is expressed as:

$$\Delta\mathbf{p} = \left[\sum_{i=1}^{N} \frac{1}{2P(\mathbf{W}(\mathbf{x}_i,\Delta\mathbf{p})|\Phi,\mathbf{p},\mathbf{y}_i)} \left(\frac{P_f}{\sqrt{H_\epsilon(\Phi(\mathbf{x}_i))}} + \frac{P_b}{\sqrt{(H_\epsilon(\Phi(\mathbf{x}_i)))}}\right) \mathbf{J}^T\mathbf{J}\right]^{-1} \times \sum_{i=1}^{N} \frac{(P_f - P_b)\mathbf{J}^T}{P(\mathbf{W}(\mathbf{x}_i,\Delta\mathbf{p})|\Phi,\mathbf{p},\mathbf{y}_i)}, \tag{8}$$

in which $\mathbf{J} = \frac{\partial H_\epsilon}{\partial \Phi}\frac{\partial \Phi}{\partial \mathbf{x}}\frac{\partial \mathbf{W}}{\partial \Delta\mathbf{p}} = \delta_\epsilon\left(\Phi\left(\mathbf{x}_i\right)\right)\nabla\Phi\left(\mathbf{x}_i\right)\frac{\partial \mathbf{W}}{\partial \Delta\mathbf{p}}$ and $\frac{\partial \mathbf{W}}{\partial \Delta\mathbf{p}}$ is the warp Jacobian.

Speed-Up via Pixel-Subsets In a straight-forward implementation the probability maps are computed dense on the area selected by the user which defines the foreground and some border padding which defined the background area. A drawback of this implementation is that the run-time significantly depends on the size of the user selection. Since the global color-model is not very sensitive to the resolution, the selection can be internally down-scaled to a maximum size (e.g. 80×80 px) which guarantees an upper bound on the run-time.

If we investigate the actually values that are considered for the computation, we see that the derivative of the Heaviside step function δ_ϵ defines a tight area along the contour in which values are greater than zero. Outside this area the values are very small and hence the influence of the other variables in Eq. 6 is suppressed. Therefore, we compute this area as a list of pixel indices and also store a precomputed index list for each neighbor. Then we iterate only over these index lists for all computations instead over the whole image matrix. To reduce the influence of the contour size on the run-time, we keep the width of the area constant at $\beta = 10$px. The cut-off threshold of $\delta_\epsilon(x)$ is controlled by the decay function $c = \frac{2\exp(\beta)}{(1+\exp(2\beta))^2}$. Only indices relating to pixel positions x for which $\delta_\epsilon(x) > c$ are stored in the final index lists. On the "hand sequence" of [6] this increases the overall framerate from 60Hz (using all pixels) to 200Hz on average using a single core of a standard 2.3GHz Intel i7 CPU.

4 Extensions

One of the major failure cases of the baseline method is confusion between fore- and background color which destroys the appearance models during run-time

and often ends in tracking failure. In order to overcome this, we discuss three extensions. First, an additional prior which guides the smoothness of the contour can be added to the geometric one (see Eq.4) as discussed in [22]. This prior penalizes long contours and therefore ones with high curvature. This slows down the expansion of the contour to wrong foreground pixels without generally shrinking it. Hence, in situations where fore- and background are hard to distinguish, the stiffness of the contour will temporarily aid as a guide.

As a second and third extension we propose modifications to the sampling for building and updating the color model. Both extensions aim to avoid sampling confusing colors either by assuming geometric locality of the foreground or by explicitly avoiding colors which also appear in the background. In the following we will discuss each one separately.

