Dynamic Objectness for Adaptive Tracking

Severin Stalder[1], Helmut Grabner[1], and Luc Van Gool[1,2]

[1]Computer Vision Laboratory, ETH Zurich, Switzerland
{sstalder,grabner,vangool}@vision.ee.ethz.ch
[2]ESAT - PSI / IBBT, K.U. Leuven, Belgium
luc.vangool@esat.kuleuven.be

Abstract. A fundamental problem of object tracking is to adapt to unseen views of the object while not getting distracted by other objects. We introduce *Dynamic Objectness* in a discriminative tracking framework to sporadically re-discover the tracked object based on motion. In doing so, drifting is effectively limited since tracking becomes more aware of objects as independently moving entities in the scene. The approach not only follows the object, but also the background to not easily adapt to other distracting objects. Finally, an appearance model of the object is incrementally built for an eventual re-detection after a partial or full occlusion. We evaluated it on several well-known tracking sequences and demonstrate results with superior accuracy, especially in difficult sequences with changing aspect ratios, varying scale, partial occlusion and non-rigid objects.

1 Introduction

Over decades of research, a vast number of tracking algorithms have been developed for increased performance in particular applications. Many assumptions about the object, the scene and the camera movement have been introduced in order to constrain tracking. In this paper, however, we cope with the most general problem of tracking an object with minimal prior knowledge, *i.e.*, "model-free" tracking. This task is particularly challenging as it is hard to introduce effective constraints.

We first analyze related literature to extract four tracking paradigms that are often used to constrain tracking in restricted situations, as depicted in Fig. 1. In fact, we categorize tracking approaches according to their assumptions on the object's appearance, the scene, the object's saliency and the use of a fixed object model.

We then propose a novel tracking approach that originates from this careful analysis of literature to explore all four identified paradigms. Our main contribution is the definition and the use of *Dynamic Objectness* to sporadically re-discover the tracked object if it moves distinctly from its surroundings. Dynamic Objectness is based on motion saliency and embedded in a discriminative tracking framework. With Dynamic Objectness, our algorithm becomes more aware of *objects* as separately moving entities in larger scenes and therefore effectively limits drifting, as demonstrated in our experiments.
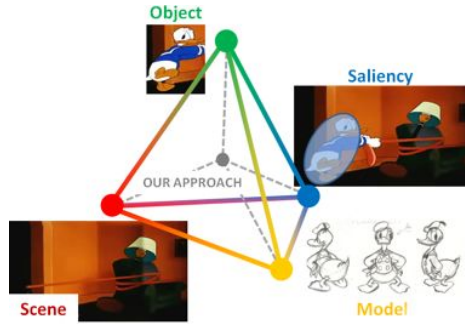
Fig. 1. Four tracking paradigms are combined in our approach. Each indicates where the object moved according to (i) the object itself, (ii) the scene, (iii) the object's saliency and (iv) a previously learned object model. Their combination allows robust tracking in a wide variety of scenarios.

2   Related Work

There is extensive work on tracking, see [1] for a survey, so a detailed review goes far beyond the scope of the paper. However, we give a brief overview to categorize representative tracking approaches according to their dominant tracking paradigms.

Approaches based on a single paradigm. Many classical tracking approaches rely mainly on one specific paradigm motivated by a particular application.

Object: Tracking approaches of this category *adaptively* model the appearance of an object. Such methods have the potential to follow the object through various appearance changes, but at the risk of adapting to other distracting objects or background. Optical flow [2] and tracking of local image features [3] are early representatives. Other approaches keep track of appearance properties of the object region, *e.g.*, a color histogram [4], a subspace model [5], parts [6] or stable structures [7]. More recent approaches combine multiple object features such as color and structure [8,9,10,11] in order to limit drifting.

Scene: Approaches relying on the scene consider objects as *outliers* to the scene model. These approaches are well suited to situations in which the appearance or geometry of the scene is relatively stable, but the object of interest is hard to describe. Their assumptions are particularly successful in static camera settings [12,13,14].

Model: Model based approaches rely either on an off-line training phase or fully on the first frame in order to learn a *fixed* appearance model. During tracking, the model is only applied to the image. Some methods search the fixed object model exhaustively and independently in every frame to not suffer from drifting that might jeopardize all future results. But these methods might fail temporally

because of missed detections (due to an incomplete object model) or false detections in the background on similar objects (due to a too general object model). Other methods with fixed models, however, iteratively perform a local search because of efficiency reasons or ambiguous models. Such approaches might end up in a permanent drifted state even with a fixed model at hand.

