In Defense of Color-based Model-free Tracking

Horst Possegger* Thomas Mauthner* Horst Bischof
Institute for Computer Graphics and Vision, Graz University of Technology
{possegger,mauthner,bischof}@icg.tugraz.at

Abstract

In this paper, we address the problem of model-free online object tracking based on color representations. According to the findings of recent benchmark evaluations, such trackers often tend to drift towards regions which exhibit a similar appearance compared to the object of interest. To overcome this limitation, we propose an efficient discriminative object model which allows us to identify potentially distracting regions in advance. Furthermore, we exploit this knowledge to adapt the object representation beforehand so that distractors are suppressed and the risk of drifting is significantly reduced. We evaluate our approach on recent online tracking benchmark datasets demonstrating state-of-the-art results. In particular, our approach performs favorably both in terms of accuracy and robustness compared to recent tracking algorithms. Moreover, the proposed approach allows for an efficient implementation to enable online object tracking in real-time.

1. Introduction

Visual object tracking is a fundamental task for a wide range of computer vision applications. Domains such as visual surveillance, robotics, human-computer interaction, and augmented reality require robust and reliable location estimates of a target throughout an image sequence. Despite significant progress in recent years, creating a generic object tracker is still rather challenging due to real-world phenomena such as illumination changes, background clutter, fast object motion changes, and occlusions.

Although some application domains allow us to incorporate strong assumptions about the target (*e.g.* pedestrian tracking [11, 36, 37, 40]), it is often desirable to build a generic tracker which can readily be used for arbitrary object classes. Such model-free online trackers neither apply pre-learned object models nor exploit class-specific prior knowledge. Instead, a representative object model must be learned given a single input frame with a (possibly noisy) initial object annotation, *e.g.* an axis-aligned bounding box.

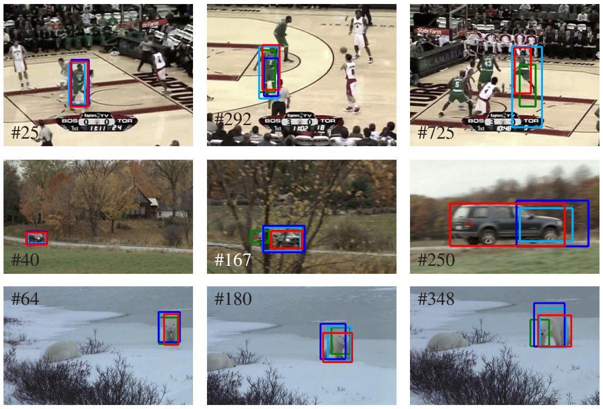


Figure 1: Comparison of the proposed approach with the state-of-the-art trackers ACT, DSST, and KCF. The example frames show the VOT14 *basketball*, *car*, and *polarbear* sequences, respectively. Best viewed in color.

Among earlier tracking approaches, color histograms (*e.g.* [12, 34, 35]) were a common method for appearance description. However, over the last decade the research focus has shifted to trackers based on well engineered features such as HOG (*e.g.* [13, 23]), correlation filters (*e.g.* [10, 22]), and more complex color features, such as color attributes [14]. Such trackers have been shown to achieve excellent performance on recent benchmark evaluations (*e.g.* [27]), whereas trackers based on standard color models yield inferior performance.

In contrast to this development, we argue that trackers based on standard color representations can still achieve state-of-the-art performance. We exploit the observation that color-based trackers tend to drift towards nearby regions with similar appearance. Using an adaptive object model which is able to suppress such regions, we can significantly reduce the drifting problem, yielding robust and reliable tracking results. Due to the favorable simplicity of our representation, it is well suited for time-critical applications such as surveillance and robotics.

Our contributions are as follows. We present a discrimi-

*Both authors contributed equally.

native object model capable of differentiating the object of interest from the background. Although it relies on standard color histograms, this representation already achieves state-of-the-art performance on a variety of challenging sequences. We extend this representation to identify and suppress distracting regions in advance which significantly improves the tracking robustness. Additionally, we propose an efficient scale estimation scheme which allows us to obtain accurate tracking results as illustrated in Figure 1. Finally, we extensively evaluate our approach on recent benchmark datasets to demonstrate its favorable performance compared to a variety of state-of-the-art trackers.

2. Related Work

In order to model the object appearance, tracking approaches can either rely on generative or discriminative representations. Generative approaches locate the object by seeking the region most similar to a reference model. Such trackers are typically based on templates (*e.g.* [2, 20, 31]), color representations (*e.g.* [12, 34, 35]), subspace models (*e.g.* [9, 38]), and sparse representations (*e.g.* [6, 24, 30]). However, typical rectangular initialization bounding boxes always include background information which is also captured by the model. Although several approaches leverage segmentation methods (*e.g.* [7, 17]) to improve the generative model, these still suffer from missing discriminative capabilities to distinguish the object from its surrounding background.

Recent benchmark evaluations (*e.g.* [27, 44]) show that generative models are often outperformed by discriminative approaches which incorporate binary classifiers capable of distinguishing the object from the background. Such trackers exploit templates (*e.g.* [22]), color cues (*e.g.* [8, 14, 32]), Haar-like features (*e.g.* [4, 21, 47]), HOG features (*e.g.* [13, 23]), and binary patterns (*e.g.* [15, 25]) to model the object appearance. Such models can either be represented in a holistic way (*e.g.* [22, 47]) or by parts (*e.g.* [42, 16]) and patches (*e.g.* [29, 30, 32]) which have been shown to perform favorably when tracking highly non-rigid objects or considering partial occlusions. Due to the success of discriminative models, a large variety of suitable learning methods has been explored for visual tracking, such as structured output SVMs [21], ranking SVMs [5], boosting [3, 18], kernel ridge regression [22, 23], and multiple-instance learning [4].

