Long-term Correlation Tracking

Chao Ma[1,2], Xiaokang Yang[1], Chongyang Zhang[1], and Ming-Hsuan Yang[2]
[1]Shanghai Jiao Tong University [2]University of California at Merced
{chaoma,xkyang,sunny_zhang}@sjtu.edu.cn, mhyang@ucmerced.edu

Abstract

In this paper, we address the problem of long-term visual tracking where the target objects undergo significant appearance variation due to deformation, abrupt motion, heavy occlusion and out-of-view. In this setting, we decompose the task of tracking into translation and scale estimation of objects. We show that the correlation between temporal context considerably improves the accuracy and reliability for translation estimation, and it is effective to learn discriminative correlation filters from the most confident frames to estimate the scale change. In addition, we train an online random fern classifier to re-detect objects in case of tracking failure. Extensive experimental results on large-scale benchmark datasets show that the proposed algorithm performs favorably against state-of-the-art methods in terms of efficiency, accuracy, and robustness.

1. Introduction

Object tracking is one of the most fundamental problems in computer vision with numerous applications. A typical scenario of visual tracking is to track an unknown object initialized by a bounding box in subsequent image frames. In this paper, we focus on the problem of long-term visual tracking, where target objects undergo significant appearance change due to deformation, abrupt motion, heavy occlusion and out-of-view.

Our approach builds on two major observations based on prior work. First, there is little change between two consecutive frames as the time interval is small (less than 0.04 second)[1] and the context around the target remains possibly unchanged even if the object is heavily occluded. Hence, it is important to model the temporal relationship of appearance consisting of a target object and its context. We develop a kernel ridge regression method based on correlation filters to encode the appearance template consisting of a target object and its surrounding context. The adaptive templates constructed by the proposed features are resistant to heavy occlusion, fast motion, and large deformation. This method differs significantly from existing correlation filters based tracking algorithms, which are prone to drifting in long-term tracking. Figure 1 shows one example where the proposed algorithm performs well against the KCF [11] and STC [28] methods. Our main contribution is an algorithm that efficiently models the temporal context information using correlation filters for long-term visual tracking.

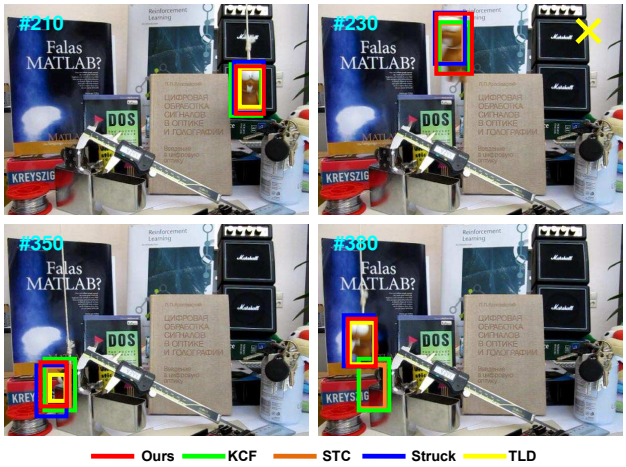

Figure 1. Comparisons of our approach with state-of-the-art trackers in challenging situations of fast motion, significant deformation and long-term occlusion on the *Lemming* sequence [24]. Our tracker takes temporal context into account for translation estimation, and performs robustly to abrupt motion and significant deformation in the 230th frame than the Struck [9] and TLD [14] methods. Our tracker is more effective in re-detecting the target in the 380th frame after long-term occlusion than the KCF [11] and STC [28] methods with the use of an online detector.

Second, it is critical to enhance the detection module of a long-term tracker to (i) estimate the scale change and (ii) re-detect the object in case of tracking failure when long-term occlusion or out-of-view occurs. For scale estimation, we train another correlation filter for a target from the most reliable frames. We use the histogram of orientation gradients (HOG) [4] as features to construct a multi-scale target pyramid and search for the optimal scale exhaustively. For object re-detection, we do not apply the target correlation filter

[1]Most videos have more than 25 frames per second.

