Accurate Scale Estimation for Robust Visual Tracking

Martin Danelljan
martin.danelljan@liu.se
Gustav Häger
hager.gustav@gmail.com
Fahad Shahbaz Khan
fahad.khan@liu.se
Michael Felsberg
michael.felsberg@liu.se

Computer Vision Laboratory
Department of Electrical Engineering
Linköping University
Linköping, Sweden

Abstract

Robust scale estimation is a challenging problem in visual object tracking. Most existing methods fail to handle large scale variations in complex image sequences. This paper presents a novel approach for robust scale estimation in a tracking-by-detection framework. The proposed approach works by learning discriminative correlation filters based on a scale pyramid representation. We learn separate filters for translation and scale estimation, and show that this improves the performance compared to an exhaustive scale search. Our scale estimation approach is generic as it can be incorporated into any tracking method with no inherent scale estimation.

Experiments are performed on 28 benchmark sequences with significant scale variations. Our results show that the proposed approach significantly improves the performance by 18.8% in median distance precision compared to our baseline. Finally, we provide both quantitative and qualitative comparison of our approach with state-of-the-art trackers in literature. The proposed method is shown to outperform the best existing tracker by 16.6% in median distance precision, while operating at real-time.

1 Introduction

Visual object tracking is a popular problem in computer vision. The problem involves estimating the location of a visual target in each frame of an image sequence. Despite significant progress in recent years, the problem is still difficult due to factors such as partial occlusion, deformation, motion blur, fast motion, illumination variation, background clutter and scale variations. Most existing approaches provide inferior performance when encountered with large scale variations in complex image sequences. In this paper, we tackle the challenging problem of scale estimation for visual tracking.

In recent years, tracking-by-detection methods [3, 9, 11, 19] have shown to provide excellent tracking performance. These approaches work by posing the task of target localization as a classification problem. The decision boundary is obtained by learning a discriminative classifier online using image patches from both the target and the background. Recently,

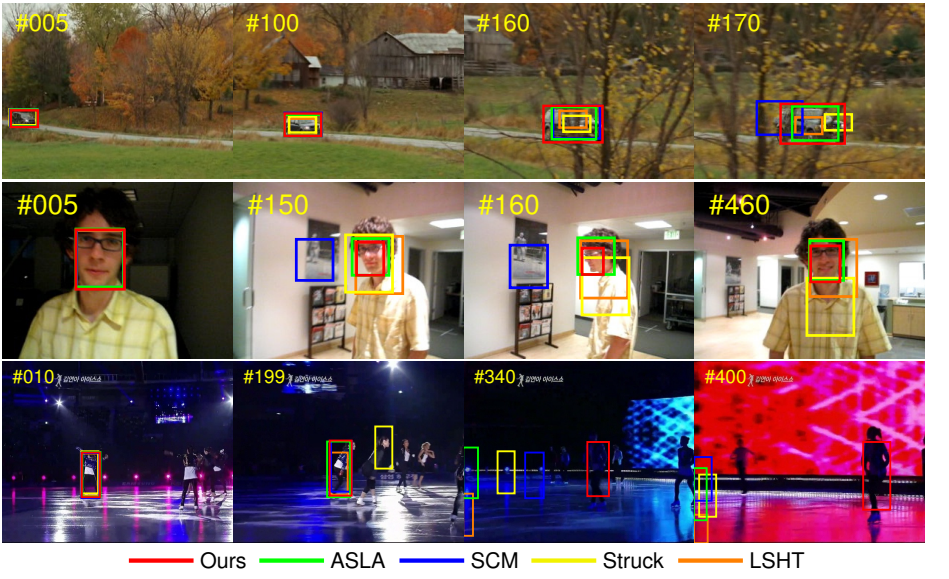


Figure 1: A comparison of our method with the state-of-the-art trackers ASLA [14], SCM [20], Struck [9] and LSHT [10] in challenging situations, namely partial occlusions, out-of-plane rotations and illumination variations. The example frames are from the *carScale*, *David* and *Skating1* sequences respectively. Our approach efficiently handles significant scale variations compared to existing approaches.