To further improve performance, several trackers incorporate contextual information (*e.g.* [15, 19, 45, 48]). Such approaches distinguish between context provided by supporting and distracting regions. Supporting regions as used by [15, 19] have different appearance than the target but co-occur with it, providing valuable cues to overcome occlusions. Distractors, on the other hand, exhibit similar appearance and may therefore be confused with the target. Typically, context-aware trackers such as [45, 48] assume that distractors are of the same object class (*e.g.* pedestrians) and need to track these distractors in addition to the target to prevent drifting. In contrast to these approaches, we impose no assumptions on the object class of distractors. Moreover, we adapt the object representation such that potentially distracting regions are suppressed in advance and thus, no explicit tracking of distractors is required.

3. Distractor-Aware Online Tracking

We base our tracking approach on two primary requirements for online model-free trackers: First, considering subsequent frames, useful object models must be able to distinguish the object from its current surrounding background. Second, to reduce the risk of drifting towards regions which exhibit similar appearance at a future time step, such distracting regions must be identified beforehand and should be suppressed to ensure a robust tracking performance. Therefore, we propose a discriminative object model which addresses these key requirements in Section 3.1. Based on this representation, Section 3.2 demonstrates how the object can be robustly localized throughout a video sequence. Furthermore, our discriminative model allows for efficient scale estimation, as will be discussed in Section 3.3.

3.1. Distractor-Aware Object Model

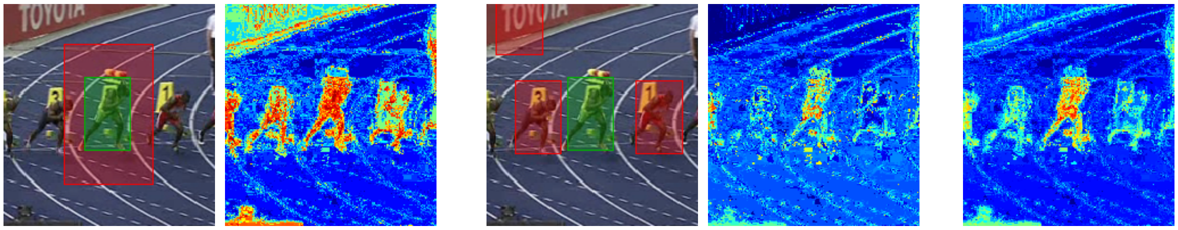
To distinguish object pixels $\mathbf{x} \in \mathcal{O}$ from surrounding background pixels, we employ a color histogram based Bayes classifier on the input image I. Let $H_{\Omega}^{I}(b)$ denote the b-th bin of the non-normalized histogram H computed over the region $\Omega \in I$. Additionally, let $b_{\mathbf{x}}$ denote the bin b assigned to the color components of $I(\mathbf{x})$. Given a rectangular object region O (*i.e.* initial bounding box annotation or current tracker hypothesis) and its surrounding region S (see Figure 2a), we apply Bayes rule to obtain the object likelihood at location $\mathbf{x}$ as

$$P(\mathbf{x}\in\mathcal{O}|O,S,b_{\mathbf{x}}) \approx \frac{P(b_{\mathbf{x}}|\mathbf{x}\in O)\, P(\mathbf{x}\in O)}{\sum\limits_{\Omega\in\{O,S\}} P(b_{\mathbf{x}}|\mathbf{x}\in\Omega)\, P(\mathbf{x}\in\Omega)}. \tag{1}$$

In particular, we estimate the likelihood terms directly from color histograms, *i.e.* $P(b_{\mathbf{x}}|\mathbf{x}\in O) \approx H_{O}^{I}(b_{\mathbf{x}})/|O|$ and $P(b_{\mathbf{x}}|\mathbf{x}\in S) \approx H_{S}^{I}(b_{\mathbf{x}})/|S|$, where $|\cdot|$ denotes the cardinality. Furthermore, the prior probability can be approximated as $P(\mathbf{x}\in O) \approx |O|/(|O|+|S|)$. Then, Eq. (1) simplifies to

$$P(\mathbf{x}\in\mathcal{O}|O,S,b_{\mathbf{x}}) = \begin{cases} \frac{H_{O}^{I}(b_{\mathbf{x}})}{H_{O}^{I}(b_{\mathbf{x}})+H_{S}^{I}(b_{\mathbf{x}})} & \text{if } I(\mathbf{x})\in I(O\cup S) \\ 0.5 & \text{otherwise,} \end{cases} \tag{2}$$

where unseen pixel values are assigned the maximum entropy prior of 0.5. This discriminative model already allows us to distinguish object and background pixels, see Figure 2a.



(a) Object-surrounding: $P(\mathbf{x}\in\mathcal{O}|O,S,b_{\mathbf{x}})$.

(b) Object-distractors: $P(\mathbf{x}\in\mathcal{O}|O,D,b_{\mathbf{x}})$.

(c) Combined.

Figure 2: Exemplary object likelihood maps for (a) the object-surrounding model and (b) the distractor-aware model illustrating the corresponding regions O, S, and D superimposed on the input images. Combining both models in (c) provides a valuable cue for localization. Hot colors correspond to high object likelihood scores. Best viewed in color.

However, one of the most common problems of color-based online trackers remains. Namely, that such algorithms may drift to nearby regions which exhibit a similar appearance compared to the object of interest. To overcome this limitation, we explicitly extend the object model to suppress such distracting regions. Since computing the object likelihood scores from Eq. (2) can be realized via an efficient lookup-table, these scores can be computed over a large search region at a very low computational cost. As will be discussed in Section 3.2, this allows us to identify potentially distracting regions in advance and handle them accordingly.

