Robust estimation of similarity transformation for visual object tracking

Yang LI

Jianke ZHU

Steven C. H. HOI
Singapore Management University, chhoi@smu.edu.sg

Wenjie SONG

Zhefeng WANG

See next page for additional authors

Citation

LI, Yang; ZHU, Jianke; HOI, Steven C. H.; SONG, Wenjie; WANG, Zhefeng; and LIU, Hantang. Robust estimation of similarity transformation for visual object tracking. (2019). *Proceedings of the 33rd AAAI Conference on Artificial Intelligence 2019: Honolulu, January 27 - February 1*. 8666-8673.
Available at: https://ink.library.smu.edu.sg/sis_research/5105

Author

Yang LI, Jianke ZHU, Steven C. H. HOI, Wenjie SONG, Zhefeng WANG, and Hantang LIU

Robust Estimation of Similarity Transformation for Visual Object Tracking

Yang Li,[1] **Jianke Zhu,**[*,1,3] **Steven C.H. Hoi,**[2] **Wenjie Song,**[1] **Zhefeng Wang,**[1] **Hantang Liu**[1]

[1]Colleage of Computer Science and Technology, Zhejiang University, Hangzhou, China
[2]School of Information Systems, Singapore Management University, Singapore
[3]Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies
{liyang89,jkzhu,seeyou,wangzhefeng,liuhantang}@zju.edu.cn, chhoi@smu.edu.sg

Abstract

Most of existing correlation filter-based tracking approaches only estimate simple axis-aligned bounding boxes, and very few of them is capable of recovering the underlying similarity transformation. To tackle this challenging problem, in this paper, we propose a new correlation filter-based tracker with a novel robust estimation of similarity transformation on the large displacements. In order to efficiently search in such a large 4-DoF space in real-time, we formulate the problem into two 2-DoF sub-problems and apply an efficient Block Coordinates Descent solver to optimize the estimation result. Specifically, we employ an efficient phase correlation scheme to deal with both scale and rotation changes simultaneously in log-polar coordinates. Moreover, a variant of correlation filter is used to predict the translational motion individually. Our experimental results demonstrate that the proposed tracker achieves very promising prediction performance compared with the state-of-the-art visual object tracking methods while still retaining the advantages of high efficiency and simplicity in conventional correlation filter-based tracking methods.

Introduction

Visual object tracking is one of the fundamental problems in computer vision with a variety of real-world applications, such as video surveillance and robotics. Although having achieved substantial progress during past decade, it is still difficult to deal with the challenging unconstraint environmental variations, such as illumination changes, partial occlusions, motion blur, fast motion and scale variations.

Recently, correlation filter-based methods have attracted continuous research attention (Mueller, Smith, and Ghanem 2017; Ma et al. 2015a; 2015b; Zhang et al. 2017; Li, Xu, and Zhu 2017) due to its superior performance and robustness in contrast to traditional tracking approaches. However, with correlation filters, little attention has been paid on how to efficiently and precisely estimate scale and rotation changes, which are typically represented in a 4-Degree of Freedom (DoF) similarity transformation. To deal with scale changes of the conventional correlation filter-based trackers, (Danelljan et al. 2017b) and (Li and Zhu 2014) extended the 2-DoF representation of original correlation filter-based methods to 3-DoF space, which handles scale changes in object
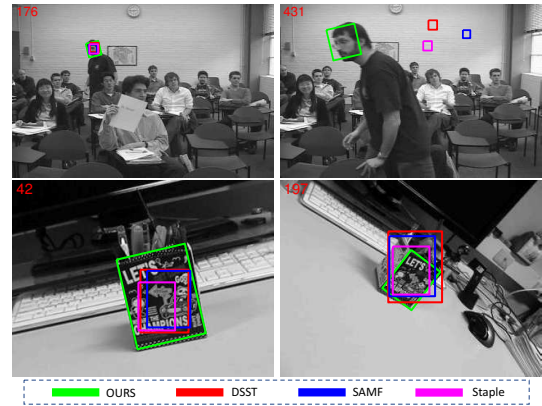

Figure 1: The similarity geometric transformation representation achieves more accurate and robust tracking results.

appearance by introducing a pyramid-like scale sampling ensemble. Unfortunately, all these methods have to intensively resample the image in order to estimate the geometric transformation, which incurs huge amounts of computational costs. In addition, their accuracy is limited to the pre-defined dense sampling of the scale pool. This makes them unable to handle the large displacement that is out of the pre-defined range in the status space. Thus, none of these methods is guaranteed to the optimum of the scale estimation. On the other hand, rotation estimation for the correlation filter-based methods has not been fully exploited yet, since it is very easy to drift away from the inaccurate rotation predictions. This greatly limits their scope of applications in various wide situations. Table 1 summarizes the properties of several typical trackers.

To address the above limitations, in this paper, we propose a novel visual object tracker to estimate the similarity transformation of the target efficiently and robustly. Unlike existing correlation filter-based trackers, we formulate the visual object tracking into a status space searching problem in a 4-DoF status space, which gives a more appropriate geometric transformation parameterization for the target. As shown in Fig. 1, the representation in similarity transformation describes the object more correctly and helps to track the visual object more accurately. To yield real-time tracking

---

*Corresponding author.

Table 1: Comparison with different kinds of trackers.