Wu *et al.* [18] performed a comprehensive evaluation of online visual tracking approaches. In their evaluation, the CSK tracker [11] is shown to provide competitive performance while possessing the highest speed. Given an image patch, the CSK tracker works by learning a kernelized least-squares classifier of the target appearance. A closely related approach, proposed by Bolme *et al.* [3], is based on finding an adaptive correlation filter by minimizing the output sum of squared error (MOSSE).

Most tracking-by-detection methods, such as the CSK and MOSSE, are limited to only estimating the target translation. This implies poor performance in sequences with significant scale variations. On the other hand, several existing approaches [14, 20] estimating scale variations operate at low frame-rates, thereby making them infeasible for real-time applications. An ideal scale estimation approach should be robust to scale changes while being computationally efficient. In this paper, we propose an efficient approach for robust scale estimation. It is based on the discriminative correlation filters employed in the MOSSE tracker [3]. The proposed scale estimation approach is generic and can be incorporated into any tracking framework.

Contributions: We propose an efficient method for estimating the target scale by training a classifier on a scale pyramid. This allows us to independently estimate the target scale after the optimal translation is found. We further show that our approach improves the accuracy over an exhaustive scale space search method, while running at 25 times faster frame-rate. To validate our approach, we perform extensive experiments on all the 28 image sequences annotated with "Scale Variation (SV)" in the recent benchmark evaluation [18]. We compare our approach with state-of-the-art trackers in literature. Despite its simplicity, our tracker achieves state-of-the-art performance, while operating at real-time. Figure 1 shows a comparison to state-of-the-art trackers on three benchmark sequences.

2 Learning Discriminative Correlation Filters

Our baseline is closely related to the MOSSE tracker [3]. The tracker learns a discriminative correlation filter used to localize the target in a new frame. The method uses a number of grayscale image patches $f_1, \dots, f_t$ of the target appearance as training samples. These are labelled with the desired correlation outputs $g_1, \dots, g_t$ from the filter. The optimal correlation filter h_t at time step t is obtained by minimizing the sum of squared errors:

$$\varepsilon = \sum_{j=1}^{t} \left\| h_t \star f_j - g_j \right\|^2 = \frac{1}{MN} \sum_{j=1}^{t} \left\| \overline{H}_t F_j - G_j \right\|^2 . \tag{1}$$

The functions f_j, g_j and h_t are all of size $M \times N$. The star $\star$ denotes circular correlation. The second equality follows from Parseval's identity. Here, capital letters denote the discrete Fourier transforms (DFTs) of the corresponding functions. The bar $\overline{H}_t$ represents complex conjugation and the product $\overline{H}_t F_j$ is point-wise. Eq. 1 is minimized by choosing:

$$H_t = \frac{\sum_{j=1}^{t} \overline{G}_j F_j}{\sum_{j=1}^{t} \overline{F}_j F_j} . \tag{2}$$

The desired correlation output g_j is constructed as a Gaussian function with its peak located at the target centre in f_j. In practice, the numerator A_t and denominator B_t of H_t in (2) are updated separately with the new observation f_t of the target by taking a weighted average.

Given an image patch z of size $M \times N$ in a new frame, the correlation scores y are computed as $y = \mathscr{F}^{-1}\{\overline{H}_t Z\}$. Here $\mathscr{F}^{-1}$ denotes the inverse DFT operator. The new target location is estimated to be at the maximum correlation score of y. The training and detection steps are performed efficiently using the fast Fourier transform (FFT). It is shown in [3] that the tracker can achieve speeds at hundreds of FPS. We refer to [3] for details and derivations.

3 Our Approach

Here we describe our approach. Section 3.1 presents how we extend the standard discriminative correlation filters to multidimensional features. In section 3.2 and 3.3 we describe our proposed scale estimation approaches.

3.1 Discriminative Correlation Filters for Multidimensional Features

The discriminative correlation filters described in section 2 have recently been extended to multi-dimensional features for a variety of applications, including visual tracking [9, 13], object detection [8, 12] and object alignment [2]. Similar to [13], we use HOG features for the translation filter and concatenate it with the usual image intensity features. However, any dense feature representation can be incorporated. The method presented here is also general in the number of dimensions of the search space. In this work, we utilize 1-dimensional filters for estimating the scale only, 2-dimensional filters for translation only and 3-dimensional filters for exhaustive scale-space localization of the target.