For now, let us assume we are given the current object hypothesis O and a set D of potentially distracting regions, as illustrated in Figure 2b. We can exploit this information to build a representation capable of distinguishing object and distractor pixels. Thus, similar to Eq. (2) we define the object-distractor model as

$$P(\mathbf{x}\in\mathcal{O}|O,D,b_{\mathbf{x}})=\begin{cases}\frac{H_O^I(b_{\mathbf{x}})}{H_O^I(b_{\mathbf{x}})+H_D^I(b_{\mathbf{x}})} & \text{if } I(\mathbf{x})\in I(O\cup D)\\ 0.5 & \text{otherwise.}\end{cases} \quad (3)$$

Combining the object-background model with the above distractor-aware representation, we obtain the final object model as $P\left(\mathbf{x}\in\mathcal{O}\,|\,b_{\mathbf{x}}\right)=\lambda_p P\left(\mathbf{x}\in\mathcal{O}\,|\,O,D,b_{\mathbf{x}}\right)+(1-\lambda_p)P\left(\mathbf{x}\in\mathcal{O}\,|\,O,S,b_{\mathbf{x}}\right)$, where λ_p is a pre-defined weighting parameter. Applying the combined object model (see Figure 2c) yields high likelihood scores for discriminative object pixels while simultaneously decreasing the impact of distracting regions. To adapt the representation to changing object appearance and illumination conditions, we update the object model on a regular basis using the linear interpolation $P_{1:t}\left(\mathbf{x}\in\mathcal{O}\,|\,b_{\mathbf{x}}\right)=\eta P\left(\mathbf{x}\in\mathcal{O}\,|\,b_{\mathbf{x}}\right)+(1-\eta)P_{1:t-1}\left(\mathbf{x}\in\mathcal{O}\,|\,b_{\mathbf{x}}\right)$, with learning rate η.

3.2. Localization

We adopt the widely used tracking-by-detection principle to localize the object of interest within a new frame at time t. In particular, we extract a rectangular search region proportional to the previous object location O_{t-1} and obtain the new target location $O_t^\star$ as

$$O_t^\star=\underset{O_{t,i}}{\arg\max}(s_v(O_{t,i})s_d(O_{t,i})), \quad (4)$$

$$s_v(O_{t,i})=\sum_{\mathbf{x}\in O_{t,i}}P_{1:t-1}\left(\mathbf{x}\in\mathcal{O}|b_{\mathbf{x}}\right), \quad (5)$$

$$s_d(O_{t,i})=\sum_{\mathbf{x}\in O_{t,i}}\exp\left(-\frac{\|\mathbf{x}-\mathbf{c}_{t-1}\|^2}{2\sigma^2}\right), \quad (6)$$

where $s_v(\cdot)$ denotes the vote score based on the combined object model and $s_d(\cdot)$ is the distance score based on the Euclidean distance to the previous object center $\mathbf{c}_{t-1}$. This distance term penalizes large inter-frame movements, similar to the Gaussian and cosine windowing approaches used by correlation based trackers such as [10, 13, 23].

We densely sample overlapping candidate hypotheses $O_{t,i}$ within the search region and compute both the vote and distance scores for each candidate. This allows us to efficiently obtain the new object location $O_t^\star$ as well as visually similar distractors, as such regions yield a high vote score, too. We consider a candidate $O_{t,i}$ to be a distractor if $s_v(O_{t,i})\geq\lambda_v s_v(O_t^\star)$, with $\lambda_v\in[0,1]$. To prevent selecting ambiguous distractors (*e.g.* located on the object itself due to increased scale) we follow an iterative non-maximum suppression strategy, *i.e.* after selecting a candidate (either $O_t^\star$ or a distractor) overlapping hypotheses are discarded. After obtaining both the new object location and the set of distractors, the object model is updated according to Eqs. (2) and (3) to suppress the background and identified distracting regions and thus reduce the risk of drifting at a later time step.

3.3. Scale Estimation

Similar to recent scale-adaptive state-of-the-art trackers such as [13], we first localize the object in a new frame and subsequently perform scale estimation. We exploit our object model to segment the object of interest for scale adaptation via thresholding on $P(\mathbf{x}\in\mathcal{O}\,|\,b_{\mathbf{x}})$. However, choosing a pre-defined threshold may impede the scale adaption due to background clutter or fast illumination changes. Therefore, we propose an adaptive threshold as follows.

Let L denote the object likelihood map obtained by evaluating the combined object model at every location of the search region, as shown in Figure 3a. Then, we compute the cumulative histograms $c_O^L(b) = \sum_{i=1}^{b} H_O^L(i)/|O|$ and $c_S^L(b) = \sum_{i=1}^{b} H_S^L(i)/|S|$ over the object region O and the surrounding region S, respectively (illustrated in Figure 3b). We can exploit these cumulative histograms to compute the adaptive segmentation threshold $\tau^\star$ as

$$\tau^\star = \arg\min_{\tau} \left(2c_O^L\left(b_\tau\right) - c_O^L\left(b_\tau + 1\right) + c_S^L\left(b_\tau\right)\right), \qquad (7)$$

$$\text{s.t.} \quad c_O^L\left(b_\tau\right) + c_S^L\left(b_\tau\right) \geq 1.$$