Typical representatives are *tracking-by-detection* methods such as [15,16], or classical template tracking [17,18].

Saliency: There are two main cues for saliency: motion and appearance. Motion segmentation approaches such as [19] typically exploit the first, whereas the latter is used for object boundary tracking [20,21].

Approaches based on multiple paradigms. Due to the obvious limitations in more general situations, recent approaches often implicitly combine multiple paradigms.

Object & Scene: Tracking formulated as a binary classification task such as [22,23,24] incorporates regular classifier training in order to distinguish the object from its surrounding background using various feature types. Recent discriminative approaches are also aware of similar objects in the scene as well as the location of the object with respect to background [25]

Object & Appearance Saliency: Such methods are usually named *tracking-by-segmentation.* For example, tracking formulated as repeated figure/ground segmentation [26] incorporates object saliency in single frames, as well as appearance-based foreground models to track salient objects through time.

Object & Model: Adaptive methods are often combined with general models to overcome their limitations. Adaptive methods are particularly susceptible to model drifting since future tracking results rely on previous ones. Conversely, pre-learned general models are drift-free, but at the cost of not being adaptive to varying appearance. Their straightforward combination consists of propagating intermittent detections across frames, for example with optical flow [27]. But there is a whole range of methods that keep some information fixed while adapting a model during tracking [28,29] or methods that introduce additional visual constraints [30]. Furthermore, *meta-level tracker combinations* are used to improve stability. The output of individual trackers can be fused as "black-boxes" [31], probabilistically within a particle filter framework [32] or for specific situations [33].

A note on features and dynamics. Please note that each of the tracking paradigms incorporates a different collection of features. For instance, the object paradigm typically relies on features describing the object which are only meaningful locally for the short-term, *e.g.*, KLT-points, whereas the features of the model paradigm are typically significant in the whole image for a longer period of time, *e.g.*, SIFT features. Dynamics can be exploited in combination with all the tracking paradigms if the object motion is predictable in the target application. For instance, particle filter based approaches are especially well suited to combine the appearance of the objects and its motion [21].

3   Dynamic Objectness for Adaptive Tracking

In this section we introduce our tracking approach based on the four spotted tracking paradigms. The *object*, *scene* and *model* paradigms are employed to precisely follow and to re-detect the object, whereas the *saliency* paradigm serves for adaptation.

Let us assume the target object is initialized in the first frame. Besides a manual initialization, an automatic initialization is possible based on: (i) the *model paradigm* with an already available object model or (ii) the *saliency paradigm* that allows to discover salient objects. After initialization, the image is divided into an object and a scene region and the tracking starts as summarized in the following.

Propagation. Given object and scene regions, both are propagated through time (*object and scene paradigms*).

Detection. The object region is retained in order to re-detect the object after a partial or full occlusion (*model paradigm*).

Exploring and Grouping with Dynamic Objectness. As the object moves in 3D space, it might change its perceived appearance in the image or becomes occluded by another object. Propagation and detection are not sufficient as it becomes necessary to delineate the new appearance from background and other distracting objects, i.e., to adapt to the changed appearance of object in an unsupervised manner. For that purpose, we introduce the *Dynamic Objectness* based on the *saliency paradigm*.
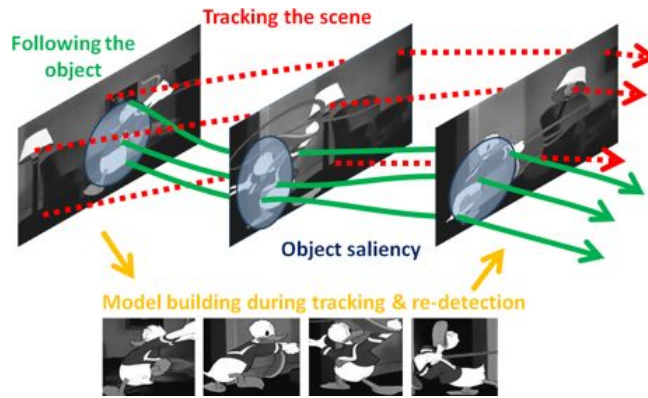
Fig. 2. We are not only following the object (green), but the whole scene (red) in order to not confuse the object with known background. Additional saliency measures (blue) help to further stabilize tracking. During the tracking process, a model (orange) of the target object is carefully built to assist tracking and for eventual re-initialization.