Type	Trackers	Sample Num.	Scale	Rot.	Pretrain	Performance	GPU	FPS
Traditional Methods	Lucas-Kanade based (Baker and Matthews 2004)	Depends	√	√	×	Fair	×	Depends
	Keypoint based (Nebehay and Pflugfelder. 2014)	Depends	√	√	×	Fair	×	1~20
	Particle filter-based (Ross et al. 2008; Ji 2012)	300~600	√	√	×	Fair	×	1~20
Deep Learning	MDNet (Nam and Han 2016)	250	√	×	√	Excellent	√	~1
	SiamFC (Bertinetto et al. 2016)	5	√	×	√	Excellent	√	15~25
Correlation Filter	original CF (Henriques et al. 2015; Bolme et al. 2010)	1	×	×	×	Fair	×	300+
	DSST (Danelljan et al. 2017b),SAMF (Li and Zhu 2014)	7~33	√	×	×	Good	×	20~80
	Ours	2~8	√	√	×	Excellent	×	20~30

performance in the 4-DoF space, we propose to tackle the optimization task of estimating the similarity transformation by applying an efficient Block Coordinates Descent (BCD) solver. Specifically, we employ an efficient phase correlation scheme to deal with both scale and rotation changes simultaneously in log-polar coordinates and utilize a fast variant of correlation filter to predict the translational motion. This scheme sets our approach free from intensive sampling, and greatly boosts the performance in the 4-DoF space. More importantly, as BCD searches the entire similarity transformation space, the proposed tracker achieves very accurate prediction performance in large displacement motion while still retaining advantages of the efficiency and simplicity in conventional correlation filter. Experimental results demonstrate that our approach is robust and accurate for both generic object and planar object tracking.

The main contributions of our work are summarized as follows: 1) a novel framework of similarity transformation estimation which only samples once for correlation filter-based trackers; 2) a joint optimization to ensure the stability in translation and scale-rotation estimation; 3) a new approach for scale and rotation estimation with efficient implementation which can improve a family of existing correlation filter-based trackers (our implementation is available at https://github.com/ihpdep/LDES).

Related Work

Traditionally, there are three genres to handle scale and rotation changes. The most widely used approach is to iteratively search in an affine status space with gradient descent-based method (Baker and Matthews 2004; Song et al. 2016). However, they are easy to get stuck at local optima, which are not robust for large displacements. Trackers based on particle filter (Ross et al. 2008; Ji 2012; Zhang, Xu, and Yang 2017; Li, Zhu, and Hoi 2015) search the status space stochastically by observing the samples, which are employed to estimate the global optima in the status space. Their results are highly related to the motion model that controls the distribution of the 6-DoF transformation. This makes the tracker perform inconsistently in different situations. Another choice is to take advantage of keypoint matching to predict the geometric transformation (Nebehay and Pflugfelder. 2014; Zuysal et al. 2010). These keypoint-based trackers first detect feature points, and then find the matched points in the following frames. Naturally, they can handle any kind of transformations with the matched feature points. Due to the lack of global information on the whole target, these track-ers cannot effectively handle the general objects (Kristan et al. 2015).

Our proposed method is highly related to correlation filter-based trackers (Henriques et al. 2015; Bolme et al. 2010). (Danelljan et al. 2017b) and (Li and Zhu 2014) extend the original correlation filter to adapt the scale changes in the sequences. (Bertinetto et al. 2015) combines color information with correlation filter method in order to build a robust and efficient tracker. Later, (Danelljan et al. 2015) and (Galoogahi, Fagg, and Lucey 2017) decouple the relationship between the size of filter and searching range. These approaches enable the correlation filter-based methods to have larger searching range while maintaining a relative compact presentation of the learned filters. (Mueller, Smith, and Ghanem 2017) learns the filter with the additional negative samples to enhance the robustness. Note that all these approaches emphasize on the efficacy issue, which employs either DSST or SAMF to deal with the scale changes. However,these methods cannot deal with rotation changes.

Fourier Mellin image registration and its variants (Ravichandran and Casasent 1994; Zokai and Wolberg 2005) are also highly related to our proposed approach. These methods usually convert both the test image and template into log-polar coordinates, in which the relative scale and rotation changes turn into the translational displacement. (Ravichandran and Casasent 1994) propose a rotation-invariant correlation filter to detect the same object from a god view. (Zokai and Wolberg 2005) propose an image registration method to recover large-scale similarity in spatial domain. Recently, (Li and Liu 2016) and (Zhang et al. 2015) introduce the log-polar coordinates into correlation filter-based method to estimate the rotation and scale. Compared with their approaches, we directly employ phase correlation operation in log-polar coordinates. Moreover, an efficient Block Coordinates Descent optimization scheme is proposed to deal with large motions with real-time performance.

Our Approach

In this paper, we aim to investigate robust visual object tracking techniques to deal with challenging scenarios especially when there are large displacements. We propose a novel robust object tracking approach, named "Large-Displacement tracking vis Estimation of Similarity (LDES), where the key idea is to enable the tracker with capability in 2D similarity transformation estimation in order to handle large displacement. Figure 2 gives an overview of the pro-
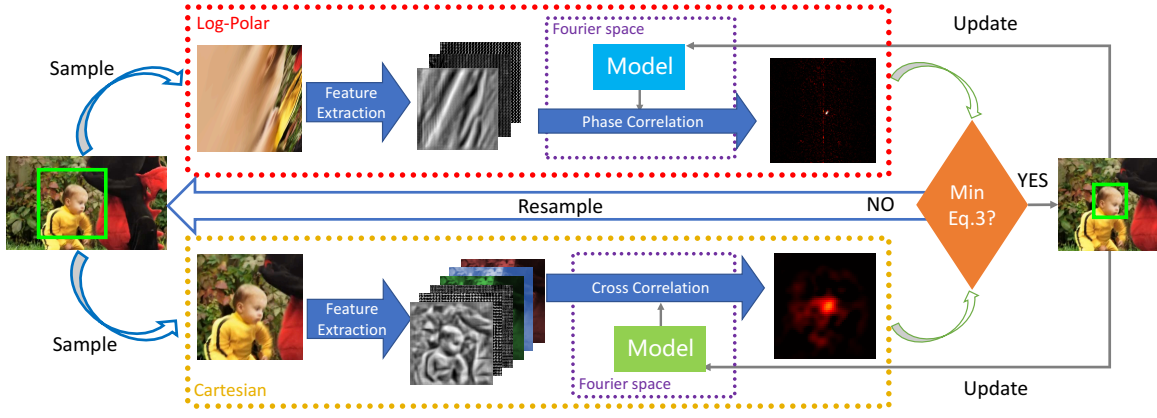
Figure 2: Overview of our proposed approach in estimation of similarity transformation.

posed LDES approach. In the following, we first formally formulate the problem as an optimization task, and then divide it into two sub-problems, translation estimation and scale-rotation prediction. We solve the two sub-problems iteratively to achieve a global optimal.