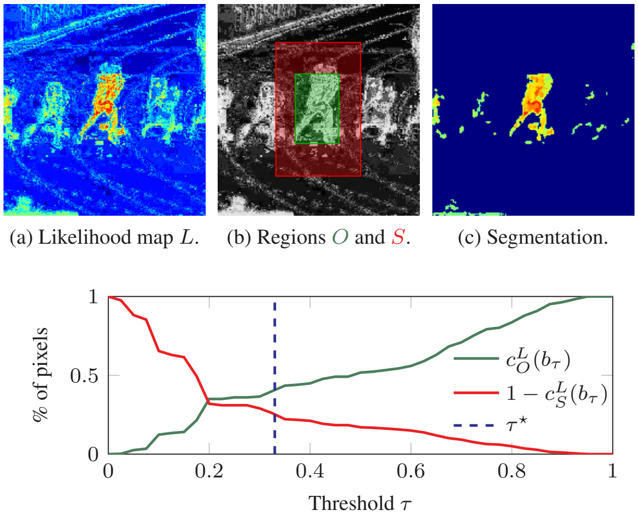
This formulation penalizes thresholds within flat regions of the cumulative object histogram c_O^L, *e.g.* thresholds within the range $[0.2, 0.3]$ in Figure 3d. The obtained threshold significantly reduces background noise while yielding a sufficiently large number of object pixels.

To adapt the scale of the current object hypothesis $O_t^\star$, we define a safe foreground region (*i.e.* the inner 80% of $O_t^\star$) and perform a connected component analysis based on the segmentation result after applying the adaptive threshold (see Figure 3c). Connected components which yield a high average object likelihood score and intersect the safe foreground region are labeled as object regions. Computing the enclosing bounding box over these regions then gives the scale estimate O_t^S for the current frame. If the estimated scale change between the current and previous frame is above a reasonable percentage, we discard the segmentation as unreliable. Otherwise, we use it to update the dimension of the object hypothesis $O_t = \lambda_s O_t^S + (1-\lambda_s) O_t^\star$. Note that in contrast to recent scale adaption approaches such as [13, 23], our scale estimation scheme is not limited to a fixed aspect ratio, as already shown in Figure 1.

4. Evaluation

We evaluate our distractor-aware tracking approach on two publicly available benchmark datasets, namely the Visual Object Tracking (VOT) challenge datasets VOT14 [27] and VOT13 [26]. Considering the number of submitted tracking approaches, these challenges are the largest model-free tracking benchmarks to date. In the following, we focus on a detailed comparison of our approach with state-of-the-art tracking algorithms. Additional visual results are included in the supplemental material.

Dataset characteristics. The sequences contained in the VOT datasets have been collected from well-known tracking evaluations, such as the experimental survey on the Amsterdam Library of Ordinary Videos (ALOV) [41], the Online Tracking Benchmark (OTB) [44], as well as recently published video sequences from various authors (including [1, 17, 28, 29, 38]). In particular, the VOT committee proposed a sequence selection methodology to compile datasets which cover various real-life visual phenomena while keeping the number of sequences reasonably low. In total, the datasets consist of 16 (VOT13) and 25 (VOT14) sequences capturing severe illumination changes, object deformations and appearance changes, abrupt motion changes, significant scale variations, camera motion, and occlusions.



(a) Likelihood map L. (b) Regions O and S. (c) Segmentation.

(d) Cumulative histograms for estimating the adaptive threshold $\tau^\star$.

Figure 3: Segmentation example. The corresponding object and surrounding regions in (b) have been superimposed on a grayscale representation of (a). See text for details.

Evaluation protocol. We follow the protocol of the VOT benchmark challenges, *i.e.* trackers are initialized at the first frame of a video using the ground truth annotation and re-initialized once they drift away from the target. The VOT framework provides a standardized analysis using two weakly correlated performance metrics, namely *Accuracy*[1] (average bounding box overlap) and *Robustness*[2] (number of re-initializations).

Additionally, the VOT framework provides a ranking analysis based on these metrics. This ranking considers the statistical significance of performance differences to ensure a fair comparison. Trackers are equally ranked if there is only a negligible difference from a practical point of view. For a detailed description of the evaluation and ranking methodology, we refer the interested reader to [26, 27].

Following the VOT evaluation protocol, we keep the parameters fixed throughout all experiments. We model the joint distribution of color values in the RGB color cube

[1] Higher is better (denoted by ↑). [2] Lower is better (denoted by ↓).

Tracker	Accuracy		Robustness		Combined
	Score$^\uparrow$	Rank$^\downarrow$	Score$^\downarrow$	Rank$^\downarrow$	Rank$^\downarrow$
ACT [14]	0.53	7.66	1.48	7.38	7.52
CMT [33]	0.48	8.89	2.64	9.14	9.02
DSST [13]	0.62	4.78	1.16	6.44	5.61
FoT [43]	0.51	8.37	2.28	9.54	8.95
IIVT [46]	0.47	9.94	3.19	9.66	9.80
KCF [23]	0.62	4.48	1.32	6.76	5.62
LGT [42]	0.46	9.33	0.66	6.20	7.77
MIL [4]	0.39	11.69	2.27	8.96	10.32
OGT [32]	0.54	7.23	3.34	9.86	8.55
PT [16]	0.44	11.02	1.40	7.20	9.11
SPOT [48]	0.48	9.92	2.16	9.36	9.64
Struck [21]	0.51	8.31	2.16	8.84	8.57
noDAT	0.55	6.43	3.68	9.72	8.08
DAT	0.56	6.80	1.08	5.72	6.26
DATs	0.61	5.05	0.84	5.14	5.09

(a) Results VOT14.