Algorithm 1: Tracking with Dynamic Objectness

---

1 init. set of target object points $\mathcal{O}$, scene points $\mathcal{S}$, and unknown points $\mathcal{U}$

2 **while** *run* **do**

3 $\quad$ PROPAGATION: object and scene paradigm

4 $\quad$ - track all local image features in the image, *i.e.*

5 $\quad$ $\quad$ from $\mathcal{O}$, $\mathcal{S}$, and $\mathcal{U}$

6 $\quad$ DETECTION: model paradigm

7 $\quad$ - detect local image features and add them to $\mathcal{U}$

8 $\quad$ - find matches of $\mathcal{U}$ with target model M_{target}

9 $\quad$ $\quad$ and add to $\mathcal{O}$

10 $\quad$ EXPLORING AND GROUPING: saliency paradigm

11 $\quad$ - get possible object regions using *Dynamic*

12 $\quad$ $\quad$ *Objectness* (Eq. 1)

13 $\quad$ - score these regions using $\mathcal{O}$ and $\mathcal{S}$ (Eq. 2)

14 $\quad$ - update point sets $\mathcal{O}$, $\mathcal{S}$ and $\mathcal{U}$ (Eq. 3)

15 $\quad$ - update object model M_{target}

16 $\quad$ TRACKING RESULT: region covered by $\mathcal{O}$

17 **end**

The tracking process is depicted in Fig. 2 and summarized in Alg. 1. Whereas the paradigms can be implemented using different types of features, we show an implementation based on local image features. In fact, we maintain sets of tracked objects points $\mathcal{O}$, tracked scene points $\mathcal{S}$ and unknown points $\mathcal{U}$ and incrementally learn a model of the target object M_{target} during tracking. The following subsections describe the employed paradigms and their implementations in more detail.

3.1 Propagation

Object paradigms: A straightforward approach to track an object is to simply look for a similar object, or part of it, in the next frame. If the appearance between two sequential frames is similar, then this approach is expected to produce accurate results.

Scene paradigm: In many scenarios, the background is more stable and is easier to track than the object itself. In such cases, the scene paradigm greatly constrains the possible locations of the object.

Implementation. For tracking object features as well as background features we simply use sparse optical flow. KLT-points [3,18] are tracked in the whole scene. Newly appearing points are put in the set of unknown points $\mathcal{U}$. All points are tracked and kept in their respective sets, *i.e.*, propagated. Such an approach deals well with perturbations like deformations at the object level or scale changes. Its accuracy, however, depends on texture and size of the object.

3.2   Detection

Model paradigm: After successfully tracking an object for an extended period of time, the tracker is expected to become more stable. An object model M_{target} is learned for this purpose. The model continuously specializes to the object being tracked. It can then be used to re-detect the object after a partial or full occlusion. Ideally, such a model finally incorporates all appearances of the object and can robustly be applied to the image.

Implementation. All object points $\mathbf{p_i} \in \mathcal{O}$ are characterized by descriptor $\mathbf{d_i}$ extracted from a local image patch around the point. The object model $M_{target} = \{\mathbf{d_i}\}$ consists of these descriptors. The set of all unknown points $\mathcal{U}$ are similarly described to allow for re-detection as object points. In fact, the descriptors from the unknown points are compared to the descriptors of the object model using normalized cross-correlation. If the nearest neighbor from an unknown point descriptor to the object point descriptors is significantly[1] closer than the second nearest neighbor, then the unknown point is considered as matched and added to the set of object points.

3.3   Exploring and Grouping with Dynamic Objectness

Saliency paradigm: The sheer amount of possible appearance changes of an object, *e.g.*, due to lighting or pose variations, can render the object itself hard to follow. Many state-of-the-art tracking algorithms solely model the specific region being tracked and lack a general concept of objects. Conversely, all objects have a number of common properties. In object detection, objects are characterized as standalone "things" with well-defined boundaries and centers, such as cows, cars, and telephones, as opposed to amorphous background "stuff", such as sky, grass and road [34]. In a temporal context, objects are relatively stable in form and tend to move distinctively from their surroundings.

Dynamic Objectness. Recent work in object detection [35,36] exploits the closed boundary characteristic of objects in order to find regions which are likely to fully contain objects. We adopt a similar strategy for discovering unseen appearances of the tracked object. Instead of using static image features such as edges, we characterize objects as entities that tend to move independently from their surroundings and rely on motion saliency. In fact, we first apply motion segmentation on all tracked feature points to get clusters $\mathcal{C}$ of similarly moving points, see Fig. 3(a) and (b).