Problem Formulation

Given an image patch $\mathbf{x}_i$ sampled from the i-th frame I_i in a video sequence, the key idea of our proposed approach is to estimate the similarity transformation $Sim(2)$ in 2D image space of the tracked traget. To this end, we need to predict a 4-DoF transformation status vector $\tau_i \in \mathcal{R}^4$ based on the output of the previous frame. Generally, τ_i is obtained by optimizing the following score function:

$$\tau_i = \arg \max_{\tau \in Sim(2)} f(\mathcal{W}(I_i, \tau); \mathbf{h}_{i-1}), \quad (1)$$

where $f(\cdot)$ is a score function with the model $\mathbf{h}_{i-1}$ learned from the previous frames $I_{1:i-1}$. $\mathcal{W}$ is an image warping function that samples the image I_i with respect to the similarity transformation status vector τ.

The 2D similarity transformation $Sim(2)$ deals with 4-DoF $\{t_x, t_y, \theta, s\}$ motion, where $\{t_x, t_y\}$ denotes the 2D translation. θ denotes the in-plane rotation angle, and s represents the scale change with respect to the template. Obviously, $Sim(2)$ has a quite large searching space, which is especially challenging for real-time applications. A typical remedy is to make use of effective sampling techniques to greatly reduce the searching space (Doucet, de Freitas, and Gordon 2001).

Since the tracking model $\mathbf{h}_{i-1}$ is learned from the previous frame, which is kept constant during the prediction. The score function f is only related to the status vector τ. We abuse the notation for simplicity:

$$f_i(\tau) = f(\mathcal{W}(I_i, \tau); \mathbf{h}_{i-1}). \quad (2)$$

Typically, most of the conventional correlation filter-based methods only take into account of in-plane translation with 2-DoF, where the score function f_i can be calculated completely and efficiently by taking advantage of Convolution Theorem. To search the 4-DoF similarity space, the total number of candidate status exponentially increases.

Although Eq. 1 is usually non-convex, the optimal translation is near to the one in the previous frame in object tracking scenarios. Thus, we assume that the function is convex and smooth in the nearby region, and split the similarity transformation $Sim(2)$ into two blocks, $\mathbf{t} = \{t_x, t_y\}$ and $\rho = \{\theta, s\}$, respectively. We propose a score function $f_i(\tau)$, which is the linear combination of three separate parts:

$$f_i(\tau; \mathbf{h}_{i-1}) = \eta f_t(\mathbf{t}; \mathbf{h_t}) + (1-\eta) f_\rho(\rho; \mathbf{h}_\rho) + g(\mathbf{t}, \rho), \quad (3)$$

where η is an interpolation coefficient. f_t is the translational score function, and f_ρ denotes the scale and rotation score function. $g(\mathbf{t}, \rho) = exp(|\tau - \tau_{i-1}|_2)^{-1}$ is the motion model which prefers the location nearby the last status. Please note that we omit the subscript $i-1$ of $\mathbf{h_t}$ and $\mathbf{h}_\rho$ for simplicity.

Eq. 3 is a canonical form which can be solved by the Block Coordinate Descent Methods (Richtárik and Takáč 2014; Nesterov 2010). We optimize the following two subproblems alternatively to achieve the global solution:

$$\arg \max_{\mathbf{t}} g(\mathbf{t}, \rho^*) + \eta f_t(\mathbf{t}), \quad (4)$$

$$\arg \max_{\rho} g(\mathbf{t}^*, \rho) + (1-\eta) f_\rho(\rho), \quad (5)$$

ρ^* and $\mathbf{t}^*$ denote the local optimal estimation result from previous iteration, which is fixed for the current subproblem. Since g can be calculated easily, the key to solving Eq. 1 in real-time is to find the efficient solvers for the above two subproblems, f_ρ and f_t.

Translation Estimation by Correlation Filter

Translation vector $\mathbf{t}$ can be effectively estimated by Discriminative Correlation Filters (DCF) (Henriques et al. 2015; Mueller, Smith, and Ghanem 2017). A large part of its success is mainly due to the Fourier trick and translation-equivariance within a certain range, which calculates the f_t in the spatial space exactly. According to the property of DCF, the following equation can be obtained:

$$f_t(\mathcal{W}(I, \mathbf{t}); \mathbf{h_t}) = \mathcal{W}(f_t(I; \mathbf{h_t}), \mathbf{t}). \quad (6)$$

Since the calculation of $\arg \max_{\mathbf{t}} \mathcal{W}(f_t(I; \mathbf{h_t}), \mathbf{t})$ is unrelated to $\mathcal{W}$, we can directly obtain the transformation vector

t from the response map. Thus, the overall process is highly efficient. The score function f_t can be obtained by

$$f_t(\mathbf{z}) = \mathcal{F}^{-1} \sum_k \hat{\mathbf{h_t}}^{(k)} \odot \hat{\Phi}^{(k)}(\mathbf{z}), \qquad (7)$$

where **z** indicates a large testing patch. $\mathcal{F}^{-1}$ denotes the inverse Discrete Fourier Transformation operator, $\odot$ is the element-wise multiplication and $\hat{\cdot}$ indicates the Fourier space. $\mathbf{h_t}^{(k)}$ and $\Phi^{(k)}$ represent the k-th channel of the linear model weights and the feature map, respectively. The whole computational cost is $\mathcal{O}(KN \log N)$, where K is the channel number and N is the dimension of the patch **z**.