Tracker	Accuracy		Robustness		Combined
	Score$^\uparrow$	Rank$^\downarrow$	Score$^\downarrow$	Rank$^\downarrow$	Rank$^\downarrow$
ACT [14]	0.60	6.43	0.94	8.12	7.28
CT [47]	0.47	12.52	1.77	9.59	11.06
DFT [39]	0.60	6.56	1.31	9.16	7.86
FoT [43]	0.63	6.05	1.38	8.38	7.21
HT [17]	0.48	11.37	3.64	8.97	10.17
IVT [38]	0.61	6.57	1.81	9.09	7.83
KCF [23]	0.61	6.10	0.88	7.62	6.86
LGT [42]	0.54	8.09	0.28	5.97	7.03
MIL [4]	0.52	10.30	1.48	8.59	9.45
PLT [26]	0.58	7.30	0.00	4.66	5.98
SPOT [48]	0.56	9.40	1.46	8.44	8.92
Struck [21]	0.51	8.49	3.94	7.97	8.23
TLD [25]	0.59	8.27	6.69	12.00	10.13
DAT	0.60	7.04	0.38	6.19	6.61
DATs	0.63	5.64	0.12	5.25	5.45

(b) Results VOT13.

Table 1: Average performance scores and ranking results on the (a) VOT14 and (b) VOT13 benchmark datasets. Best, second best, and third best results have been highlighted. Note that the VOT rankings are based on statistical significance of the performance metrics. See Sections 4.1 (VOT14) and 4.2 (VOT13) for details.

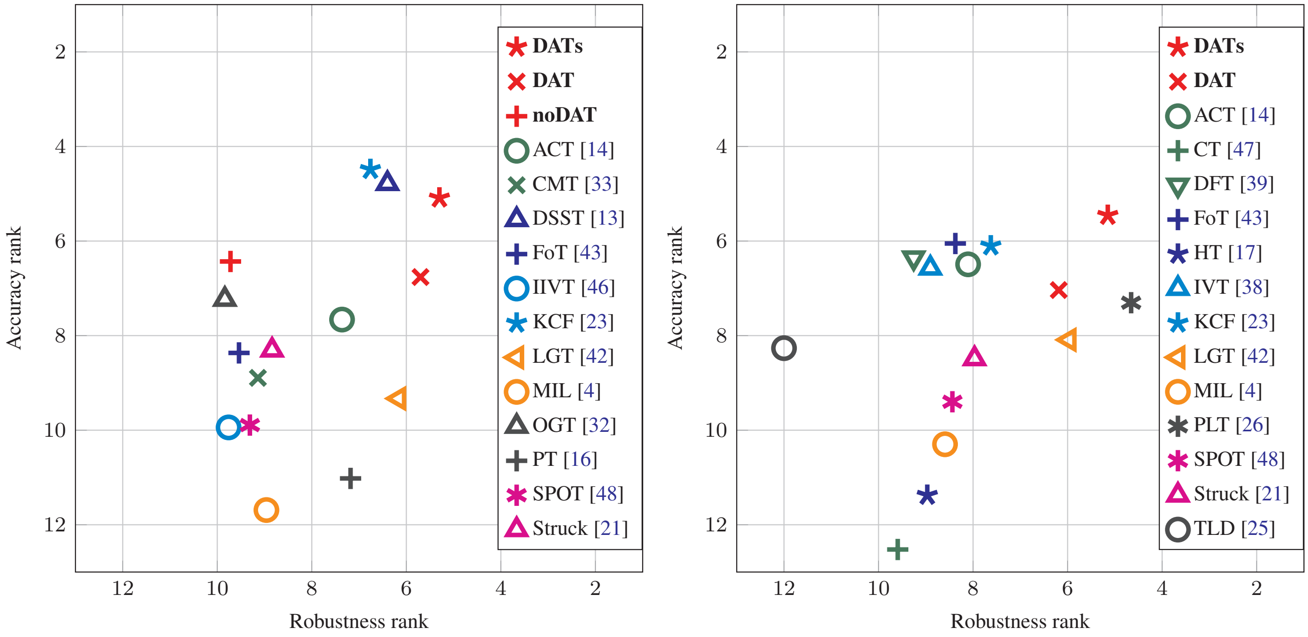
with histograms using 10 bins per channel. Additionally, we use the model weighting factor $\lambda_p = 0.5$ and the update rate $\eta = 0.1$. The search region is set to three times the dimension of the previous object hypothesis O_{t-1} and the surrounding region is twice the size of O_t. To identify distracting regions, we use the vote factor $\lambda_v = 0.5$. The scale update is performed using $\lambda_s = 0.2$.

4.1. Results VOT14 Benchmark

To ensure a fair and unbiased comparison, we use the original results submitted to the VOT14 challenge by the corresponding authors or the VOT committee (based on the corresponding publicly available implementations). We compare our approach to recent state-of-the-art algorithms including the winner of the VOT14 challenge, DSST [13], and two of the top-performing trackers of the online tracking benchmark [44], namely Struck [21] and CSK [22]. For the latter we use its recent extension, KCF [23] (*i.e.* scale-adaptive results submitted to VOT14). Furthermore, we include the color attribute based ACT [14], the keypoint based CMT [33] and IIVT [46], the part based LGT [42], OGT [32], and PT [16], the discriminative MIL [4], as well as FoT [43]. Additionally, we provide results for the context-aware SPOT [48] using their online available implementation.