Color Locality The fore- and background probability maps P_f and P_b in Eq.6 are computed using two separate color-histograms. In the baseline method each pixel votes into them by equal weight. Hence, any information about spatial content is lost. Whereas this results in a rotation invariant description, it is quite sensitive to wrong pixel assignment to fore- and background in cases of bad contour estimation. Therefore we build the appearance histograms in a similar manner to the locality sensitive histogram algorithm [15]. Each pixel vote is weighted according to its distance from the window center, leading to a histogram expressed as:

$$H^L(b) = \sum_{p=1}^{W} \alpha^{|p-c|} \mathbb{I}\left(\mathbf{I}_p, b\right), \quad b = 1, \cdots, B \tag{9}$$

where W is the number of pixels composing the window. The value B is the total number of bins and $\mathbb{I}\left(\mathbf{I}_p, b\right)$ is an indicator function resulting in zero except when the pixel information $\mathbf{I}_p$ belongs to bin b. The weight $\alpha \in [0,1]$ controls the decay of importance for pixels far away from the window center. The histograms are then normalized using the factor $n_f^L = \sum_{p=1}^{W} \alpha^{|p-c|}$. For the histogram relating to the foreground, the centroid of the bounding box is used for the distance calculation. Conversely, the background weighting scheme is computed considering the bounding box borders as the origins from which to calculate the distances.

Color Frequency As discussed for the previous extension, two separate histograms for modeling the fore- and background are computed. This leads to controversy in scenarios where a color appears in both, background and foreground. A color bin is much more informative and distinctive if it appears only in either one of the histograms. Similarly problems have been discussed in the information retrieval community. Documents are commonly represented by a vector of frequencies of the composing words employing the '*term frequency-inverse document frequency*'(*tf-idf*) weighting technique. [3]. We propose to adopt a similar weighting scheme to emphasize the uniqueness of colors by computing:

$$P(\mathbf{y}|M_f) = \frac{|\mathbf{y}_f|}{\eta_f} \cdot \frac{1}{\max\left(|\mathbf{y}_b|, 1\right)} \tag{10}$$

in which the first term accounts for the frequency of occurrence of the color $\mathbf{y}$ in the foreground while the second term considers how much the same color appears in the background. The same is applied to the background probability $P(\mathbf{y}|M_b)$.

5 Evaluations

In this section we will demonstrate that the baseline method and our proposed extensions rank among the top performers for two well known benchmark datasets. As a qualitative validation of our implementation, we reproduced the results on four sequences of [6][4]. All frames of the sequences were successfully tracked. Visual results for the "hand sequence" are depicted in figure 2.

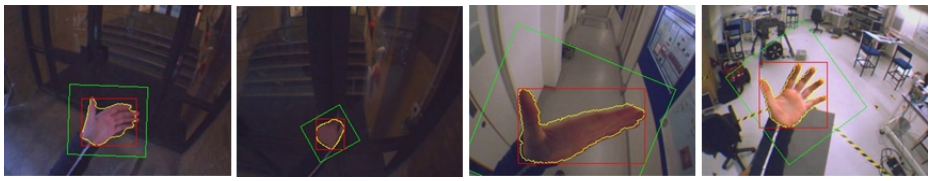

Fig. 2. Four frames with our results from the "hand sequence" presented in [6]. The contour is shown in yellow, the full 6DOF bounding box (including rotation) in green and rectified, tight predicted bounding box used for overlap computations in red.

OOT - Online Object Tracking This benchmark[5] was presented in [28] and compared 29 state-of-art trackers on 50 sequences. For brevity we will compare the *Baseline* (i.e. our implementation of [6]) and our *Proposed Extensions* (using all extensions together) against the 5 top-ranked trackes from the *one-pass evaluation* (OPE) experiment, which were: ASLA [17], CXT [9], SCM [30], STRUCK [14] and TLD [18]. The evaluation scheme measured two values: success and precision rate. The first one measures the rate of overlap between the groundtruth bounding box and the predicted one considering various overlap thresholds. To summarize these values the area under the resulting curve (AUC) is measured. The second rate measures how often the centers of the groundtruth and predicted box are close enough, again considering various distance thresholds. To summarize these values, the precision rate at a distance threshold of 20 pixels is used. Since both the *Baseline* method and the *Proposed Extensions* of this paper intrinsically depend on color information, the evaluation is performed in two versions: one using all sequences and one using only color sequences (35 out of 50).

In table 1 the results for the *Baseline* and the *Proposed Extensions* as well as the 5 best trackers are summarized (left using all 50 sequences, right considering

[4] Sequences were kindly provided by Esther Horbert from the Computer Vision Group, RWTH Aachen University.