The object is typically split into several small clusters, whereas the image points in the scene fall in much bigger clusters. In many scenarios the object is moving in foreground with rather diverse 2D motion vectors. In contrast, the scene is at distance and its motion vectors are more homogeneous, even with a

---

[1] Let us consider m_1 and m_2 as the best and second best match respectively, then a match is considered if $m_2/m_1 > \theta_{match}$. We used $\theta_{match} = 0.6$ for all our experiments.

(a) Tracking (b) Motion segmentation (c) Dynamic Objectness (d) Updating tracked object
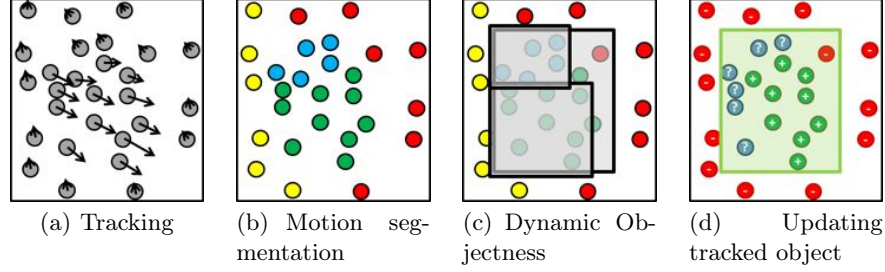
Fig. 3. The tracked feature points shown in (a) are grouped in (b) according to their motion. Our novel Dynamic Objectness allows to locate possible object regions (c). Using the propagated points, the tracked object region can robustly be updated (d).

moving camera. Our Dynamic Objectness is designed to group these foreground clusters together based on their spatial proximity and the compactness of the grouping. It is defined analogue to the "superpixel straddling" score of Alexe *et al.* [35]:

$$DO(R) := 1 - \sum_{C \in \mathcal{C}} \frac{\min(n_{in}(R,C), n_{out}(R,C))}{n(R)}, \tag{1}$$

where $n(R) := \mathrm{card}(\{\mathbf{p}|\mathbf{p} \in R\})$ is the overall number of points in the region R, $n_{in}(R,C) := \mathrm{card}(\{\mathbf{p}|\mathbf{p} \in R, \mathbf{p} \in C\})$ the number of points belonging to cluster C in R and $n_{out}(R,C) := \mathrm{card}(\{\mathbf{p}|\mathbf{p} \notin R, \mathbf{p} \in C\})$ the number of points belonging to cluster C outside R. Similar to [35], clusters entirely inside or outside R contribute 0 to the sum. For a straddling cluster C, the contribution is lower when it is contained either mostly inside R, as part of the object, or mostly outside R, as part of the background, see Fig. 3(c). In doing so the Dynamic Objectness allows to naturally incorporate knowledge about the size of the object and its location during tracking.

Implementation. For all tracked feature points $\mathbf{p_i}$ we build a motion vector $\mathbf{m_i} = [dx, dy]_i$ where dx and dy are the x and y displacements of the tracked point between two successive frames. Based on these motion vectors we apply [37][2] clustering in order to obtain the motion clusters $\mathcal{C}$. For possible regions R we only consider rectangular regions of the approximate object size to find interesting regions.

3.4 Updating Point Sets and Object Model

In case of occlusion or abrupt motion, point tracks tend to be short and need to be updated. This update process is the most critical part of every model-free tracking algorithm since small errors can easily accumulate and lead to drifting.

[2] `http://people.cs.uchicago.edu/~pff/segment/`, 2010/04/21 with parameters: threshold = 0.5, min_size = 1, and $k = 30$-nearest neighborhood.

In our tracking approach, we use Dynamic Objectness that find regions likely to fully contain objects to regularly update the region to be tracked. More precisely, we are interested in re-discovering the target object for updating the object and the scene sets. Therefore we want to select the region proposed by the Dynamic Objectness which most likely corresponds to our target object, see Fig. 3(d). So we define a target score $T(R)$ with respect to the whole scene, $i.e.$, using the set of known object and scene points. Let $n_{obj}(R) := \text{card}(\{\mathbf{p}|\mathbf{p} \in R, \mathbf{p} \in \mathcal{O}\})$ be the number of points belonging to the object set and $n_{sce}(R) := \text{card}(\{\mathbf{p}|\mathbf{p} \in R, \mathbf{p} \in \mathcal{S}\})$ be the number of points belonging to the scene set in region R. The target score

$$T(R) = \frac{n_{obj}(R) - n_{sce}(R)}{n(R)}.$$ (2)

is highest if all points in R are in the object set and none is in the scene or unknown set. For updating we select the most likely region, see Fig. 3(d) as

$$R_{update} = \begin{cases} \underset{R:DO(R)>\theta_{obj}}{\text{argmax}} \quad T(R) & \text{if } T(R) > \theta_{target} \\ \{\} & \text{otherwise} \end{cases}$$ (3)