We consider a d-dimensional feature map representation of a signal (*e.g.* an image). Let f be a rectangular patch of the target, extracted from this feature map. We denote feature dimension number $l \in \{1, \dots, d\}$ of f by f^l. The objective is to find an optimal correlation

filter h, consisting of one filter h^l per feature dimension. This is achieved by minimizing the cost function:

$$\varepsilon = \left\| \sum_{l=1}^{d} h^l \star f^l - g \right\|^2 + \lambda \sum_{l=1}^{d} \left\| h^l \right\|^2. \tag{3}$$

Here, g is the desired correlation output associated with the training example f. The parameter $\lambda \geq 0$ controls the impact of the regularization term. Note that (3) only considers one training sample and hence generalizes (1) in the case of $t = 1$. The solution to (3) is:

$$H^l = \frac{\overline{G} F^l}{\sum_{k=1}^{d} \overline{F^k} F^k + \lambda}. \tag{4}$$

As mentioned in [3], the regularization parameter alleviates the problem of zero-frequency components in the spectrum of f, which would lead to division by zero. An optimal filter can be obtained by minimizing the output error over all training patches [2, 3]. However, this requires solving a $d \times d$ linear system of equations per pixel, which is costly for online learning applications. To obtain a robust approximation, here we update the numerator A_t^l and denominator B_t of the correlation filter H_t^l in (4) separately as:

$$A_t^l = (1-\eta) A_{t-1}^l + \eta \overline{G}_t F_t^l \tag{5a}$$

$$B_t = (1-\eta) B_{t-1} + \eta \sum_{k=1}^{d} \overline{F_t^k} F_t^k \tag{5b}$$

Here, η is a learning rate parameter. The correlation scores y at a rectangular region z of a feature map are computed using (6). The new target state is then found by maximizing the score y.

$$y = \mathscr{F}^{-1} \left\{ \frac{\sum_{l=1}^{d} \overline{A^l} Z^l}{B + \lambda} \right\} \tag{6}$$

As a baseline, we learn a filter using HOG features for only translation estimation. To train the filter, we extract the feature map f of the target patch. The filter h_{trans} is then updated using (5). We estimate the location in a new frame by extracting the feature map z at the predicted target location. The correlation scores y are then computed using (6). The following subsections describe how these filters are used for scale estimation in tracking.

3.2 Exhaustive Scale Space Tracking

We propose a method for joint translation-scale tracking based on learning a 3-dimensional scale space correlation filter. The filter size is fixed to $M \times N \times S$, where M and N are the height and width of the filter and S is the number of scales. To update the filter, we first compute a feature pyramid in a rectangular area around the target. The pyramid is constructed such that the size of the target is $M \times N$ at its estimated scale. The training sample f is then set to a rectangular cuboid of the feature pyramid. The cuboid is of size $M \times N \times S$ and is centred at the target's estimated location and scale. We use a 3-dimensional Gaussian function as the corresponding desired correlation output g. Finally, the scale space tracking filter is updated using (5).

To locate the target in a new frame, we extract a $M \times N \times S$ rectangular cuboid z from the feature pyramid as above. The cuboid is centred at the predicted target location and scale. The correlation scores y are then computed using (6). The new target location and scale are obtained by finding the maximum score in y.

Algorithm 1 Proposed tracking approach: iteration at time step t.

Input:

Image I_t.

Previous target position $\boldsymbol{p}_{t-1}$ and scale s_{t-1}.

Translation model A_{t-1}^{trans}, B_{t-1}^{trans} and scale model A_{t-1}^{scale}, B_{t-1}^{scale}.

Output:

Estimated target position $\boldsymbol{p}_t$ and scale s_t.

Updated translation model A_t^{trans}, B_t^{trans} and scale model A_t^{scale}, B_t^{scale}.

Translation estimation:

Extract a translation sample z_{trans} from I_t at $\boldsymbol{p}_{t-1}$ and s_{t-1}.
Compute the translation correlation y_{trans} using z_{trans}, A_{t-1}^{trans} and B_{t-1}^{trans} in (6).
Set $\boldsymbol{p}_t$ to the target position that maximizes y_{trans}.