[5] http://visual-tracking.net

only color sequences). The segmentation-based method successfully ranks in the top for both success rate and precision rate, especially when considering only color sequences. Our *Proposed Extensions* always show an increased performance w.r.t. the *Baseline*. This demonstrates the power of color-based segmentation in respect to well-known learning-based techniques and motivates the combination of both worlds. Visual results for the sequences "FaceOcc1", "Woman" and "Lemming" are depicted in figure 3.

(a) OOT using all sequences.

OOT Benchmark (all sequences)			
Success AUC		Precision rate Location error at 20 px	
Tracker	Score	Tracker	Score
SCM [30]	**0.5052**	STRUCK[14]	**0.7088**
STRUCK[14]	**0.4777**	SCM [30]	**0.7069**
TLD[18]	**0.4405**	TLD[18]	**0.6774**
ASLA[17]	0.4387	Proposed Extensions	0.6389
CXT[9]	0.4288	CXT[9]	0.6121
Proposed Extensions	0.4217	ASLA[17]	0.6030
Baseline	0.4103	Baseline	0.5988

(b) OOT using only color sequences.

OOT Benchmark (only color sequences)			
Success AUC		Precision rate Location error at 20 px	
Tracker	Score	Tracker	Score
SCM [30]	**0.4517**	Proposed Extensions	**0.6485**
Proposed Extensions	**0.45**	STRUCK[14]	**0.6364**
Baseline	**0.4411**	Baseline	**0.6288**
STRUCK[14]	0.4304	SCM [30]	0.6242
TLD[18]	0.3859	TLD[18]	0.5957
ASLA[17]	0.3698	ASLA[17]	0.5172
CXT[9]	0.3499	CXT[9]	0.5107

Table 1. OOT Benchmarks: left, top-5 as reported in [28] and our proposed methods using all 50 sequences; right, the results when considering only color-sequences. Highest result is marked in red, the second highest is marked in blue and the third highest is marked in green.

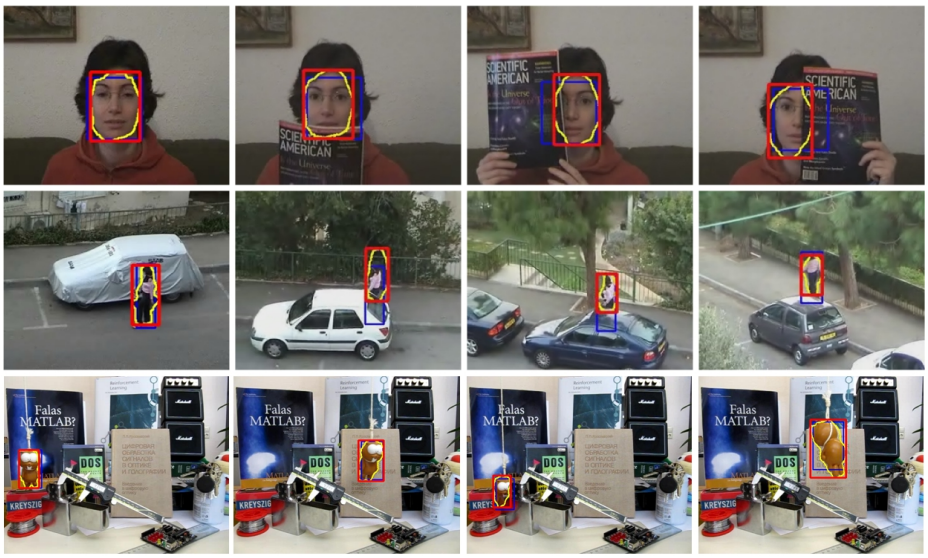

Fig. 3. Three videos from the OOT[28] dataset showing the groundtruth in blue and our method in yellow (for the contour) and red (for the final predicted bounding box).