So the update region must be discovered as an object $(DO(R) > \theta_{obj})$ and furthermore as the object which we are interested to track $(T(R) > \theta_{target})$. Every point in R_{update} is added to the object set $\mathcal{O}$ and every other point is added to the scene set $\mathcal{S}$. Furthermore all the object points are described via a local descriptor and added to the discriminative target model M_{target}. Inspired by [30] we remove every point from the object model that matches to any point not in R_{update}. This ensures that no point is simultaneously in the object model and the scene set.

Implementation.   All regions R with a dynamical objectness measure $DO(R)$ above $\theta_{obj} = 0.85$ are possible object regions and if additionally its target score $T(R)$ is above $\theta_{target} = 0.85$ the region is considered to be part of the target object. These parameters are subject to the well-known stability-plasticity dilemma [38]. The experimental results with fixed parameters provide empirical evidence that their choice is not critical and allows to track objects robustly in many circumstances.

4   Experiments

In this section we first show qualitative results for the Dynamic Objectness that implements the *saliency paradigm*, the backbone of our tracking approach. Then, detailed experiments with quantitative and qualitative results are conducted on competitive tracking sequences. During the experiments, we kept all the parameters fixed as described in the previous sections.

Resources.   For KLT point tracking we use the function *cvCalcOpticalFlow-PyrLK()* of *OpenCV Library 2.0*[3]. In order to increase its robustness we imple-

---

[3] http://opencv.willowgarage.com/, 2011/11/01.

Fig. 4. The *Dynamic Objectness* groups motion clusters (1st and 3rd column; clusters are color coded) into meaningful object regions (2nd and 4th column).

ment a forward-backward verification scheme [30]. For interest point matching we use the *OpenSurf C++* code of Chris Evans[4]. The overall tracking algorithm typically runs with about 5 frames/s, mainly limited by tracking and matching feature points. Dynamic Objectness, the core of our algorithm, typically runs in the order of ms.

4.1   Dynamic Objectness

Fig. 4 depicts results for the proposed Dynamic Objectness. It discovers possible moving objects (2nd and 4th column), given a motion segmentation (1st and 3rd column) and the rough size of the object of interest. The convex hull of all points in the object set define the object region. Please note, that they usually consist of more than one motion cluster.