978-1-4673-6964-0/15/$31.00 ©2015 IEEE

to scan across the entire frame due to computational efficiency as this filter is trained in the high-dimensional HOG feature space. We instead train an online detector by using a random fern [18] classifier and scan through the window when it is activated.

We further address two issues of tracking-by-detection approaches where tracking is usually formulated as an online learning problem with the goal of learning an appearance classifier discriminating the target from the background. The first issue is the well-known stability-plasticity dilemma [16, 20]. If the classifier is trained with more stable samples, e.g., only the target in the first frame, it is more robust to occlusions and less prone to drifting caused by model update with noisy samples. However, such an approach does not take appearance change into account and is unlikely to perform well for long-term tracking. On the other hand, highly adaptive online classifiers easily result in drifting in the case of noisy updates [16]. Our algorithm effectively alleviates this dilemma by modelling the temporal context correlation and the target appearance using two regression models based on correlation filters with different adaptive rates. The temporal context regressor is designed to aggressively adapt to translation estimation against significant deformation and heavy occlusion. The target regressor is conservatively adapted and applied on an appearance pyramid for scale estimation. Therefore, our approach effectively adapts to appearance change and alleviates the risk of drifting. Another issue with online classifiers is the sampling ambiguity, where hard negative samples are necessary to train a robust classifier and the binary labels are less effective for representing the spatial relationship between samples. By transferring the correlation procedure into an element-product in the Fourier domain, our regression models consider all the circular shifts [10, 11] of input features as training samples with Gaussian-weighted labels and thus alleviates the sampling problem.

One main contribution of this work is to address the problem of long-term visual tracking by effectively decomposing the tracking task into translation and scale estimation of target objects in conjunction with a complementary re-detection scheme. The translation estimation relies on a temporal context regression model robust against significant deformation, illumination variation, background clutter, and abrupt motion. Equipped with the estimated translation, a target pyramid is constructed to determine the scale change by using a target regression model. Our approach effectively alleviates the model update problems which often leads to drifting, and performs robustly in complex image sequences with large scale variations. In addition, we propose a novel scheme to activate target re-detection in case of tracking failure and make a decision whether to adopt the re-detected results by using the target regressor. We evaluate the proposed tracking algorithm on a large-scale benchmark with 50 challenging image sequences [24]. Extensive experimental results show that the proposed long-term correlation tracking algorithm performs favorably against state-of-the-art methods in terms of accuracy, efficiency, and robustness.

2. Related work and Problem Context

Visual tracking has been studied extensively with numerous applications [25, 21]. In this section, we discuss the methods closely related to this work: (i) correlation tracking and (ii) tracking-by-detection.

Correlation tracking. Correlation filters have been widely used in numerous applications such as object detection and recognition [15]. Since the operator is readily transfered into the Fourier domain as element-wise multiplication, correlation filters have attracted considerable attention recently to visual tracking due to its computational efficiency. Bolme et al. propose to learn a minimum output sum of squared error (MOSSE) [3] filter for visual tracking on gray-scale images, where the learned filter encodes target appearance with update on every frame. With the use of correlation filters, the MOSSE tracker is computationally efficient with a speed reaching several hundreds frames per second. Heriques et al. propose to use correlation filters in a kernel space with the CSK method [10] which achieves the highest speed in a recent benchmark [24]. The CSK method builds on illumination intensity features and is further improved by using HOG features in the KCF tracking algorithm [11]. In [6], Danelljan et al. exploit the color attributes of a target object and learn an adaptive correlation filter by mapping multi-channel features into a Gaussian kernel space. Recently, Zhang et al. [28] incorporate context information into filter learning and model the scale change based on consecutive correlation responses. The DSST tracker [5] learns adaptive multi-scale correlation filters using HOG features to handle the scale change of target objects. However, these methods do not address the critical issues regarding online model update. Therefore, these correlation trackers are susceptible to drifting and less effective for handling long-term occlusion and out-of-view problems. Figure 1 shows one example where the KCF method is more effective in handling the fast motion and deformation than the Struck and TLD methods in the 230th frame of the *Lemming* sequence, but fails to track the target object after long-term occlusion in the 380th frame due to the stability-plasticity problem (where the model is updated adequately in the 230th frame but incorrectly in the 380th frame).