VOT - Visual Object Tracking The idea of the OOT benchmark was to provide a reference dataset for better comparison of tracking methods. It compared

publicly available trackers (using default parameter settings) on a manually defined collection of videos. The VOT2013 carried this idea further and created an open competition. All participants could individually tune their algorithm to the benchmark dataset for best performance guaranteeing a fair comparison. The 16 benchmark videos were automatically extracted from a pool of about 60 sequences. A whole committee supervised the challenge (i.e. selection of algorithms and dataset) and the results were published in [19]. Due the big success, the challenge will be continued (e.g. VOT2014[20]). The evaluation scheme is similar to the OOT. The performances are captured by two measures: accuracy and robustness. The accuracy value measures how well the bounding box predicted by the tracker overlaps with the groundtruth bounding box. The robustness of a tracker is evaluated by the failure rate which expresses how many times the tracker completely loses the target within a sequence. The results are averaged over 15 runs. Three experiments were carried out: `Baseline`, in which a tracker is tested on all sequences by initializing it on the groundtruth bounding boxes; `Region noise`, like the `Baseline` but initialized with noisy bounding box; `Grayscale`, like the `Baseline` one but on grayscale converted sequences. Similarly to OOT benchmark, we exclude the last experiment as our segmentation-based methods heavily rely on color as a descriptor.

The official results are summarized in the left part of Table 2. The top performing trackers on all the experiments are PLT[6], FoT[27] and EDFT[11] respectively. Interestingly, the top-performing learning-based methods STRUCK[14] and TLD[18] from the OOT benchmark rank only in the middle. The reasons are not clear, other than that the dataset is different, yet the visual challenges in the videos are quite similar. We also report the scores of the state-of-the-art segmentation-based tracker HT[12] which also ranks in the lower part.

In the right part of Table 2, the results for our segmentation-based method (*Baseline* and the *Proposed Extensions*) are reported. The average overlap as well as the mean number of restarts are comparable to the two top-performing methods. As the ranking procedure was only accessible to the authors of the VOT2013 challenge, it was not possible to compute the exact rank our proposed methods would have reached. Nevertheless, in comparing the obtained performances in both parts of Table 2 w.r.t. the rank of the winning trackers, it is reasonable to place both versions of the considered tracker among the first ones. Our proposed extensions enhanced the performance of the baseline version in terms of *robustness*. The drop in accuracy is due to the intrinsic tendency to focus on unique colors, especially that ones close to the object center, leading to a smaller bounding-box. It is exactly that feature that allowed it to be less sensible w.r.t. noisy initializations keeping the track more steady on the target.

Overall, the experiments confirm the reliability of the presented segmentation-based tracking approach, especially in presence of challenging conditions such as changes in lighting condition (e.g. see top row of figure 4), deformations, etc.. The results strengthen the intuition to exploit more information related to the

[6] No official publication available. Only a brief abstract in [19].

color appearance. Visual results for the three sequences "David", "Bolt" and "Bicycle" are depicted in figure 4.

VOT2013 CHALLENGE: Official Results								Evaluated Methods	
Experiment	Metric	#1: PLT	#2: FoT[27]	#3: EDFT[11]	#12: STRUCK[14]	#17: TLD[18]	#20: HT[12]	Baseline	Proposed Extensions
baseline	Accuracy	**0.6185**	**0.6492**	0.6	0.584	0.5962	0.4727	**0.6316**	0.6053
	Robustness	**0**	0.6545	0.4455	1.365	2.9383	1.6558	**0.1875**	**0.125**
region-noise	Accuracy	0.5892	**0.6**	0.5693	0.5275	0.5735	0.4723	**0.6256**	**0.6054**
	Robustness	**0.0236**	0.6921	0.6593	1.4710	2.9839	1.8432	**0.2667**	**0.1375**

Table 2. VOT2013 challenge: results comparison. Highest result is marked in red, the second highest is marked in blue and the third highest is marked in green.