Overall results. As can be seen from the ranking results in Table 1a and Figure 4a, our distractor-aware tracker (DAT) and its scale-adaptive version (DATs) rank amongst the top trackers both with respect to accuracy and robustness. In the combined ranking our approach outperforms all competitors due to its favorable robustness. We achieve accuracy scores competitive to state-of-the-art scale-adaptive trackers (*i.e*. DSST [13] and KCF [23]), while significantly reducing the drifting problem, as can be seen from the detailed robustness scores in Table 3. A key finding is that the proposed discriminative object representation significantly outperforms other color-based trackers, such as ACT [14] and OGT [32], as well as trackers based on a combination of image gradients and color information, *e.g*. PT [16]. Moreover, note that the top 3 trackers (DATs, DSST, and KCF) employ scale estimation, whereas the proposed DAT without scale adaption achieves the combined 4th rank, outperforming the remaining competitors.

Benefits of distractor-awareness. To demonstrate the importance of the distractor-aware object representation, we compare our approach (DAT) with a baseline tracker only using the discriminative object-background model (*i.e.* Eq. (2)), denoted *noDAT*. Overall, noDAT achieves mid-range performance (see Table 1a and Figure 4a) due to the fact that the discriminative model yields competitive accuracy. However, without exploiting the knowledge about potential distractors noDAT suffers from drifting, as can be seen from the detailed robustness scores in Table 3. Including the distractor-aware representation (DAT) significantly reduces this limitation for sequences with visually similar regions, *e.g. basketball*, *bolt*, *fish1*, *jogging*, and *skating*. On average, noDAT achieves a robustness score of 3.68, whereas including distractor-awareness improves



(a) Ranking results VOT14.

(b) Ranking results VOT13.

Figure 4: Ranking results on the benchmark datasets of the (a) VOT14 and (b) VOT13 challenges. Top-performing trackers are located top-right. See Sections 4.1 (VOT14) and 4.2 (VOT13) for details. Best viewed in color.

the result to 1.08 while yielding similar accuracy. Thus, the distractor-aware object model proves to be an important cue for creating a robust online tracker.

Robustness to noisy initializations. The VOT framework provides an additional experimental setup which randomly perturbs the initialization bounding boxes. According to the VOT protocol, we perform 15 runs with such noisy initializations and report the average results. Table 2 compares our approach to the top 5 performing competitors on this noise experiment. Despite the unreliable initializations, both DAT and DATs outperform the top ranking trackers DSST [13], KCF [23], LGT [42], ACT [14], and Struck [21]. The proposed object representation allows us to recover from these initialization errors and performs favorably both in terms of accuracy and robustness.

Runtime performance. Including the distractor-aware representation comes at a reasonably low computational cost. On a PC with a 3.4 GHz Intel CPU our pure MATLAB prototype of DAT runs at 17 fps, whereas tracking without distractor information (noDAT) achieves up to 18 fps on average. The scale estimation step is very efficient, as the full tracking approach (DATs) still processes 15 fps on average. Thus, the proposed DAT tracker can already be used for time-critical application domains, such as visual surveillance or robotics.

Tracker	**Accuracy**		**Robustness**		**Combined**
	Score↑	**Rank**↓	**Score**↓	**Rank**↓	**Rank**↓
ACT [14]	0.49	5.02	1.77	4.56	4.79
DSST [13]	0.57	3.10	1.28	3.98	3.54
KCF [23]	0.57	3.44	1.51	4.28	3.86
LGT [42]	0.46	5.12	0.64	3.54	4.33
Struck [21]	0.48	5.42	2.22	5.00	5.21
DAT	0.55	3.20	1.06	3.38	3.29
DATs	0.58	2.70	1.03	3.26	2.98

Table 2: Average performance scores and ranking results on the VOT14 benchmark using randomly perturbed initializations. See text for details.

4.2. Results VOT13 Benchmark

Additionally, we evaluate our approach on the VOT13 benchmark dataset. Similar to the previous evaluation, we use the original VOT13 challenge results as verified by the VOT committee. We compare our approach to the VOT13 challenge winner PLT [26] which is an extension of Struck [21]. Furthermore, we include CT [47], DFT [39], FoT [43], HT [17], IVT [38], LGT [42], MIL [4], and TLD [25]. We also report results for the recent ACT [14], KCF [23], and SPOT [48] trackers using their publicly available implementations.

The averaged performance metrics and ranking results are shown in Table 1b and Figure 4b. Again, our approaches rank amongst the top-performers of this challenge. In particular, our scale-adaptive DATs outperforms the VOT13 challenge winner PLT, while the single-scale DAT ranks third. This demonstrates that our efficient distractor-aware model performs favorably compared to rather complex color representations (*e.g*. ACT [14]) as well as state-of-the-art HOG based trackers (*e.g*. KCF [23]).

5. Conclusion

We proposed a generic object tracking approach based on very efficient discriminative color models. To overcome the drifting problem of state-of-the-art color based trackers, we identify distracting regions in advance and adapt the object representation to suppress these regions. Our detailed evaluations on recent benchmark datasets demonstrate that color based trackers can achieve competitive accuracy on challenging real-world sequences. Moreover, using the proposed distractor-aware object model significantly improves the tracking robustness, even if only noisy initializations are available. Overall, our color-based representation yields favorable performance compared to recent state-of-the-art trackers based on more complex features and achieves high frame rates suitable for time-critical applications.

Acknowledgments. This work was supported by the Austrian Science Foundation (FWF) under the project Advanced Learning for Tracking and Detection (I535-N23).

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust Fragments-based Tracking using the Integral Histogram. In *Proc. CVPR*, 2006.

[2] N. Alt, S. Hinterstoisser, and N. Navab. Rapid Selection of Reliable Templates for Visual Tracking. In *Proc. CVPR*, 2010.

[3] S. Avidan. Ensemble Tracking. *PAMI*, 29(2):261–271, 2007.

[4] B. Babenko, M.-H. Yang, and S. Belongie. Robust Object Tracking with Online Multiple Instance Learning. *PAMI*, 33(7):1324–1338, 2011.