Tracking-by-detection. To alleviate the stability-plasticity dilemma regarding online model update in visual tracking, Kalal et al. decompose the tracking task into tracking, learning and detection (TLD) [14] where tracking and detection facilitates each other, i.e., the results from the tracker provide training data to update the detector, and the detector

re-initializes the tracker when it fails. This mechanism is shown to perform well for long-term tracking [19, 22, 12]. Zhang et al. combine multiple classifiers with different adaptive rates and design an entropy measure to fuse all the tracking outputs [27]. Our algorithm bears some resemblance to these two methods with significant differences in that the tracking components in [14, 22, 12] are based on the Lucas-Kanade method [2] without fully exploiting temporal context information. In this work, we use a ridge regression model to learn the temporal correlation of context rather than a binary classifier (e.g., the online SVM classifier used in [27]). To alleviate the problem of noisy samples for online model update, Hare et al. [9] consider the spatial distribution of samples within a search space and propose to learn a joint structured output (Struck) to predict the object location, which has been shown to perform well [24]. Since the correlation operator is computed in the Fourier domain and takes all the circular shifts of input features into account, the regression model effectively handles the sampling ambiguity problem prevalent in online tracking with an online binary classifier.

3. Tracking Components

As we aim to develop an online tracking algorithm that is adaptive to significant appearance change without being prone to drifting, we decompose the task into translation and scale estimation of objects, where the translation is estimated by using the correlation of the temporal context and the scale estimation is carried out by learning a discriminative correlation filter. In addition, we train a complementary detector using online random ferns [18] to re-detect target objects in case of tracking failure.

3.1. Correlation Tracking

A typical tracker [3, 10, 6, 28, 5] based on correlation filters models the appearance of a target object using a filter $\mathbf{w}$ trained on an image patch $\mathbf{x}$ of $M \times N$ pixels, where all the circular shifts of $\mathbf{x}_{m,n}$, $(m,n) \in \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}$, are generated as training samples with Gaussian function label $y(m,n)$, i.e.,

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{m,n} |\phi(\mathbf{x}_{m,n}) \cdot \mathbf{w} - y(m,n)|^2 + \lambda |\mathbf{w}|^2, \quad (1)$$

where ϕ denotes the mapping to a kernel space and λ is a regularization parameter ($\lambda \geq 0$). Since the label $y(m,n)$ is not binary, the learned filter $\mathbf{w}$ contains the coefficients of a Gaussian ridge regression [17] rather than a binary classifier. Using the fast Fourier transformation (FFT) to compute the correlation, this objective function is minimized as $\mathbf{w} = \sum_{m,n} \mathbf{a}(m,n)\phi(\mathbf{x}_{m,n})$, and the coefficient $\mathbf{a}$ is defined by

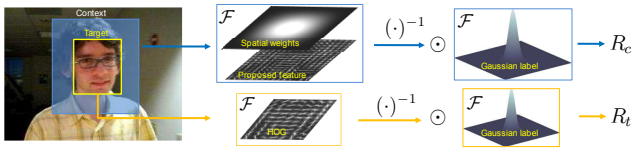

Figure 2. Two regression models learned from a single frame. The model R_c exploits the temporal correlation of target and surrounding context while R_t only models target appearance. To train the model R_c, a layer of spatial weights are added on the feature space. Here $\mathcal{F}$ denotes the discrete Fourier operator and $\odot$ is the Hadamard product.

$$A = \mathcal{F}(\mathbf{a}) = \frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}(\phi(\mathbf{x}) \cdot \phi(\mathbf{x})) + \lambda}. \quad (2)$$

In (2), $\mathcal{F}$ denotes the discrete Fourier operator and $\mathbf{y} = \{y(m,n) | (m,n) \in \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}\}$. The tracking task is carried out on an image patch $\mathbf{z}$ in the new frame with the search window size $M \times N$ by computing the response map as

$$\hat{\mathbf{y}} = \mathcal{F}^{-1}\big(A \odot \mathcal{F}(\phi(\mathbf{z}) \cdot \phi(\hat{\mathbf{x}}))\big), \quad (3)$$

where $\hat{\mathbf{x}}$ denotes the learned target appearance model and $\odot$ is the Hadamard product. Therefore, the new position of target is detected by searching for the location of the maximal value of $\hat{\mathbf{y}}$.