To this end, we need to learn a model $\mathbf{h_t}$ in the process. Note that any quick learning method can be used. Without loss of generality, we briefly review a simple correlation filter learning approach (Bolme et al. 2010) as follows:

$$\left\| \sum_k \Phi^{(k)}(\mathbf{x}) \star \mathbf{h_t}^{(k)} - \mathbf{y} \right\|_2^2 + \lambda_1 \|\mathbf{h_t}\|_2^2, \qquad (8)$$

where $\star$ indicates the correlation operator and λ_1 is the regularization filters. **y** is the desired output, which is typically a Gaussian-like map with maximum value of one. According to Parseval's theorem, the formulation can be calculated without correlation operation. By stalling each channel and vectorizing the matrix, Eq. 8 can be reformulated as a normal ridge regression without correlation operation. Thus, the solution to Eq. 8 can expressed as follows:

$$\hat{\mathbf{h_t}} = (\hat{\mathbf{X}}^T \hat{\mathbf{X}} + \lambda_1 \mathbf{I})^{-1} \hat{\mathbf{X}}^T \hat{\mathbf{y}}, \qquad (9)$$

where $\hat{\mathbf{X}} = [\mathbf{diag}(\hat{\Phi}^{(1)}(\mathbf{x}))^T, ..., \mathbf{diag}(\hat{\Phi}^{(K)}(\mathbf{x}))^T]$ and $\hat{\mathbf{h_t}} = [\hat{\mathbf{h_t}}^{(1)T}, ..., \hat{\mathbf{h_t}}^{(K)T}]^T$. In this form, we need to solve a $KD \times KD$ linear system, where D is the dimension of testing patch **x**.

To solve our sub-problem efficiently, we assume that every channel is independent. Thus, by applying Parseval's theorem, the whole system can be simplified as element-wise operation. The final solution can be derived as below:

$$\begin{aligned} \hat{\mathbf{h_t}}^{(k)} &= \hat{\alpha} \odot \hat{\Psi}^{(k)} \\ &= (\hat{\mathbf{y}} \odot^{-1} (\sum_k \mathbf{V}_k^* \odot \mathbf{V}_k + \lambda)) \odot \mathbf{V}_k^*, \end{aligned} \qquad (10)$$

where α denotes the parameters in dual space and Ψ indicates the model sample in feature space. $\mathbf{V}_k$ is the feature data, $\hat{\Phi}^{(k)}(\mathbf{x})$ and $\odot^{-1}$ is the element-wise division. Thus, the solution can be very efficiently obtained with a computational cost of $\mathcal{O}(KD)$. With Eq. 10, the computational cost of Eq. 8 is $\mathcal{O}(KD \log D)$ which is dominated by the FFT operation. For more details, please refer to the seminal work (Henriques et al. 2015; Kiani, Sim, and Lucey 2013; Mueller, Smith, and Ghanem 2017).

Scale and Rotation in Log-polar Coordinates

We introduce an efficient method to estimate scale and rotation changes simultaneously in the log-polar coordinates.

Log-Polar Coordinates Suppose an image $I(x, y)$ in the spatial domain, the log-polar coordinates $I'(s, \theta)$ can be viewed as a non-linear and non-uniform transformation of the original Cartesian coordinates. Like polar coordinates, the log-polar coordinates needs a pivot point as the pole and a reference direction as the polar axis in order to expend the coordinates system. One of the dimension is the angle between the point and the polar axis. The other is the logarithm of the distance between the point and the pole.

Given the pivot point (x_0, y_0) and the reference direction **r** in Cartesian coordinates, the relationship between Cartesian coordinates and Log-polar coordinates can be formally expressed as follows:

$$\begin{aligned} s &= \log(\sqrt{(x - x_0)^2 + (y - y_0)^2}) \\ \theta &= \cos^{-1}(\frac{< \mathbf{r}, (x - x_0, y - y_0) >}{\|\mathbf{r}\| \sqrt{(x - x_0)^2 + (y - y_0)^2}}). \end{aligned} \qquad (11)$$

Usually, the polar axis is chosen as the x-axis in Cartesian coordinates, where θ can be simplified as $\tan^{-1}(\frac{y - y_0}{x - x_0})$. Suppose two images are related purely by rotation $\tilde{\theta}$ and scale $e^{\tilde{s}}$ which can be written as $I_0(e^s \cos\theta, e^s \sin\theta) = I_1(e^{s+\tilde{s}} \cos(\theta+\tilde{\theta}), e^{s+\tilde{s}} \sin(\theta+\tilde{\theta}))$ in Cartesian coordinates. The log-polar coordinates enjoy an appealing merit that the relationship in the above equation can be derived as the following formula in log-polar coordinates:

$$I_0'(s, \theta) = I_1'(s + \tilde{s}, \theta + \tilde{\theta}), \qquad (12)$$

where the pure rotation and scale changes in Log-polar coordinates can be viewed as the translational moving along the axis. As illustrated in Fig. 3, this property naturally can be employed to estimate the scale and rotation changes of the tracked target.