---

[4] `http://www.chrisevansdev.com/opensurf/`, 2011/11/03
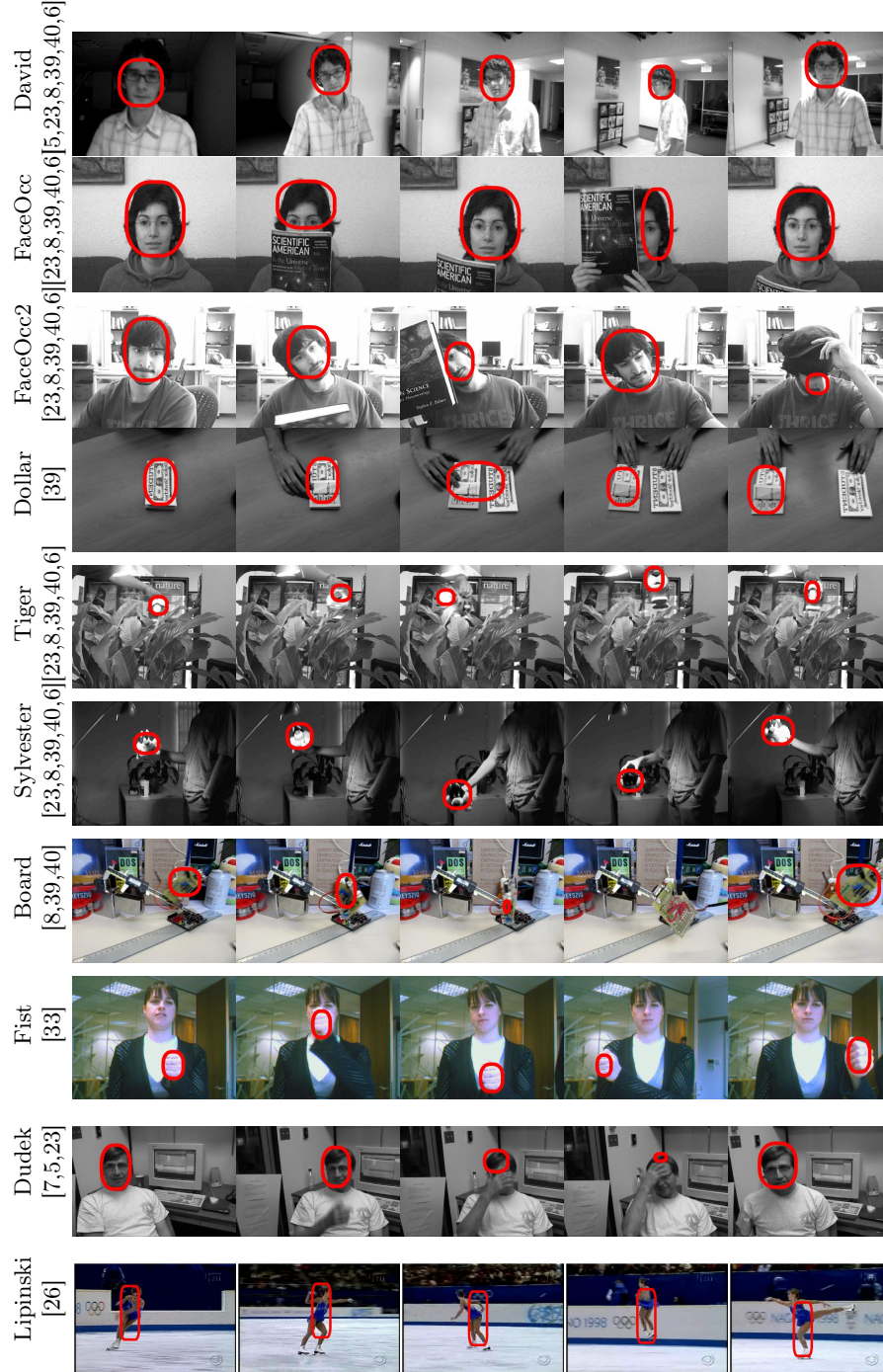
Fig. 5. Our approach is able to precisely distinguish the object from distractors while also being capable of adapting to appearance changes of the target object.

4.2   Quantitative Tracking Results

We compare our method against Multiple Instance Learning (MIL) [39], Superpixel Tracker (SPT) [41][5], Hough Tracker (HTT) [6][6], Tracking-Learning-Detection (TLD) [30][7] and an Ideal Fixed Scale and Aspect Ratio Tracker (IFSART). All trackers are initialized with the same bounding box. As a comparison measure, we adopted the average of the standard PASCAL overlap, *i.e.*,

$$\frac{1}{T} \sum_{t=1}^{T} \frac{A_{gt}(t) \cap A_{tr}(t)}{A_{gt}(t) \cup A_{tr}(t)}, \tag{4}$$

where $A_{gt}(t)$ is the ground truth area and $A_{tr}(t)$ the predicted area by the tracker at time t. Results are shown in Table 1 and discussed below.

Sequence	MIL [39]	IFSART	SPT [41]	HTT [6]	TLD [30]	**ours**
David	0.54	(0.64)	0.12	<u>0.61</u>	0.59	**0.74**
FaceOcc	0.45	(0.55)	0.13	<u>0.56</u>	0.49	**0.58**
FaceOcc2	<u>0.49</u>	(0.49)	0.37	0.39	0.34	**0.57**
Dollar	<u>0.69</u>	(1.00)	0.37	0.66	**0.79**	0.63
Tiger	**0.33**	(0.47)	0.26	0.11	<u>0.29</u>	0.28
Sylvester	0.62	(0.72)	0.65	<u>0.69</u>	**0.71**	<u>0.69</u>
Board	<u>0.32</u>	(0.65)	0.21	0.19	0.16	**0.64**
Fist	-	-	-	0.34	<u>0.39</u>	**0.56**

Table 1. 8 sequences were analyzed quantitatively using the average PASCAL overlap with the ground truth. The best results are printed in bold faced letters, the second best results are underlined.