Differently from prior work, we train two regression models based on correlation filters from one single frame. As shown in Figure 2, the temporal context model R_c takes both the target and surrounding context into account, since this information remains temporally stable and useful to discriminate the target from the background in the case of occlusion [28]. To remove the boundary discontinuities of the response map, the extracted feature channels of the target and context are weighted by a cosine window [3]. It is important for the regression model R_c to be adaptive to estimate the translation when the target undergoes occlusion, deformation, and abrupt motion. The R_c model is thus updated with a learning rate α frame by frame as

$$\hat{\mathbf{x}}^t = (1-\alpha)\hat{\mathbf{x}}^{t-1} + \alpha \mathbf{x}^t, \quad (4a)$$

$$\hat{A}^t = (1-\alpha)\hat{A}^{t-1} + \alpha A^t, \quad (4b)$$

where t is the index of the current frame.

In contrast to existing tracking methods [20, 30] where the target in the first frame is used to measure confidence of tracking results in following frames, we learn another discriminative regression model R_t from the most reliable tracked targets. Specifically, we use the maximal value of $\hat{\mathbf{y}}$ to determine the confidence of tracking results. To maintain the model stability, we use a pre-defined threshold $\mathcal{T}_a$ and

only update R_t using (4) if $\max(\hat{\mathbf{y}}) \geq \mathcal{T}_a$. Note that there are no cosine spatial weights for model R_t in the feature space (See Figure 2). During tracking, we construct a target pyramid around the estimated translation location for scale estimation (See Figure 3). Let $P \times Q$ be the target size in a test frame and N indicate the number of scales $S = \{a^n | n = \lfloor -\frac{N-1}{2} \rfloor, \lfloor -\frac{N-3}{2} \rfloor, \ldots, \lfloor \frac{N-1}{2} \rfloor\}$. For each $s \in S$, we extract an image patch J_s of size $sP \times sQ$ centered around the estimated location. Unlike [5], we propose to uniformly resize all patches with size $P \times Q$ again and use HOG features to construct the scale feature pyramid. Let $\hat{\mathbf{y}}_s$ denote the correlation response map of the target regressor R_t to J_s, the optimal scale $\hat{s}$ of target is

$$\hat{s} = \underset{s}{\operatorname{argmax}} \big(\max(\hat{\mathbf{y}}_1), \max(\hat{\mathbf{y}}_2), \ldots, \max(\hat{\mathbf{y}}_S)\big). \quad (5)$$

Accordingly, the regression model R_t is updated by (4) if $\max(\hat{\mathbf{y}}_{\hat{s}}) \geq \mathcal{T}_a$.

3.2. Online Detector

It is clear that a robust long-term tracking algorithm requires a re-detection module in the case of tracking failure, e.g., long-term occlusion and re-entering the field of view. Different from previous trackers [19, 22, 12], where re-detection is carried out on each frame, we use a threshold $\mathcal{T}_r$ to activate the detector if $\max(\hat{\mathbf{y}}_{\hat{s}}) < \mathcal{T}_r$. For computational efficiency, we do not use the regression model R_t as a detector and instead use the online random fern classifier [14]. As the detector is applied to the entire frame with sliding windows when $\max(\hat{\mathbf{y}}_{\hat{s}}) < \mathcal{T}_r$, we train an online random ferns detector with a conservative update scheme. Let c_i, $i \in \{0, 1\}$ be the indicator of class labels and let f_j, $j \in \{1, 2, \ldots, N\}$ be the set of binary features, which are grouped into small sets as ferns. The joint distribution for features in each fern is

$$P(f_1, f_2, \ldots, f_N | C = c_i) = \prod_{k=1}^{M} P(F_k | C = c_i), \quad (6)$$

where $F_k = \{f_\sigma(k, 0), f_\sigma(k, 2), \ldots, f_\sigma(k, N)\}$ represents the k-th fern, and $\sigma(k, n)$ is a random permutation function with range from 1 to N. For each fern F_k, its conditional probability can be written as $P(F_k | C = c_i) = \frac{N_{k,c_i}}{N_k}$, where N_{k,c_i} is the number of training samples of class c_i that belongs to the k-th fern and N_k is the total number of training samples that fell into the leaf-node corresponding to the k-th fern. From the Bayesian perspective, the optimal class $\hat{c}_i$ is detected as $\hat{c}_i = \operatorname{argmax}_{c_i} \prod_{k=1}^{M} P(F_k | C = c_i)$ [18].