Scale and Rotation Changes By taking advantage of the log-polar coordinates, Eq. 5 can be calculated very efficiently. Similarly, scale-rotation invariant can be hold as in Eq. 6. The scale-rotation can be calculated as below:

$$f_\rho(\mathcal{W}(I_i, \rho); \mathbf{h}_\rho) = \mathcal{W}(f_\rho(I_i; \mathbf{h}_\rho), \rho'), \qquad (13)$$

where $\rho' = \{\theta', s'\}$ is the coordinates of ρ in log-polar space. $s = e^{s' \log(W/2)/W}$ and $\theta = 2\pi\theta'/H$. H and W is the height and width of the image I_i, respectively. Similar to estimating the translation vector **t** by f_t, the whole space of f_ρ can be computed at once through the Fourier trick:

$$f_\rho(\mathbf{z}) = \mathcal{F}^{-1} \sum_k \hat{\mathbf{h}}_\rho^{(k)} \odot \hat{\Phi}^{(k)}(\mathcal{L}(\mathbf{z})), \qquad (14)$$

where $\mathcal{L}(x)$ is the log-polar transformation function, and $\mathbf{h}_\rho$ is a linear model weights for scale and rotation estimation. Therefore, the scale and rotation estimation can be obtained very efficiently without any transformation sampling $\mathcal{W}$. Note that the computational cost of Eq. 14 is unrelated to the sample numbers of scale or rotation. This is extremely efficient compared to the previous enumerate methods (Li and Zhu 2014; Danelljan et al. 2017b)

To obtain the $\hat{\mathbf{h}}_\rho$ efficiently, we employ the phase-correlation to conduct the estimation,

$$\hat{\mathbf{h}}_\rho = \hat{\Upsilon}^* \odot^{-1} |\hat{\Upsilon} \odot \hat{\Phi}(\mathcal{L}(\mathbf{x}))|, \qquad (15)$$
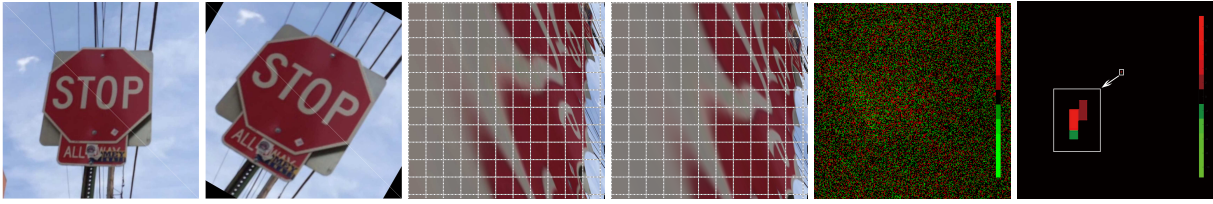
Figure 3: The 3rd and 4th charts are corresponding Log-polar coordinates of the 1st and 2nd images. 2nd image is a $30°$ rotation and 1.2 times scale version of the first image. The last two charts are the phase correlation response maps. In log-polar coordinates, the response is a peak while it is noisy in Cartesian coordinates.

where $\Upsilon = \sum_j \beta_j \Phi(\mathcal{L}(\mathbf{x}_j))$ is the linear combination of previous feature patch and $|\cdot|$ is the normal operation. Intuitively, we compute the phase correlation between current frame and the average of previous frames to align the image.

Implementation Details

In this work, we alternatively optimize Eq. 4 and Eq. 5 until $f(\mathbf{x})$ does not decrease or reaches the maximal number of iterations. After the optimization, we update the correlation filter model as

$$\hat{\Psi}_i = (1 - \lambda_\phi)\hat{\Psi}_{i-1} + \lambda_\phi\hat{\Phi}(\mathbf{x_i}), \qquad (16)$$

where λ_ϕ is the update rate of the feature data model in Eq. 10. The kernel weight in dual space is updated as below:

$$\begin{aligned}\hat{\alpha}_i =&(1 - \lambda_\alpha)\hat{\alpha}_{i-1} \\ &+\lambda_\alpha(\hat{\mathbf{y}}\odot^{-1}(\sum_k \hat{\Phi}^{(k)}(\mathbf{x}_i)^* \odot \hat{\Phi}^{(k)}(\mathbf{x}_i) + \lambda_1)),\end{aligned} \quad (17)$$

where λ_α is the update of the kernel parameter in dual space of Eq. 10. Although there exist some theoretical sounding updating schemes (Kiani, Sim, and Lucey 2015; Danelljan et al. 2017a; 2015), the reason we use linear combination is due to its efficiency and the comparable performance.

Meanwhile, we also update the scale and rotation model as a linear combination,

$$\Upsilon_i = (1 - \lambda_w)\Upsilon_{i-1} + \lambda_w\Phi(\mathcal{L}(\mathbf{x}_i)), \qquad (18)$$

where λ_w can be explained as an exponentially weighted average of the model $\beta_j\Phi(\mathcal{L}(\mathbf{x}_j))$. We update the model upon Φ instead of x_i because $\Phi(\sum_i \mathcal{L}(\mathbf{x}_i))$ is not defined. The logarithm function in log-polar transformation intends to blur the image due to the nonuniform sampling. This will decrease the visual information in the original images. To alleviate the artificial effects casued by discretization, we interpolate the f_t and f_ρ with a centroid-based method to obtain sub-pixel level precision. In addition, we use different size of $\mathbf{z}$ in testing and $\mathbf{x}$ in training since a larger search range ($N > D$) help to improve the robustness for the solution to sub-problems. To match the different dimension N and D, we pad $\mathbf{h}$ with zero in spatial space.

Experiments

In this section, we conduct four different experiments to evaluate our proposed tracker LDES comprehensively.

Experimental Settings

All the methods were implemented in Matlab and the experiments were conducted on a PC with an Intel i7-4770 3.40GHz CPU and 16GB RAM. We employ HoG feature for both translational and scale-rotation estimation, and the extra color histogram is used to estimate translational. All patch is multiplied a Hann window as suggested in (Bolme et al. 2010). η is 0.15 and λ is set to $1e^{-4}$. λ_ϕ and λ_α are both set to 0.01. λ_ω is 0.015. The size of learning patch D is 2.2 larger than the original target size. Moreover, the searching window size N is about 1.5 larger than the learning patch size D. For scale-rotation estimation, the phase correlation sample size is about 1.8 larger than the original target size. All parameters are fixed in the following experiments.