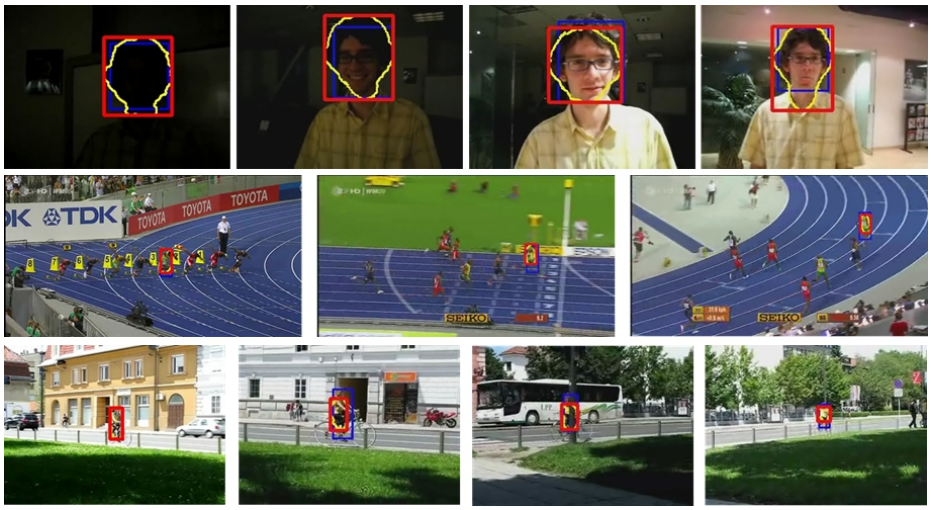

Fig. 4. Three videos from the VOT2013[19] dataset showing the groundtruth in blue and our method in yellow (for the contour) and red (for the final predicted bounding box).

Segmentation-Based Trackers Since segmentation-based trackers are under-represented in the two benchmarks discussed above, we represent a comparison of three state-of-the-art methods using segmentation: PaFiSS [5], PixelTrack [10] and HT[12]. Following [10], we measure the percentage of frames in which the object is correctly tracked. The tracking is considered correct if the overlap measure between the output bounding box and the groundtruth is above 10%.

The evaluation is performed using the dataset presented in [10], which contains objects that undergo rigid and non-rigid deformations, large lighting variations as well as partial occlusions. In table 3 the results are summarized by two average values. The first average value in the last row is the performance considering all sequences. The second value only considers values for which PAFISS results were available. Our proposed approach outperforms all other methods. The extensions perform slightly worse due to the same reasons as reported for the VOT2013 dataset. Visual results for the sequences "Cliff-dive 1" and "Mountain-Bike" are depicted in figure 5.

Sequence	HT[12]	PixelTrack[10]	PaFiSS[5]	Baseline	Proposed Extensions
David	89.25	45.16	–	**99.78**	99.57
Sylvester	55.02	36.80	–	**99.92**	99.82
Girl	92.22	93.21	80.64	**99.80**	**99.80**
Face Occlusion 1	99.44	**100**	98.78	99.77	99.66
Face Occlusion 2	**100**	88.34	71.41	99.88	99.88
Coke	72.88	91.53	–	**99.65**	**99.65**
Tiger 1	26.76	46.48	–	**99.72**	**99.72**
Tiger 2	41.10	98.63	–	**99.72**	**99.72**
Cliff-dive 1	**100**	**100**	**100**	**100**	**100**
Motocross 1	**100**	57.59	23.78	99.39	99.39
Skiing	**100**	**100**	61.63	98.76	97.53
Mountain-bike	**100**	94.55	98.68	99.56	99.56
Cliff-dive 2	**100**	32.79	23.19	**100**	**100**
Volleyball	45.12	**100**	34.6	58.60	58.40
Motocross 2	**100**	**100**	**100**	69.56	69.56
Transformer	38.71	94.35	**100**	**100**	**100**
Diving	21.21	88.74	26.84	**100**	87.01
High Jump	77.87	94.26	9.02	65.57	66.39
Gymnastics	98.87	99.09	19.95	**100**	90.87
Average	77.13 / 84.32	**82.18 / 88.78**	- / 60.61	**94.19 / 92.21**	**92.97 / 90.57**

Table 3. PixelTrack [10] comparison: percentage of correctly tracked frames. Highest result is marked in red, the second highest is marked in blue and the third highest is marked in green. The first average value in the last row is the performance considering all sequences. The second value only considers values for which PAFISS results were available.