4. Implementation

We present an outline of our method in Algorithm 1 and show the flowchart of our method in Figure 3. More implementation details are discussed as follows.

Algorithm 1: Proposed tracking algorithm.

Input : Initial target bounding box $\mathbf{x}_0$,
Output: Estimated object state $\mathbf{x}_t = (\hat{x}_t, \hat{y}_t, \hat{s}_t)$, temporal context regression model R_c, target appearance regression model R_t, and random fern detector D_{rf}.

repeat
- Crop out the searching window in frame t according to $(\hat{x}_{t-1}, \hat{y}_{t-1})$ and extract the features;
- `// Translation estimation`
- Compute the correlation map $\mathbf{y}_t$ using R_c and (3) to estimate the new position (x_t, y_t);
- `// Scale estimation`
- Build the target pyramid around (x_t, y_t) and compute the correlation map $\mathbf{y}_s$ using R_t and (3);
- Estimate the optimal scale $\hat{s}$ using (5);
- $\mathbf{x}_t = (x_t, y_t, \hat{s})$;
- `// Target re-detection`
- **if** $\max(\mathbf{y}_{\hat{s}}) < \mathcal{T}_r$ **then**
  - Use detector D_{rf} to perform re-detection and find the possible candidate states X;
  - **foreach** *state* $\mathbf{x}'_i$ *in* X **do** computing confidence score $\mathbf{y}'_i$ using R_t and (3);
  - **if** $\max(\mathbf{y}'_i) > \mathcal{T}_t$ **then** $\mathbf{x}_t = \mathbf{x}'_i$, where $i = \operatorname{argmax}_i \mathbf{y}'_i$;
- **end**
- `// Model update`
- Update R_c using (4);
- **if** $\max(\mathbf{y}_{\hat{s}}) > \mathcal{T}_a$ **then**
  - Update R_t using $J_{\hat{s}}$ and (4);
- **end**
- Update D_{rf};

until *End of video sequences*;

Features. In this work, each feature vector $\mathbf{x}$ is represented by a concatenation of multiple channels [7]. In addition to HOG features with 31 bins, we use another histogram feature of intensity in a 6×6 local window with 8 bins. To provide robustness to drastic illumination variations, we compute the histogram of local intensity on brightness channel and we also add a transformed channel by applying a non-parametric local rank transformation [26] on the brightness channel. Therefore, we use feature vectors with 47 channels to train the temporal context regressor R_c. For the target model R_t, we only use HOG features to construct the target pyramid. For the random fern detector, each tracked result with high confidence is resized to 15×15 to form a feature vector of intensity values.

Kernel selection. We use a Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{|\mathbf{x}-\mathbf{x}'|^2}{\sigma^2})$, which defines a mapping ϕ as $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}')$, in both regression models R_c and R_t. We