Ground truth.   Our ground truth captures the visible part of the target *object* by a rectangular bounding box, *i.e.*, it takes scale as well as aspect ratio changes into account. As many state-of-the-art trackers [40,39,23,8,33] rely on a fixed scale and fixed aspect ratio, we defined the IFSART as a purely hypothetical tracker. It indicates the maximum possible performance of such methods. For these trackers the IFSART performance was typically used as ground truth even though it contained only parts of the target object, large parts of other distracting objects, or the background. MIL is a representative of a tracker with fixed aspect ratio and scale, TLD is scale adaptive, and SPT and HTT output an object segmentation. Please note that all approaches are evaluated against our ground truth.

Discussion.   The strong assumption of IFSART does only hold in very restricted cases, notably the *Dollar* sequence with a rectangular target object that precisely fits a rectangular bounding box with a fixed aspect ratio. In such

---

[5] http://ice.dlut.edu.cn/lu/iccv_spt_webpage/iccv_spt.htm, 2012/02/29

[6] http://lrs.icg.tugraz.at/research/houghtrack, 2012/03/05

[7] https://github.com/zk00006/OpenTLD, 2012/02/29

Fig. 6. In contrast to the image segmentation based tracker using hough transform (HTT, 1st and 3rd column) our approach (2nd and 4th column) based on Dynamic Objectness includes motion information which is often a more dominant cue.

cases IFSART and also the TLD (which also uses a rectangular template for updating) outperform our method.

TLD uses sparse optical flow to learn an object model much like our method. There are, however, notable differences. Firstly, our approach tracks the object and the background to not easily integrate other distracting objects. Secondly, our approach additionally integrates motion segmentation to sporadically rediscover the object. These differences led to superior performance in more challenging sequences where the object does not tightly fit a bounding box like the *FaceOcc* sequences or the *Board* sequence. Also abrupt motion and unreliable optical flow, as in *Fist*, are better coped with the help of Dynamic Objectness.

The recent SPT and HTT also incorporate object segmentation similar to our method. However, we employ motion segmentation on relatively few feature points, whereas SPT relies on slower image segmentation. Our superior results suggest that motion segmentation is often a more valuable cue than image segmentation to robustify tracking. As illustrated in Fig. 6 the object often moves differently from its surrounding background. Therefore our approach is able to more precisely encompass the object.

4.3   Qualitative Tracking Results

Additional results on the Dudek sequence and the Lipinski sequence are provided for a qualitative impression of our tracking approach. They include a wide range of difficulties such as cluttered background, similar objects, moving cameras, changing illumination, pose and scale changes as well as re-detection of the object after disappearance. Many state-of-the-art tracking methods are not aware of occlusion and cannot adapt the object region. Moreover, they continuously update the object model and risk of drifting to other objects or the background. In contrast, our approach updates the object model only very carefully with the help of the Dynamic Objectness. Robust adaption to unknown aspects of the object is achieved.

5   Conclusion

We proposed a novel motion segmentation technique, coined Dynamic Objectness, for adaptive object tracking. Dynamic Objectness allows to sporadically

re-discover the tracked object in order to correctly adapt to unseen views. The object is not segmented in every frame as in other methods, but only in case of compact motion clusters and for the purpose of adaptation. Additionally, the approach incrementally builds an appearance model of the object for an eventual re-detection after partial or full occlusion. Finally, the approach not only follows the object, but also the background to not easily drift to other distracting objects. We evaluated our approach on several well-known tracking sequences and demonstrated results with superior accuracy especially in difficult sequences with occlusion and rapid appearance changes.