Scale estimation:

Extract a scale sample z_{scale} from I_t at $\boldsymbol{p}_t$ and s_{t-1}.
Compute the scale correlation y_{scale} using z_{scale}, A_{t-1}^{scale} and B_{t-1}^{scale} in (6).
Set s_t to the target scale that maximizes y_{scale}.

Model update:

Extract samples f_{trans} and f_{scale} from I_t at $\boldsymbol{p}_t$ and s_t.
Update the translation model A_t^{trans}, B_t^{trans} using (5).
Update the scale model A_t^{scale}, B_t^{scale} using (5).

3.3 Fast Scale Space Tracking

Incorporating scale estimation into a tracker comes at a higher computational cost. Ideally, an accurate scale estimation approach should be robust while computationally efficient. To achieve this, we propose a fast scale estimation approach by learning separate filters for translation and scale. This helps by restricting the search area to smaller parts of the scale space. In addition, we gain the freedom of selecting the feature representation for each filter independently.

We augment the baseline by learning a separate 1-dimensional correlation filter to estimate the target scale in an image. The training example f for updating the scale filter is computed by extracting features using variable patch sizes centred around the target. Let $P \times R$ denote the target size in the current frame and S be the size of the scale filter. For each $n \in \left\{ \left\lfloor -\frac{S-1}{2} \right\rfloor, \ldots, \left\lfloor \frac{S-1}{2} \right\rfloor \right\}$, we extract an image patch J_n of size $a^n P \times a^n R$ centred around the target. Here, a denotes the scale factor between feature layers. The value $f(n)$ of the training example f at scale level n is set to the d-dimensional feature descriptor of J_n. Finally, (5) is used to update the scale filter h_{scale} with the new sample f.

In visual tracking scenarios, the scale difference between two frames is typically smaller compared to the translation. Therefore, we first apply the translation filter h_{trans} given a new frame. Afterwards, the scale filter h_{scale} is applied at the new target location. An example z is extracted from this location using the same procedure as for f. By maximizing the correlation output (6) between h_{scale} and z, we obtain the scale difference. Algorithm 1 provides a brief outline of our tracker. Our approach accurately estimates both translation and scale while being computationally efficient.

Method	median OP	median DP	median CLE	median FPS
Baseline (no scale)	37.8	74.5	15.9	**44.1**
Exhaustive Scale Search (this paper)	52.2	87.6	11.8	0.96
Fast Scale Search (this paper)	**75.5**	**93.3**	**10.9**	24.0

Table 1: Comparison of our fast scale estimation method with the baseline and exhaustive search trackers. Our approach significantly improves the performance, while being computationally efficient.

4 Experiments

We first show that replacing the conventional intensity values with HOG features significantly improves the performance. We then compare our fast scale estimation approach with the exhaustive method. Finally, we provide both quantitative and qualitative comparisons with state-of-the-art trackers.

4.1 Features and Parameters

The regularization parameter is set to $\lambda = 0.01$. We set the standard deviations for the desired correlation output to $1/16$ of the target size for the translation filter and 1.5 in the scale filter. The filter size $M \times N$ is set to twice the initial target size. We use $S = 33$ number of scales with a scale factor of $a = 1.02$. The learning rate is set to $\eta = 0.025$ for our methods. We use the same parameter values for all the sequences.

We use PCA-HOG [7] for image representation. The implementation provided by [8] is employed in this work. To achieve a pixel-dense feature representation for the translation filter, the cell size is set to 1×1. We further augment the HOG feature vector with the image intensity (grayscale) value. This feature representation is also employed in the exhaustive scale-space filter described in section 3.2. For the scale filter introduced in section 3.3 we compute a feature descriptor of the image patch J_n by first resizing the patch to a fixed size. PCA HOG features are then extracted using a cell size of 4×4. The fixed patch size is set to the initial target size. However, for targets with an initial area larger than 512 pixels, we calculate a fixed size with a preserved aspect ratio and an area of 512 pixels. This ensures a maximum feature descriptor length of 992. Finally, the extracted features are always multiplied by a Hann window, as described in [3].