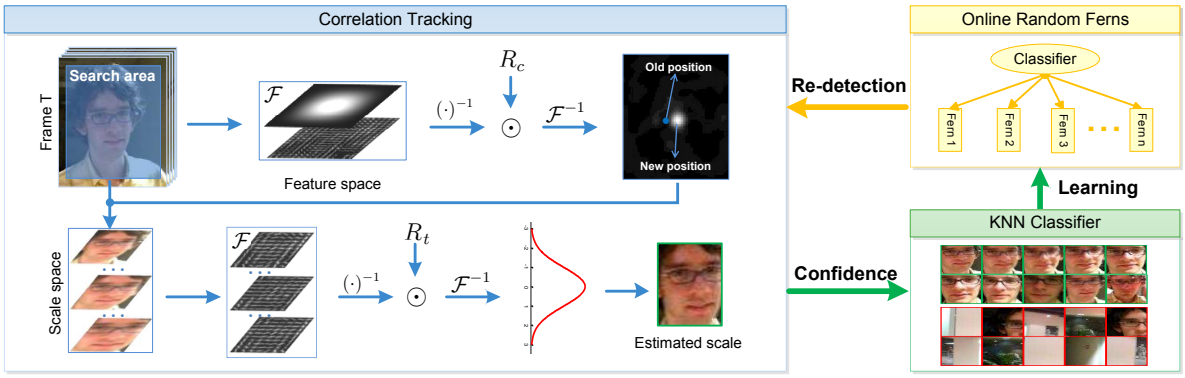

Figure 3. Flowchart of the proposed tracking algorithm. The tracking task is decomposed into translation and scale estimation: translation is estimated by the temporal context regression model R_c, and scale is estimated by the target appearance model R_t. The KNN classifier selects the most confident tracked results to train an detector using random ferns for re-detecting the target in the case of tracking failure.

compute the full kernel correlation in (2) and (3) efficiently in the Fourer domain.

Random ferns. In our implementation, the detector trained by a random fern classifier uses pixel comparison as binary features in a way similar to [18]. Each fern performs a number of pixel comparisons on the patch with two feature vectors that point to the leaf-node with the posterior probability. The posteriors from all ferns are averaged as target response. Similarly to [23], detection is based on the scanning window strategy. Unlike [14], where the P-N ferns are updated online, we use a k-nearest neighbor (KNN) classifier to select the most confident tracked results as positive training samples, e.g., a new patch is predicted as the target if k nearest feature vectors in the training set all have positive labels (e.g., $k = 5$ in this work).

5. Experimental Results

We evaluate the proposed method on a large benchmark dataset [24] that contains 50 videos with comparisons to state-of-the-art methods. All the tracking methods are evaluated by three metrics, (i) distance precision, which shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth; (ii) overlap success rate, which is defined as the percentage of frames where the bounding box overlap surpasses a threshold; and (iii) center location error, which indicates the average Euclidean distance between the ground-truth and the estimated center location. More results can be found in the supplementary material.

Setups. The regularization parameter of (1) is set to $\lambda = 10^{-4}$. The size of the search window for translation estimation is set to 1.8 times of the target size. The Gaussian kernel width σ is set to 0.1. The learning rate α in (4) is set to 0.01. The number of scale space is $|S| = 21$ and the scale factor a is set to 1.08. There are several thresholds for correlation tracking. We set a lower threshold $\mathcal{T}_r = 0.25$ to activate the trained random fern detector, and set a higher threshold $\mathcal{T}_t = 0.5$ to adopt the re-detection result. The threshold settings indicate that we rely on correlation tracking results. We set $\mathcal{T}_a = 0.5$ to update the target regressor R_t. We use the same parameter values for all the sequences. The proposed tracking algorithm is implemented in Matlab on an Intel I7-4770 3.40 GHz CPU with 32 GB RAM, and the source code and more experimental results are available at `http://faculty.ucmerced.edu/mhyang/`.

Component analysis. We implement three more algorithms based on correlation filters to demonstrate the effectiveness of the proposed long-term correlation tracking (LCT) algorithm. First, we implement a tracker (CTHOG) by learning a single correlation filter using HOG features as our baseline algorithm. We also implement a tracker (CTNRE) without an online detector by learning a single correlation filter with the proposed 47 channel features used in the regressor R_c. In addition, a tracker similar to the proposed LCT method without scale estimation is referred to as CTFSC. We report the results on the 50 benchmark sequences using the distance precision and overlap success rate by the area-under-the-curve (AUC). As shown in Figure 4, the compared CTNRE tracker outperforms the CTHOG method due to the use of histogram of intensity. The CTFSC tracker significantly outperforms the CTNRE method due to the effectiveness of the target re-detection scheme in case of tracking failure. The proposed LCT algorithm (equipped with all the components) performs favorably against the other three alternative implementations. Although the CTFSC tracker performs well in distance precision, it is not effective in dealing with scale change.