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys **38** (2006)
2. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence **17** (1981) 185–203
3. Shi, J., Tomasi, C.: Good Features to Track. In: Proceedings of IEEE CVPR. (1994)
4. Comaniciu, D., Ramesh, V., Meer, P.: Real-Time Tracking of Non-Rigid Objects using Mean Shift. In: Proceedings of IEEE CVPR. (2000)
5. Lim, J., Ross, D., Lin, R., Yang, M.: Incremental Learning for Robust Visual Tracking. In: IJCV. Number 77. (2008) 125–141
6. Godec, M., Roth, P., Bischof, H.: Hough-based Tracking of Non-Rigid Objects. In: Proceedings of IEEE ICCV. (2011)
7. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust Online Appearance Models for Visual Tracking. In: Proceedings of IEEE CVPR. (2001)
8. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: Proceedings of IEEE CVPR. (2006)
9. Tang, F., Brennan, S., Zhao, Q., Tao, H.: Co-Tracking Using Semi-Supervised Support Vector Machines. In: Proceedings of IEEE CVPR. (2007)
10. Kwon, J., Lee, K.M.: Tracking by Sampling Trackers. In: Proceedings of IEEE ICCV. (2011)
11. Cehovin, L., Kristan, M.: An adaptive coupled-layer visual model for robust visual tracking. Proceedings of IEEE CVPR (2011)
12. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proceedings of IEEE CVPR. (1999)
13. Haritaoglu, I., Harwood, D., Davis, L.S.: W4: Real-Time Surveillance of People and Their Activities. IEEE PAMI **22** (2000) 809–830
14. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-Camera People Tracking with a Probabilistic Occupancy Map. IEEE PAMI **30** (2008) 267–282
15. Özuysal, M., Lepetit, V., Fleuret, F., Fua, P.: Feature Harvesting for Tracking-by-Detection. In: Proceedings of ECCV. (2006)
16. Leibe, B., Schindler, K., Cornelis, N., Van Gool, L.: Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles. IEEE PAMI **30** (2008) 1683–1698

17. Hager, G.D., Belhumeur, P.N.: Efficient Region Tracking With Parametric Models of Geometry and Illumination. IEEE PAMI **20** (1998) 1025–1039
18. Baker, S., Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework. IJCV **56** (2004) 221–255
19. Brox, T., Malik, J.: Object Segmentation by Long Term Analysis of Point Trajectories. In: Proceedings of ECCV. (2010)
20. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models-their training and application. CVIU **61** (1995) 38–59
21. Isard, M., Blake, A.: CONDENSATION – Conditional Density Propagation for Visual Tracking. IJCV **29** (1998) 5–28
22. Collins, R.T., Liu, Y., Leordeanu, M.: Online Selection of Discriminative Tracking Features. Proceedings of IEEE ICCV (2003)
23. Grabner, H., Bischof, H.: On-line Boosting and Vision. In: Proceedings of IEEE CVPR. (2006)
24. Mahadevan, V., Vasconcelos, N.: Saliency-based Discriminant Tracking. In: Proceedings of IEEE CVPR. (2009)
25. Dinh, T.B., Vo, N., Medioni, G.: Context Tracker : Exploring Supporters and Distracters in Unconstrained Environments. Proceedings of IEEE CVPR (2011)
26. Ren, X., Malik, J.: Tracking as Repeated Figure/Ground Segmentation. In: Proceedings of IEEE CVPR. (2007)
27. Everingham, M., Sivic, J., Zisserman, A.: Taking the bite out of automated naming of characters in TV video. Image and Vision Computing **27** (2009) 545–559
28. Matthews, I., Ishikawa, T., Baker, S.: The Template Update Problem. IEEE PAMI **26** (2004) 810–815
29. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised On-line boosting for Robust Tracking. In: Proceedings of ECCV. (2008)
30. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-Learning-Detection (in press). IEEE PAMI (2012)
31. Leichter, I., Lindenbaum, M., Rivlin, E.: A General Framework for Combining Visual Trackers – The "Black Boxes" Approach. IJCV **67** (2006) 343–363
32. Perez, P., Vermaak, J., Blake, A.: Data Fusion for Visual Tracking With Particles. Proceedings of the IEEE **92** (2004) 495–513
33. Stenger, B., Woodley, T., Cipolla, R.: Learning to Track with Multiple Observers. In: Proceedings of IEEE CVPR. (2009)
34. Heitz, G., Koller, D.: Learning Spatial Context: Using Stuff to Find Things. In: Proceedings of ECCV. (2008)
35. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: Proceedings of IEEE CVPR. (2010)
36. Feng, J., Wei, Y., Tao, L., Zhang, C., Sun, J.: Salient Object Detection by Composition. In: Proceedings of IEEE CVPR. (2011)
37. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV **59** (2004) 167–181
38. Grossberg, S.: Competitive Learning: From Interactive Activation to Adaptive Resonance. Cognitive Science **11** (1987) 23–63
39. Babenko, B., Yang, M.H., Belongie, S.: Visual Tracking with Online Multiple Instance Learning. In: Proceedings of IEEE CVPR. (2009)
40. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: PROST: Parallel Robust Online Simple Tracking. In: Proceedings of IEEE CVPR. (2010)
41. Wang, S., Lu, H., Yang, F., Yang, M.H.: Superpixel Tracking. In: Proceedings of IEEE ICCV. (2011)