Experiments on Proposed Scale Estimator

As one of the contributions in our work is a fast scale estimator, we first evaluate our proposed log-polar based scale estimator on OTB-2013 and OTB-100 dataset (Wu, Lim, and Yang 2013; 2015).Three baseline trackers are involved in the scale estimation evaluation. They are SAMF (Li and Zhu 2014), fDSST (Danelljan et al. 2017b) and ECO (Danelljan et al. 2017a). For fair comparison, we implement three counterpart-trackers including fDSST-LP, SAMF-LP and ECO-LP, which replace the original scale algorithm with our proposed scale estimator.

In Fig. 5, these variant trackers with our scale component outperform their original implementation. This indicates that our proposed scale estimator has superior performance compared with current state-of-the-art scale estimator. Specifically, ECO-LP achieves 69.1% and 67.3% in OTB-2013 and OTB-2015 respectively, compared with its original CPU implementation's 67.8% and 66.8%. This proves the effectiveness of our proposed scale method since it can even improve the state-of-the-art tracker with a simple replacement of the scale component.

Since the proposed scale estimator only samples once in each frame, the most significant part is the efficiency of scale estimating. In Table 2, the proposed approach has a 3.8X+ speedup on SAMF, and ECO, which obtains a significant improvement on efficiency. Even with fDSST which is designed in efficiency with many tricks, our method can still reduce its computational time. This strongly supports that our proposed scale estimator is superior to current state-of-the-art scale estimating approaches. In addition, our method is very easy to implement and plug-in to other trackers.

(a) Overall results on OTB-2013   (b) Overall results on OTB-100
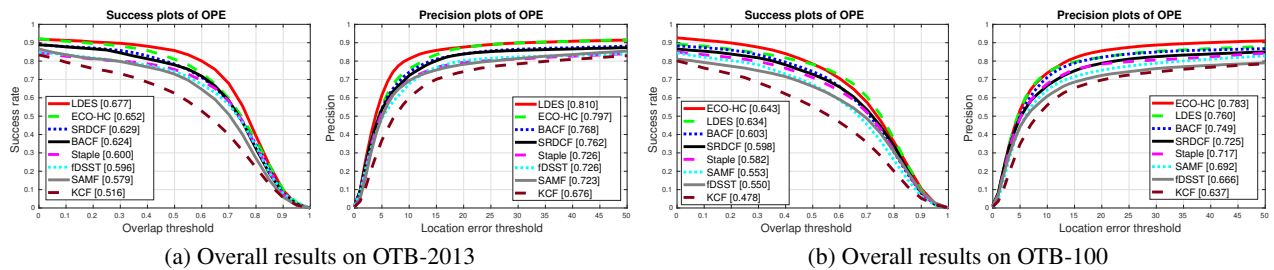
Figure 4: The overall and scale performance in precision and success plots on OTB-2013 and OTB-100 dataset.
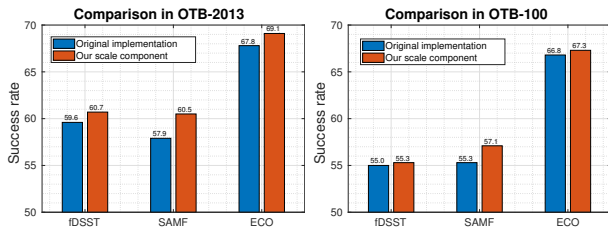

Figure 5: Evaluation of tracking success rate improvements over the original implementations of different trackers enhanced by the proposed scale estimator.

Comparison with Correlation Filter Trackers

With efficient and effective scale estimator, our proposed tracker performs very promising in different situations. We select seven state-of-the-art Correlation Filter-based trackers as reference methods, including ECO-HC (Danelljan et al. 2017a), SRDCF (Danelljan et al. 2015), Staple (Bertinetto et al. 2015), SAMF, fDSST, BACF (Galoogahi, Fagg, and Lucey 2017), and KCF (Henriques et al. 2015). We initialize the proposed tracker with axis-aligned bounding box and ignore the rotation parameter in the similarity transformation as tracking output since the benchmarks only provide axis-aligned labels.

In Fig. 4, it can be clearly seen that our proposed method outperforms most of the state-of-the-art correlation filter-based trackers and obtains 67.7% and 81.0% in OTB-2013 success and precision plots, and 63.4% and 76.0% in OTB-100 plots respectively. ECO-HC achieves better results in OTB-100. However, we can see that our method is more accurate above 0.6 overlap threshold in success plot and comparable in the precision plot. The reason is that introducing rotation improves the accuracy but also enlarges the search space and hurts the robustness when large deformation occurs. In general, our method is very promising in generic object tracking task.

The proposed approach maintains 20 fps with similarity estimation and is easy to implement due to its simplicity. Moreover, our no-BCD version tracker achieves 82 fps in the benchmark while stall maintains comparable performance (67.5% and 62.2% accuracy in OTB-2013 and OTB-100, respectively).

Please note that our proposed LDES is quite stable in

Table 2: Evaluation of speedup results on different trackers achieved by applying the proposed Log-Polar (LP) based scale estimator.

Trackers	FPS	Trackers	FPS	Speedup
fDSST	101.31	fDSST-LP	112.77	1.11X
SAMF	20.95	SAMF-LP	86.75	4.14X
ECO	2.58	ECO-LP	9.88	3.82X

searching the 4-DoF status space. Introducing rotation gives the tracker more status choice in tracking process while the benchmark only provides axis-aligned labels which make the performance less robust in OTB-100. However, our proposed tracker still ranks 1st and 2nd in OTB-2013 and OTB-100 respectively, and beats most of the other correlation filter-based trackers.