4.2 Experimental Setup

The approach proposed in this paper is implemented in Matlab. We perform the experiments on an Intel Xenon 2 core 2.66 GHz CPU with 16 GB RAM.

Datasets: We employ all the 28 sequences[1] annotated with the scale variation attribute in the recent evaluation of tracking methods [18]. The sequences also pose challenging problems such as illumination variation, motion blur, background clutter and occlusion.

Evaluation Methodology: The performance of our approach is quantitatively validated by following the protocol[1] used in [18]. We present the results using distance precision (DP), centre location error (CLE) and overlap precision (OP). The first metric, CLE, is computed

[1] The sequences together with the ground-truth and matlab code is available at:
https://sites.google.com/site/trackerbenchmark/benchmarks/v10

	Ours	ASLA [8]	SCM [28]	Struck [9]	TLD [13]	EDFT [6]	L1APG [2]	DFT [18]	LOT [14]	CSK [10]	LSHT [11]	CT [29]
Median OP	**75.5**	*69.9*	56.6	43.1	37.9	33.6	31.1	30.3	28.9	28.2	27.9	17.9
Median DP	**93.3**	70.8	64.3	*76.8*	45.4	48.2	38.1	41.4	48	55.5	55.9	25.7
Median CLE	**10.9**	30.2	23.5	*14.3*	48.1	78.6	62.3	58.5	60.9	35.1	31.2	71.5
Median FPS	24	0.959	0.0828	8.96	21	20.6	1.01	10.5	0.517	**152**	12.5	*69.1*

Table 2: Comparison with state-of-the-art trackers on the 28 benchmark sequences. Our approach significantly outperforms existing methods in overlap precision (OP) (%), distance precision (DP) (%) and centre location error (CLE) (in pixels). Additionally our method is faster compared to the best performing existing trackers: Struck, ASLA and SCM.

	boy	car4	carScale	couple	crossing	david	dog1	doll	dudek	fleetface	freeman1	freeman3	freeman4	girl
Ours	**100**	**100**	**84.5**	10.7	**100**	**100**	**100**	**99.6**	**98.1**	66.5	**36.8**	31.3	41.7	24.2
ASLA	43.5	**100**	71	22.1	**100**	94.9	89.9	92.2	90.5	64.5	32.8	91.7	17	86.8
SCM	43.9	37.5	66.7	47.1	**100**	28	85.3	99.3	80.3	**86.6**	32.2	**99.8**	**57.6**	35.6
Struck	97.5	39.9	43.3	**60.7**	95.8	23.6	65.2	68.9	**98.1**	78.1	20.2	17.6	18.7	**97**
LSHT	50.7	27.6	44.8	9.29	40	28.2	54.3	23	89.9	65.5	18.4	15.7	20.1	14.4
TLD	82.9	24	68.7	22.9	45.8	61.1	75.6	69.3	67	44.1	23.3	64.6	21.6	72.6

	ironman	lemming	liquor	matrix	mRolling	shaking	singer1	skating1	skiing	soccer	trellis	walking	walking2	woman
Ours	13.3	26.9	40.9	**18**	6.71	**100**	**100**	54.8	4.94	**52.8**	**96.8**	**99.8**	**100**	93.3
ASLA	**15.1**	16.7	68.8	7	9.76	23.3	**100**	**73.3**	**12.3**	10.5	85.8	**99.8**	39.8	88.6
SCM	9.04	16.6	37	17	9.15	55.6	**100**	64.8	9.88	18.1	87.2	95.1	**100**	78.2
Struck	4.82	**63.8**	40.5	11	15.2	52.9	29.9	31.3	4.94	18.1	54	52.4	43	**93.5**
LSHT	2.41	40.7	60.1	2	9.15	69.9	27.6	18.3	3.7	9.18	44.3	54.4	38.4	83.9
TLD	8.43	63.4	**75.5**	7	**17.7**	3.29	98.6	9	7.41	11.2	40.6	35.2	20.8	32.8

Table 3: Per-video overlap precision (OP) in percent on the 28 sequences. The best results are reported in bold. Our approach performs favourably compared to existing trackers.

as the average Euclidean distance between the ground-truth and the estimated centre location of the target. The second metric, DP, is computed as the relative number of frames in the sequence where the centre location error is smaller than a certain threshold. The DP values at a threshold of 20 pixels [11, 18] are reported. The third metric, OP, is defined as the percentage of frames where the bounding box overlap surpasses a threshold $t \in [0,1]$. We report the results at a threshold of 0.5, which correspond to the PASCAL evaluation criteria. We provide results using median DP, CLE and OP over all 28 sequences. We also report the speed of the trackers in median frames per second (FPS) over all the 28 sequences.