[5] Y. Bai and M. Tang. Robust Tracking via Weakly Supervised Ranking SVM. In *Proc. CVPR*, 2012.

[6] C. Bao, Y. Wu, H. Ling, and H. Ji. Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach. In *Proc. CVPR*, 2012.

[7] V. Belagiannis, F. Schubert, N. Navab, and S. Ilic. Segmentation Based Particle Filtering for Real-Time 2D Object Tracking. In *Proc. ECCV*, 2012.

[8] C. Bibby and I. Reid. Robust Real-Time Visual Tracking using Pixel-Wise Posteriors. In *Proc. ECCV*, 2008.

[9] M. J. Black and A. D. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *IJCV*, 26(1):63–84, 1998.

[10] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual Object Tracking using Adaptive Correlation Filters. In *Proc. CVPR*, 2010.

[11] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online Multi-Person Tracking-by-Detection from a Single, Uncalibrated Camera. *PAMI*, 33(9):1820–1833, 2011.

[12] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking. *PAMI*, 25(5):564–577, 2003.

[13] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Accurate Scale Estimation for Robust Visual Tracking. In *Proc. BMVC*, 2014.

[14] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer. Adaptive Color Attributes for Real-Time Visual Tracking. In *Proc. CVPR*, 2014.

[15] T. B. Dinh, N. Vo, and G. Medioni. Context Tracker: Exploring Supporters and Distracters in Unconstrained Environments. In *Proc. CVPR*, 2012.

[16] S. Duffner and C. Garcia. PixelTrack: a fast adaptive algorithm for tracking non-rigid objects. In *Proc. ICCV*, 2013.

[17] M. Godec, P. M. Roth, and H. Bischof. Hough-based Tracking of Non-Rigid Objects. In *Proc. ICCV*, 2011.

[18] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised On-line Boosting for Robust Tracking. In *Proc. ECCV*, 2008.

[19] H. Grabner, J. Matas, L. Van Gool, and P. Cattin. Tracking the Invisible: Learning Where the Object Might be. In *Proc. CVPR*, 2010.

[20] G. D. Hager and P. N. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. *PAMI*, 20(10):1025–1039, 1998.

[21] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured Output Tracking with Kernels. In *Proc. ICCV*, 2011.

[22] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the Circulant Structure of Tracking-by-detection with Kernels. In *Proc. ECCV*, 2012.

[23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-Speed Tracking with Kernelized Correlation Filters. *PAMI*, 2015.

[24] X. Jia, H. Lu, and M.-H. Yang. Visual Tracking via Adaptive Structural Local Sparse Appearance Model. In *Proc. CVPR*, 2012.

[25] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-Detection. *PAMI*, 34(7):1409–1422, 2012.

[26] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Čehovin, G. Nebehay, G. Fernandez, T. Vojíř, et al. The Visual Object Tracking VOT2013 challenge results. In *Proc. VOT (ICCV Workshop)*, 2013.

[27] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay, T. Vojíř, G. Fernandez, et al. The Visual Object Tracking VOT2014 challenge results. In *Proc. VOT (ECCV Workshop)*, 2014.

[28] J. Kwon and K. M. Lee. Visual Tracking Decomposition. In *Proc. CVPR*, 2010.

[29] J. Kwon and K. M. Lee. Highly Non-Rigid Object Tracking via Patch-based Dynamic Appearance Modeling. *PAMI*, 35(10):2427–2441, 2013.

[30] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust Tracking Using Local Sparse Appearance Model and K-Selection. In *Proc. CVPR*, 2011.