Overall performance. We evaluate the proposed algorithm on the benchmark with comparisons to 11 state-of-the-art trackers from three typical categories of tracking algorithms, (i) correlation trackers (CSK [10], STC [28], and KCF [11]); (ii) tracking by single online classifier (MIL [1],

Table 1. Comparisons with state-of-the-art trackers on the 50 benchmark sequences. Our approach performs favorably against existing methods in distance precision (DP) at a threshold of 20 pixels, overlap success (OS) rate at an overlap threshold 0.5 and center location error (CLE). The first and second highest values are highlighted by bold and underline.

	LCT	CSK [10]	STC [28]	KCF [11]	MIL [1]	Struck [9]	CT [29]	ASLA [13]	TLD [14]	SCM [30]	MEEM [27]	TGPR [8]
DP (%)	**85.4**	54.5	54.7	74.1	47.5	65.6	40.6	53.2	60.8	64.9	<u>74.4</u>	70.5
OS (%)	**76.9**	44.3	36.5	62.2	37.3	55.9	34.1	51.1	52.1	61.6	<u>64.9</u>	62.8
CLE (pixel)	**25.8**	88.8	80.5	<u>35.5</u>	62.3	50.6	78.9	73.1	48.1	54.1	41.6	51.3
Speed (FPS)	27.4	**269**	<u>232</u>	39.1	28.1	10.0	38.8	7.5	21.7	0.4	19.4	0.7

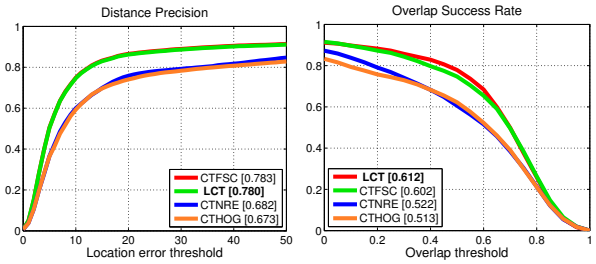

Figure 4. Comparisons of component effectiveness. The CTHOG and CTNRE implementations are based on a single correlation filter with different features (e.g., HOG and the proposed features with 47 channels used in regressor R_c respectively). The CTFSC tracker is similar to the proposed LCT method while incapable of scale estimation. The proposed LCT algorithm performs favorably against the other three alternative implementations and is able to deal with scale change.

Struck [9], CT [29], and ASLA [13]); and (iii) tracking by multiple online classifiers (TLD [14], SCM [30], MEEM [27], and TGPR [8]). For fair evaluations, we compare all the methods on gray scale images following the protocol of the benchmark study [24]. We report the results in one-pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) using the distance precision and overlap success rate in Figure 5. In addition, we present the quantitative comparisons of distance precision at 20 pixels, overlap success rate at 0.5, center location errors, and tracking speed in Table 1.

Table 1 shows that our algorithm performs favorably against state-of-the-art methods in distance precision (DP), overlap success (OS) and center location error (CLE). Among the trackers in the literature, the MEEM method achieves the best results with an average DP of 74.4% and OS of 64.9%. Our algorithm performs well with DP of 85.4% and OS of 76.9%. The KCF tracker performs well with CLE of 35.5 pixels and our method achieves lower CLE of 25.8 pixels. While the CSK, STC and KCF methods achieves higher frame rate than the LCT method, our algorithm performs well at 27.4 frames per second. The main computational load of our tracker is the feature pyramid construction for scale estimation.

Figure 5 shows that our approach performs well against the existing methods (KCF, MEEM) in OPE, TRE and SRE validation schemes. Note that although the TRE and SRE