In addition, the results are presented using precision and success plots [18]. The average distance precision is plotted over a range of thresholds in the precision plot. In the legend, we report the average DP score at 20 pixels for each method. The average overlap precision (OP) is plotted in the success plot. The *area under the curve* (AUC) is included in the legend. The precision and success plots provide the *mean* results over all the 28 sequences. Finally, we provide the qualitative analysis of our approach with existing tracking methods.

4.3 Experiment 1: Image Representation using HOG

The intensity based baseline roughly corresponds to the MOSSE tracker proposed in [3], but without any explicit failure detection component. The HOG based image representation significantly improves the tracking performance by 11.6% and 6.9% in median distance precision (DP) and overlap precision (OP) respectively. Similarly, the HOG based tracker reduces the median CLE from 31.2 to 15.9 pixels. In summary, our results clearly suggest that the HOG based image representation, popular in object detection, also improves the performance for visual tracking. Due to its performance, we also employ the HOG features for image representation in our scale estimation approach.

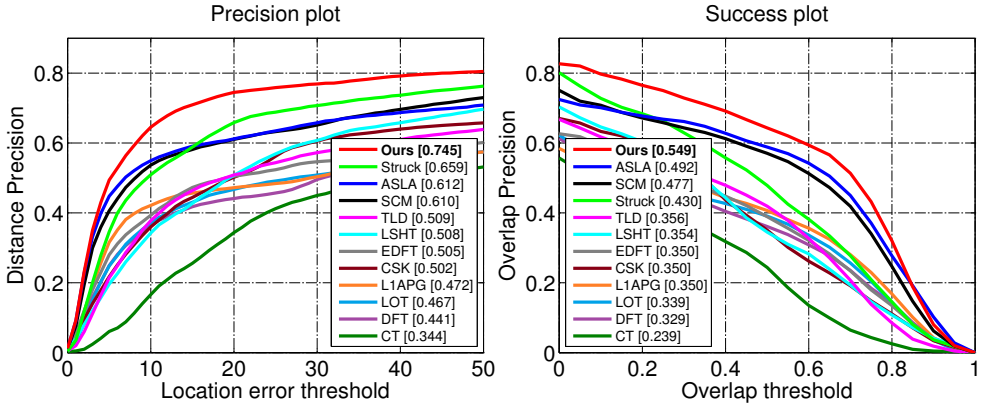


Figure 2: Precision and success plots over all the 28 sequences. The average DP score at 20 pixels for each method is reported in the legend of the precision plot. The legend of the success plot contains the *area-under-the-curve* (AUC) score for each tracker.

4.4 Experiment 2: Robust Scale Estimation

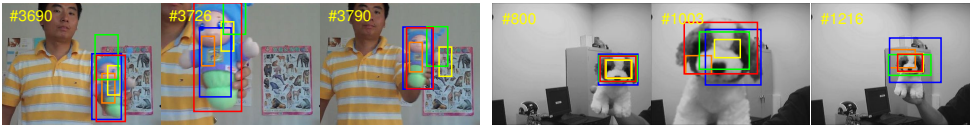
Table 1 shows the results of our scale estimation methods discussed in section 3. We use the HOG based tracker with no scale estimation capability as a baseline. Augmenting the baseline with the simple exhaustive scale estimation method (section 3.2) improves the tracking performance. The method improves the baseline by 14.4% in median OP. Similarly, the exhaustive scale search method provides a median DP 87.6% compared to 74.5% obtained using the baseline approach. However, this performance gain is achieved at the cost of a higher computational load. Our fast scale approach (section 3.3) further improves the tracking performance by 23.3% in median OP, while being 24 times faster compared to the exhaustive scale search method.