Sequence	DATs		DAT		noDAT		ACT [14]		DSST [13]		KCF [23]		LGT [42]		OGT [32]	
	Acc↑	Rob↓	Acc↑	Rob↓	Acc↑	Rob↓	Acc↑	Rob↓	Acc↑	Rob↓	Acc↑	Rob↓	Acc↑	Rob↓	Acc↑	Rob↓
ball	0.87	0.00	0.67	0.00	0.67	0.00	0.41	0.00	0.56	1.00	0.75	1.00	0.31	1.13	0.72	0.00
basketball	0.61	1.00	0.73	1.00	0.68	26.00	0.66	0.00	0.64	1.00	0.64	0.00	0.50	0.80	0.55	8.93
bicycle	0.54	0.00	0.48	1.00	0.48	1.00	0.46	1.00	0.58	0.00	0.62	0.00	0.53	0.93	0.65	0.20
bolt	0.52	0.00	0.49	1.00	0.47	2.00	0.54	1.00	0.56	1.00	0.49	3.00	0.38	0.67	0.67	23.20
car	0.80	0.00	0.41	0.00	0.41	0.00	0.53	1.00	0.73	0.00	0.70	0.00	0.51	0.80	0.49	0.00
david	0.64	0.00	0.64	0.00	0.61	1.00	0.72	0.00	0.80	0.00	0.82	0.00	0.56	0.00	0.50	0.13
diving	0.41	1.00	0.35	1.00	0.44	5.00	0.20	4.00	0.44	1.00	0.25	4.00	0.33	1.27	0.23	4.40
drunk	0.53	0.00	0.48	0.00	0.44	4.00	0.46	0.00	0.55	0.00	0.53	0.00	0.52	0.00	0.55	1.00
fernando	0.45	0.00	0.44	0.00	0.42	0.00	0.43	1.00	0.34	1.00	0.41	1.00	0.47	0.47	0.35	1.67
fish1	0.72	0.00	0.58	2.00	0.41	6.00	0.44	0.00	0.32	1.00	0.42	3.00	0.36	0.93	0.51	2.13
fish2	0.41	2.00	0.36	3.00	0.43	4.00	0.31	5.00	0.35	4.00	0.26	6.00	0.28	1.80	0.23	5.73
gymnastics	0.58	0.00	0.57	0.00	0.57	0.00	0.51	2.00	0.63	5.00	0.53	1.00	0.48	1.00	0.56	2.73
hand1	0.59	1.00	0.61	0.00	0.63	0.00	0.41	5.00	0.21	2.00	0.56	3.00	0.55	0.00	0.57	1.33
hand2	0.59	0.00	0.56	3.00	0.56	3.00	0.39	8.00	0.52	6.00	0.49	6.00	0.49	1.20	0.49	8.87
jogging	0.72	0.00	0.75	0.00	0.80	6.00	0.70	1.00	0.79	1.00	0.79	1.00	0.35	1.00	0.61	1.73
motocross	0.35	4.00	0.34	4.00	0.34	8.00	0.47	3.00	0.42	4.00	0.36	2.00	0.41	1.00	0.20	5.40
polarbear	0.82	0.00	0.55	0.00	0.57	0.00	0.51	0.00	0.63	0.00	0.78	0.00	0.65	0.00	0.65	0.00
skating	0.51	5.00	0.50	9.00	0.52	14.00	0.50	0.00	0.59	0.00	0.68	1.00	0.32	0.40	0.56	6.93
sphere	0.78	0.00	0.71	0.00	0.71	0.00	0.72	0.00	0.92	0.00	0.90	0.00	0.64	0.00	0.50	0.00
sunshade	0.57	0.00	0.58	0.00	0.54	1.00	0.79	0.00	0.78	0.00	0.76	0.00	0.55	0.40	0.74	0.00
surfing	0.76	0.00	0.85	0.00	0.87	3.00	0.83	0.00	0.90	0.00	0.79	0.00	0.57	0.00	0.70	0.00
torus	0.82	0.00	0.75	0.00	0.65	1.00	0.79	0.00	0.81	0.00	0.85	0.00	0.63	0.00	0.74	0.53
trellis	0.50	0.00	0.51	0.00	0.51	1.00	0.58	2.00	0.80	0.00	0.79	0.00	0.48	0.00	0.68	1.40
tunnel	0.47	7.00	0.34	2.00	0.43	4.00	0.31	0.00	0.80	0.00	0.68	0.00	0.36	1.47	0.50	3.53
woman	0.65	0.00	0.63	0.00	0.68	2.00	0.66	3.00	0.79	1.00	0.74	1.00	0.36	1.13	0.63	3.67
Mean	0.61	0.84	0.56	1.08	0.55	3.68	0.53	1.48	0.62	1.16	0.62	1.32	0.46	0.66	0.54	3.34

Table 3: Detailed results for the VOT14 benchmark dataset comparing our approach to the top 5 competitors. For each sequence, we report the average accuracy (*Acc*) and robustness (*Rob*) scores. Best, second best, and third best results have been highlighted. For the non-deterministic trackers LGT and OGT, we report the results averaged over 15 runs.

[31] I. Matthews, T. Ishikawa, and S. Baker. The Template Update Problem. *PAMI*, 26(6):810–815, 2004.

[32] H. Nam, S. Hong, and B. Han. Online Graph-Based Tracking. In *Proc. ECCV*, 2014.

[33] G. Nebehay and R. Pflugfelder. Consensus-based Matching and Tracking of Keypoints for Object Tracking. In *Proc. WACV*, 2014.

[34] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An Adaptive Color-Based Particle Filter. *IVC*, 21(1):99–110, 2002.

[35] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based Probabilistic Tracking. In *Proc. ECCV*, 2002.

[36] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof. Occlusion Geodesics for Online Multi-Object Tracking. In *Proc. CVPR*, 2014.

[37] H. Possegger, S. Sternig, T. Mauthner, P. M. Roth, and H. Bischof. Robust Real-Time Tracking of Multiple Objects by Volumetric Mass Densities. In *Proc. CVPR*, 2013.

[38] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental Learning for Robust Visual Tracking. *IJCV*, 77(1-3):125–141, 2008.

[39] L. Sevilla-Lara and E. Learned-Miller. Distribution Fields for Tracking. In *Proc. CVPR*, 2012.

[40] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based Multiple-Person Tracking with Partial Occlusion Handling. In *Proc. CVPR*, 2012.

[41] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual Tracking: an Experimental Survey. *PAMI*, 36(7):1442–1468, 2014.

[42] L. Čehovin, M. Kristan, and A. Leonardis. Robust Visual Tracking using an Adaptive Coupled-layer Visual Model. *PAMI*, 35(4):941–953, 2013.

[43] T. Vojíř and J. Matas. Robustifying the Flock of Trackers. In *Proc. CVWW*, 2011.

[44] Y. Wu and M.-H. Yang. Online Object Tracking: A Benchmark. In *Proc. CVPR*, 2013.

[45] M. Yang, Y. Wu, and G. Hua. Context-Aware Visual Tracking. *PAMI*, 31(7):1195–1209, 2009.

[46] K. M. Yi, H. Jeong, B. Heo, H. J. Chang, and J. Y. Choi. Initialization-Insensitive Visual Tracking Through Voting with Salient Local Features. In *Proc. ICCV*, 2013.

[47] K. Zhang, L. Zhang, and M.-H. Yang. Real-time Compressive Tracking. In *Proc. ECCV*, 2012.

[48] L. Zhang and L. van der Maaten. Preserving Structure in Model-Free Tracking. *PAMI*, 36(4):756–769, 2014.