6 Conclusions

In this paper we motivated the use of segmentation-based tracking. We adopted a baseline method [6] which intelligently couples rigid registration and fast level-set segmentation and reported impressive results. Our implementation verifies the conclusions of the authors as we could qualitatively reproduce the published results. However, the method did not receive the attention it deserves in the

Fig. 5. Two videos from the PixelTrack dataset [10] showing the groundtruth in blue and our method in yellow (for the contour) and red (for the final predicted bounding box).

latest tracking benchmarks. We believe this is mainly due to the challenges in reimplementation and a missing quantitative evaluation on common datasets. In this paper we provide such an evaluation using the latest benchmark datasets and following the state-of-the-art evaluation protocols. The results show that this method ranks among the best performing trackers, which demonstrates the often underestimated power of segmentation-based tracking. In many applications contours and color-models are much more useful than the typically used gradient-based descriptors, making such a segmentation-based tracking algorithm an interesting alternative. Furthermore, we presented various extensions which aim to mitigate a major drawback of the global color-based appearance model. In future work, we would like to couple this segmentation-based tracker with a tracking-by-detection algorithm to increase the overall tracking performance even further.

Acknowledgments We thank Esther Horbert (Computer Vision Group RWTH Aachen University) for providing four evaluation sequences and valuable feedback for resolving open questions on the hidden details of [6].

References

1. Aeschliman, C., Park, J., Kak, A.: A probabilistic framework for joint segmentation and tracking. In: CVPR (2010)
2. Avidan, S.: Ensemble tracking. PAMI (2007)
3. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval. ACM press New York (1999)
4. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. IJCV (2004)
5. Belagiannis, V., Schubert, F., Navab, N., Ilic, S.: Segmentation based particle filtering for real-time 2d object tracking. In: ECCV (2012)

6. Bibby, C., Reid, I.: Robust real-time visual tracking using pixel-wise posteriors. In: ECCV (2008)
7. Chen, D., Yuan, Z., Hua, G., Wu, Y., Zheng, N.: Description-discrimination collaborative tracking. In: ECCV (2014)
8. Chockalingam, P., Pradeep, N., Birchfield, S.: Adaptive fragments-based tracking of non-rigid objects using level sets. In: ICCV (2009)
9. Dinh, T., Vo, N., Medioni, G.: Context tracker: Exploring supporters and distracters in unconstrained environments. In: CVPR (2011)
10. Duffner, S., Garcia, C.: Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects. In: ICCV (2013)
11. Felsberg, M.: Enhanced distribution field tracking using channel representations. In: ICCVW (2013)
12. Godec, M., Roth, P., Bischof, H.: Hough-based tracking of non-rigid objects. In: ICCV (2011)
13. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. ECCV (2008)
14. Hare, S., Saffari, A., Torr, P.: Struck: Structured output tracking with kernels. In: ICCV (2011)
15. He, S., Yang, Q., Lau, R., Wang, J., Yang, M.H.: Visual tracking via locality sensitive histograms. In: CVPR (2013)
16. Javed, O., Ali, S., Shah, M.: Online detection and classification of moving objects using progressively improving detectors. In: CVPR (2005)
17. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: CVPR (2012)
18. Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: CVPR (2010)
19. Kristan, M.e.a.: The visual object tracking vot2013 challenge results. In: ICCVW (2013)
20. Kristan, M.e.a.: The visual object tracking vot2014 challenge results. In: ECCV (2014)
21. Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., Hengel, A.V.D.: A survey of appearance models in visual object tracking. TIST (2013)
22. Mitzel, D., Horbert, E., Ess, A., Leibe, B.: Multi-person tracking with sparse detection and continuous segmentation. In: ECCV (2010)
23. Papoutsakis, K., Argyros, A.: Integrating tracking with fine object segmentation. Image and Vision Computing (2013)
24. Paragios, N., Deriche, R.: Geodesic active contours and level sets for the detection and tracking of moving objects. PAMI (2000)
25. Tsai, D., Flagg, M., Rehg, J.M.: Motion coherent tracking with multi-label mrf optimization. BMVC (2010)
26. Unger, M., Mauthner, T., Pock, T., Bischof, H.: Tracking as segmentation of spatial-temporal volumes by anisotropic weighted tv. In: Energy Minimization Methods Workshop in CVPR (2009)
27. Vojíř, T., Matas, J.: Robustifying the flock of trackers. In: CVWW (2011)
28. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: CVPR (2013)
29. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. Acm computing surveys (CSUR) (2006)
30. Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: CVPR (2012)