In summary, a significant gain of 37.7% median OP is obtained using our robust scale estimation approach compared to the baseline with no scale estimation. This clearly demonstrates the effectiveness of incorporating a robust scale estimation into a visual tracker. It is worthy to mention that our robust scale estimation approach is generic and can be incorporated into any tracking framework.

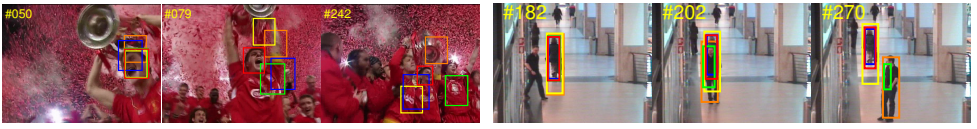
4.5 Comparison with State-of-the-Art

We compare our approach with 11 state-of-the-art trackers: CT [19], TLD [15], DFT [17], EDFT [6], ASLA [14], L1APG [1], CSK [11], SCM [20], LOT [16], Struck [9] and LSHT [10], which have shown to provide excellent performance in literature. The comparison on the 28 benchmark sequences is shown in table 2. We present the results using median OP, DP and CLE over all sequences. Moreover, a speed comparison in median FPS is also provided.

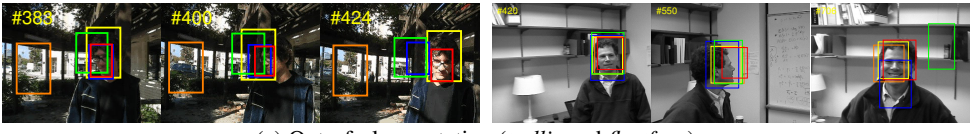
Among the existing trackers, Struck provides the best results with a median CLE of 14.3 pixels. Our approach improves the performance with a reduction in median CLE by 3.4 pixels. Similarly, Struck and SCM provide a median DP of 76.8% and 64.3% respectively. Our approach significantly improves the tracking performance by achieving a median DP of 93.3%. Finally in overlap precision (OP), ASLA provides the best results among the existing methods with a median OP of 69.9%. Our approach, despite its simplicity, outperforms ASLA by 5.6% in median OP. It is worthy to mention that our approach is significantly



(a) Scale variation (*doll* and *dog*).



(b) Partial occlusion (*soccer* and *walking*).



(c) Out-of-plane rotation (*trellis* and *fleetface*).

Ours ASLA SCM Struck LSHT

Figure 3: A visualization of the tracking results of our approach and the state-of-the-art visual trackers ASLA [14], SCM [20], Struck [9] and LSHT [10] on six benchmark sequences. The image sequences pose challenging situations such as scale variations (a), partial occlusions (b) and out-of-plane rotations (c).

faster than the best performing compared trackers. Our method is more than 2.5 times faster than Struck, 25 times faster than ASLA and 250 times faster than SCM in median FPS.

Table 3 provides a per-video comparison with the top 5 existing trackers in our evaluation. The per-video results are presented using overlap precision (OP). Our approach provides better or similar performance on 16 out of the 28 sequences.

Finally, figure 2 contains the precision and success plots illustrating the *mean* distance and overlap precision over all the 28 sequences. In both precision and success plots, our approach significantly outperforms the best existing method (Struck and ASLA). In summary, the precision plot demonstrates that our approach is superior in robustness compared to existing trackers. Similarly, the success plot shows that our method tracks scale more accurately on the benchmark sequences.

4.6 Qualitative evaluation

Here we provide a qualitative comparison of our approach with existing trackers. Figure 3a illustrates two sequences with significant scale variations, namely *doll* and *dog*. Both ASLA and SCM are capable of tracking scale changes, but suffer from significant scale drift in the presence of rotating motions and fast scale variations. Despite these challenges, our tracker accurately estimates both the scale and position of the target.

Figure 3b illustrates the results on two sequences with partial occlusions. The compared trackers fail to handle the significant clutter and occlusions in the *soccer* sequence. However, our tracker accurately estimates the scale and position, as shown in frame 242 in figure 3b and in figure 4. Similarly, our method manages to track the target in the *walking* sequence despite the partial occlusion by a similar object at frame 202, where ASLA and LSHT fail.