Comparison with State-of-the-Art trackers on POT

To better evaluate our proposed approach to rotation estimation, we conduct an additional experiment on POT benchmark (Liang, Wu, and Ling 2018) which is designed to evaluate the planar transformation tracking methods. The POT dataset contains 30 objects with 7 different categories which yield 210 videos in total. Alignment error and homography discrepancy are employed as the evaluation metrics. In addition, six state-of-the-art trackers and two rotation-enabled trackers are involved. They are ECO-HC, ECO (Danelljan et al. 2017a), MDNet (Nam and Han 2016), BACF (Galoogahi, Fagg, and Lucey 2017), ADNet (Yun et al. 2017), SiameseFC (Bertinetto et al. 2016), IVT (Ross et al. 2008) and L1APG (Ji 2012). To illustrate the POT plots appropriately, we set the maximal value of the alignment error axis from 20 to 50 pixels in precision plot and utilize the AUC as the metrics for ranking in both precision and homography discrepancy plots as same as OTB (Wu, Lim, and Yang 2015).

Fig. 6 shows that our proposed tracker, with hand-craft features only, performs extremely well in all sequences attributed plots and even outperforms deep learning based methods with a large margin. In Fig. 6a, our LDES achieves 50.64% and 51.31% compared with second rank tracker ECO's 35.99% and 37.79% in precision and success rate plots within all 210 sequences, which is a 13%+ performance improvement. Since the POT sequences are quite different from OTB, it indicates our proposed method has better
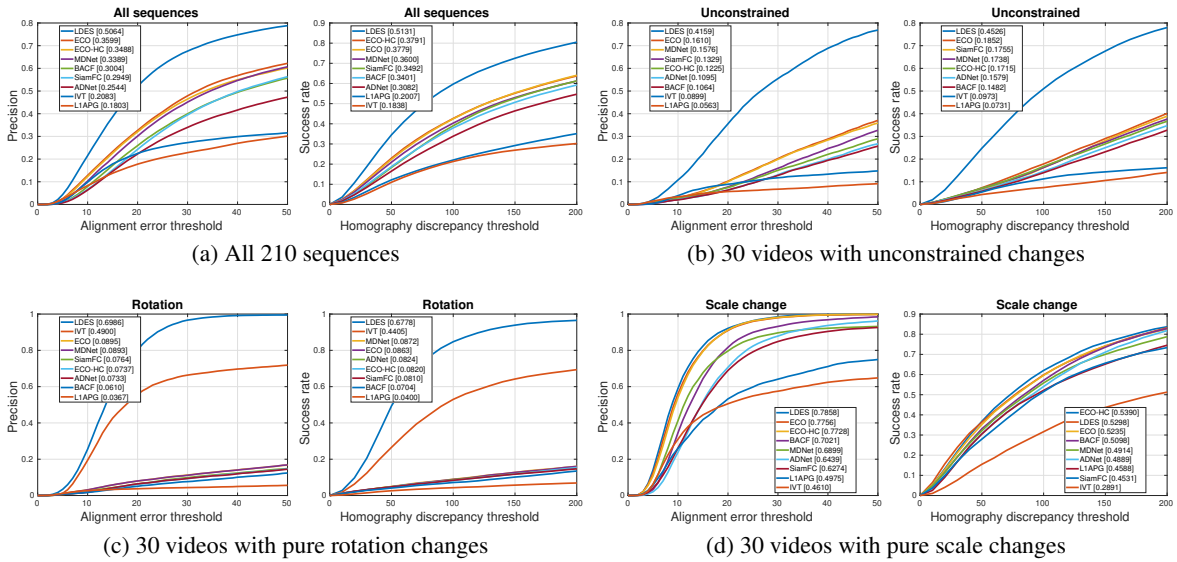
(a) All 210 sequences

(b) 30 videos with unconstrained changes

(c) 30 videos with pure rotation changes

(d) 30 videos with pure scale changes

Figure 6: Precision and success plots on the POT dataset.

Precision	Scale	Rotation	Persp. Dist.	Motion Blur	Occlusions	Out of View	Unconstrained	ALL
LDES	**0.7858**	**0.6986**	**0.2807**	**0.1699**	0.6277	**0.5663**	**0.4159**	**0.5064**
LDES-NoBCD	0.6461	0.6898	0.2528	0.1595	**0.6679**	0.5507	0.3759	0.4775
Success	Scale	Rotation	Persp. Dist.	Motion Blur	Occlusions	Out of View	Unconstrained	ALL
LDES	**0.5298**	0.6778	**0.3339**	**0.44**	0.5947	**0.5629**	**0.4526**	**0.5131**
LDES-NoBCD	0.4724	**0.6849**	0.3215	0.43	**0.6392**	0.5615	0.4353	0.5064

Table 3: Comparison in all 7 categories attributed videos on POT benchmark.

generalization capabilities compared with pure deep learning based approaches in wide scenarios. It also shows that our proposed method is able to search the 4-DoF similarity status simultaneously, efficiently and precisely.

Moreover, our method ranks 1st in almost all other plots. It not only validates the effectiveness of our proposed rotation estimation but also shows the superiority of our method compared with traditional approaches. In Fig. 6d, we argue that our proposed log-polar based scale estimation is at least comparable with mainstream methods in performance.

BCD Framework Evaluation on POT

To verify the proposed framework with Block Coordinate Descent (BCD), we implement an additional variant, named LDES-NoBCD, which turns off the BCD framework and only estimates the object status once in each frame. We conduct comparison experiments on POT benchmark with LDES and LDES-NoBCD.

In Table 3, LDES performs better than its No BCD version in most of the categories. Specifically, BCD contributes more performance in scale attributed videos and unconstrained videos. LDES achieves 0.7858 and 0.5298 in scale compared with LDES-NoBCD's 0.6461 and 0.4724, which is about 14% improvement in precision plot and 5% in success plot, respectively. This indicates that the proposed framework ensures the stable searching in the 4-DoF space.

In rotation column, the ranks in precision and success rate metrics are inconsistent. The reason is that rotation attributed videos contain pure rotation changes. This gives rotation estimation a proper condition to achieve a promising result. The only category that LDES performs inferior is occlusion attributed videos. When the occlusion occurs, BCD framework tries to find the best status of the templated object while the original object is being occluded and cannot be seen properly. This leads the algorithm to an inferior status. In contrast, No-BCD version algorithm does not search an optimal point in the similarity status space.