Figure 3c shows the results on two sequences, *trellis* and *fleetface*, with significant illumination variation and out-of-plane rotation. Both SCM and LSHT fail due to the varying

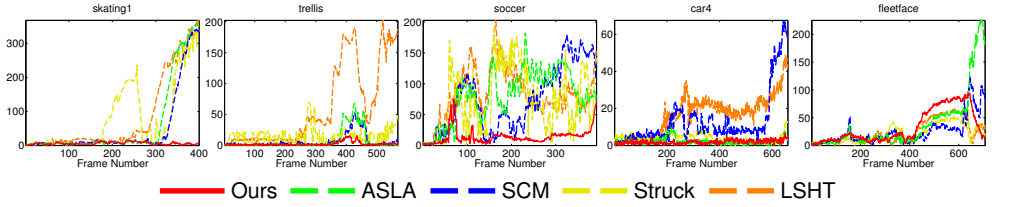


Figure 4: A frame-by-frame comparison of the centre location error (in pixels) on the five challenging sequences *skating1*, *trellis*, *soccer*, *car4* and *fleetFace*. Our approach provides promising results compared to existing trackers on these sequences.

lighting conditions encountered in the *trellis* sequence. Our approach provides better robustness to these conditions (figure 4). On the *fleetface* sequence, our approach drifts on frame 550 in figure 3c due to out-of-plane rotation, but manages to accurately reacquire the target at frame 706 (figure 4).

The proposed tracker struggles in the *freeman3* and *freeman4* sequences, where the target is initially very small (around 15×15 pixels) and then slowly increases in size. Our method does not manage to identify the increasing scale in these conditions. This is most likely due to the HOG-based feature representation, that performs poorly at low resolutions.

5 Conclusions

In this paper, we propose an accurate scale estimation approach for visual tracking. Our method learns discriminative correlation filters for estimating translation and scale independently. Compared to an exhaustive scale space search scheme, our tracker provides improved performance while being computationally efficient. Since our scale estimation approach is independent, it can be incorporated in any tracking method lacking this component.

Experiments are performed on 28 challenging benchmark sequences with significant scale variation. Both quantitative and qualitative evaluations are performed to validate our approach. The results clearly demonstrate that our approach outperforms state-of-the-art methods, while operating at real-time.

Acknowledgments: This work has been supported by SSF through a grants for the project CUAS, VR through a grants for the project ETT, through the Strategic Area for ICT research ELLIIT and the Linnaeus research environment CADICS.

References

[1] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *CVPR*, 2012.

[2] Vishnu Naresh Boddeti, Takeo Kanade, and B. V. K. Vijaya Kumar. Correlation filters for object alignment. In *CVPR*. IEEE, 2013.

[3] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Yui M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.

[4] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost van de Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014.

[5] Piotr Dollár. Piotr's Image and Video Matlab Toolbox (PMT). http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.

[6] Michael Felsberg. Enhanced distribution field tracking using channel representations. In *ICCV Workshop*, 2013.

[7] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[8] H.K. Galoogahi, T. Sim, and S. Lucey. Multi-channel correlation filters. In *ICCV*, 2013.

[9] Sam Hare, Amir Saffari, and Philip Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[10] Shengfeng He, Qingxiong Yang, Rynson Lau, Jiang Wang, and Ming-Hsuan Yang. Visual tracking via locality sensitive histograms. In *CVPR*, 2013.

[11] Joao Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.

[12] Joao F. Henriques, Joao Carreira, Rui Caseiro, and Jorge Batista. Beyond hard negative mining: Efficient detector learning via block-circulant decomposition. In *ICCV*, 2013.

[13] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *CoRR*, abs/1404.7584, 2014.

[14] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, 2012.

[15] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, 2010.

[16] Shaul Oron, Aharon Bar-Hillel, Dan Levi, and Shai Avidan. Locally orderless tracking. In *CVPR*, 2012.

[17] Laura Sevilla-Lara and Erik G. Learned-Miller. Distribution fields for tracking. In *CVPR*, 2012.

[18] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[19] K. Zhang, L. Zhang, and M.H. Yang. Real-time compressive tracking. In *ECCV*, 2012.

[20] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, 2012.