Conclusion

In this paper, we proposed a novel visual object tracker for robust estimation of similarity transformation with correlation filter. We formulated the 4-DoF searching problem into two 2-DoF sub-problems and applied a Block Coordinates Descent solver to search in such a large 4-DoF space with real-time performance on a standard PC. Specifically, we employed an efficient phase correlation scheme to deal with both scale and rotation changes simultaneously in log-polar coordinates and utilized a fast variant of correlation filter to predict the translational motion. Experimental results demonstrated that the proposed tracker achieves very promising prediction performance compared with the state-of-the-art visual object tracking methods.

Acknowledgments

References

Baker, S., and Matthews, I. 2004. Lucas-kanade 20 years on: A unifying framework. *IJCV* 56(3):221–255.

Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; and Torr, P. 2015. Staple: Complementary learners for real-time tracking. In *CVPR*.

Bertinetto, L.; Valmadre, J.; Henriques, J.; Vedaldi, A.; and Torr, P. H. 2016. Fully-convolutional siamese networks for object tracking. *ECCV Workshop*.

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

Danelljan, M.; Häger, G.; Khan, F.; and Felsberg, M. 2015. Learning spatially regularized correlation filters for visual tracking. In *ICCV*.

Danelljan, M.; Bhat, G.; Khan, F.; and Felsberg, M. 2017a. Eco: Efficient convolution operators for tracking. In *CVPR*.

Danelljan, M.; Häger, G.; Khan, F.; and Felsberg, M. 2017b. Discriminative scale space tracking. *TPAMI*.

Doucet, A.; de Freitas, N.; and Gordon, N. 2001. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

Galoogahi, H. K.; Fagg, A.; and Lucey, S. 2017. Learning background-aware correlation filters for visual tracking. In *ICCV*.

Henriques, J.; Caseiro, R.; Martins, P.; and Batista, J. 2015. High-speed tracking with kernelized correlation filters. In *IEEE TPAMI*, volume 37, 583–596.

Ji, H. 2012. Real time robust l1 tracker using accelerated proximal gradient approach. In *CVPR*, 1830–1837.

Kiani, H.; Sim, T.; and Lucey, S. 2013. Multi-channel correlation filters. In *ICCV*.

Kiani, H.; Sim, T.; and Lucey, S. 2015. Correlation filters with limited boundaries. In *CVPR*.

Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Cehovin, L.; Fernandez, G.; Vojir, T.; Hager, G.; Nebehay, G.; and Pflugfelder, R. 2015. The visual object tracking vot2015 challenge results. In *ICCV Workshops*.

Li, Y., and Liu, G. 2016. Learning a scale-and-rotation correlation filters for robust visual tracking. In *ICIP*.

Li, Y., and Zhu, J. 2014. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV Workshops*.

Li, Y.; Xu, Z.; and Zhu, J. 2017. Cfnn: Correlation filter neural network for visual object tracking. In *IJCAI*.

Li, Y.; Zhu, J.; and Hoi, S. C. 2015. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *CVPR*.

Liang, P.; Wu, Y.; and Ling, H. 2018. Planar object tracking in the wild: A benchmark. In *ICRA*.

Ma, C.; Huang, J.-B.; Yang, X.; and Yang, M.-H. 2015a. Hierarchical convolutional features for visual tracking. In *ICCV*.

Ma, C.; Yang, X.; Zhang, C.; and Yang, M.-H. 2015b. Long-term correlation tracking. In *CVPR*.

Mueller, M.; Smith, N.; and Ghanem, B. 2017. Context-aware correlation filter tracking. In *CVPR*.

Nam, H., and Han, B. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*.

Nebehay, G., and Pflugfelder., R. 2014. Consensus-based matching and tracking of keypoints. In *Winter Conference on Applications of Computer Visio*.

Nesterov, Y. 2010. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization* 22(2):341–362.

Ravichandran, G., and Casasent, D. 1994. Advanced in-plane rotation-invariant correlation filters. *TPAMI* 16(4):415–420.

Richtárik, P., and Takáč, M. 2014. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming* 144(1):1–38.

Ross, D. A.; Lim, J.; Lin, R.-S.; and Yang, M.-H. 2008. Incremental learning for robust visual tracking. *IJCV* 77(1-3):125–141.

Song, W.; Zhu, J.; Li, Y.; and Chen, C. 2016. Image alignment by online robust pca via stochastic gradient descent. *IEEE Transactions on Circuits and Systems for Video Technology* 26(7):1241–1250.

Wu, Y.; Lim, J.; and Yang, M.-H. 2013. Online object tracking: A benchmark. In *CVPR*.

Wu, Y.; Lim, J.; and Yang, M. 2015. Object tracking benchmark. *TPAMI* 37(9):1834–1848.

Yun, S.; Choi, J.; Yoo, Y.; Yun, K.; and Choi, J. Y. 2017. Action-decision networks for visual tracking with deep reinforcement learning. In *CVPR*.

Zhang, M.; Xing, J.; Gao, J.; Shi, X.; Wang, Q.; and Hu, W. 2015. Joint scale-spatial correlation tracking with adaptive rotation estimation. In *ICCV Workshop*.

Zhang, H.; Fu, H.; Zhou, T.; Zhang, X.; Lou, T.; and Chen, H. 2017. Learning dynamic siamese network for visual object tracking. In *Chinese Control Conference (CCC)*.

Zhang, T.; Xu, C.; and Yang, M.-H. 2017. Multi-task correlation particle filter for robust object tracking. In *CVPR*.

Zokai, S., and Wolberg, G. 2005. Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. *TIP* 14(10):1422–1434.

Zuysal, M.; Calonder, M.; Lepetit, V.; and Fua, P. 2010. Fast keypoint recognition using random ferns. *TPAMI* 32(3):448.