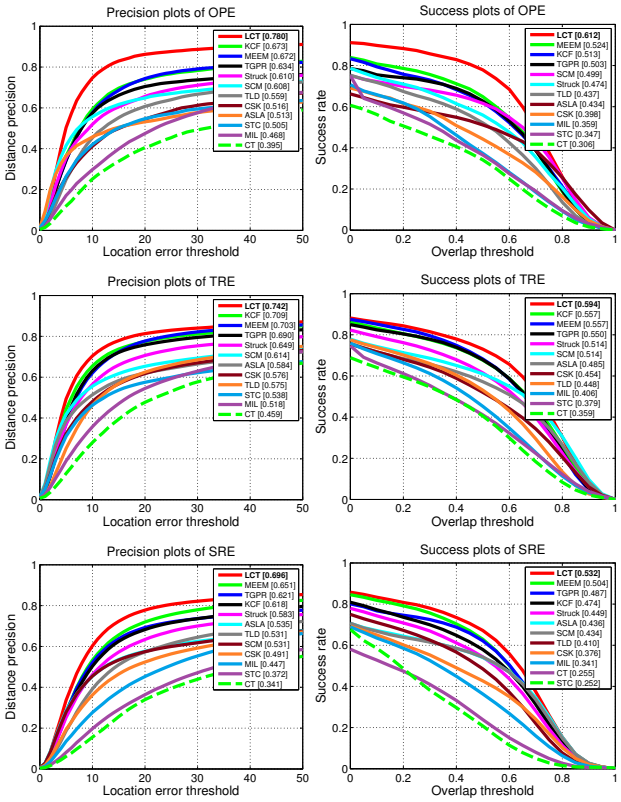

Figure 5. Distance precision and overlap success plots over 50 benchmark sequences using one-pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE). The legend contains the area-under-the-curve score for each tracker. The proposed LCT method performs favorably against the state-of-the-art trackers.

evaluation schemes do not fully reflect the merits of our approach (e.g., TRE splits a video into several fragments and the importance of target re-detection in long term tracking is less accounted for, and SRE spatially shifts the bounding boxes and thus the importance of temporal context correlation is considered less), the proposed algorithm still performs well against state-of-the-art methods.

Attribute-based evaluation. The videos in the benchmark dataset [24] are annotated with 11 attributes to describe the different challenges in the tracking problem, e.g., occlusions or out-of-view. These attributes are useful for analyzing the performance of trackers in different

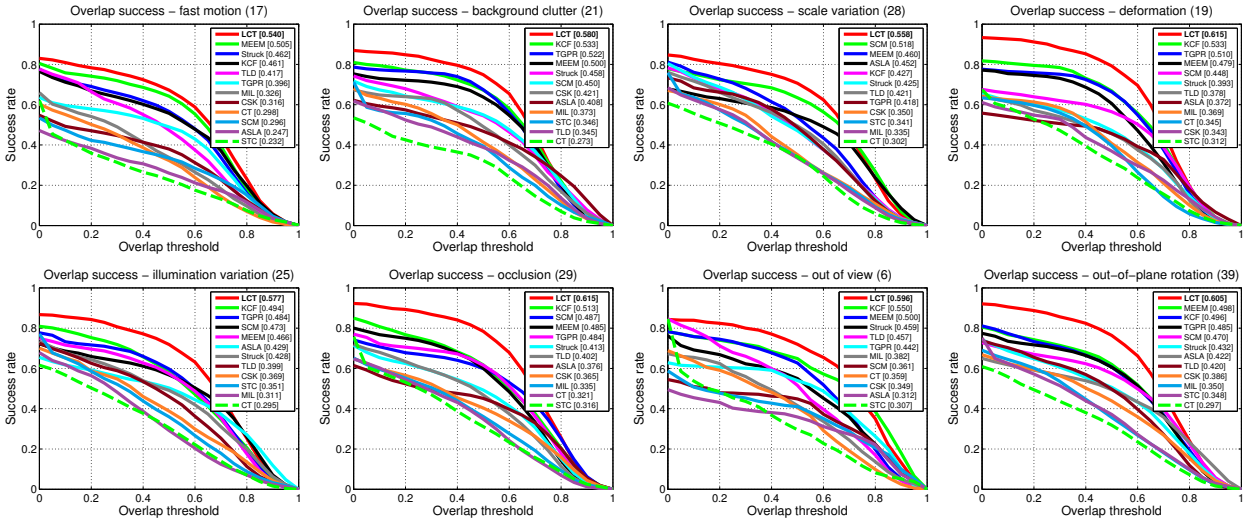

Figure 6. Overlap success plots over eight tracking challenges of fast motion, background clutter, scale variation, deformation, illumination variation, occlusion, out-of-view, and out-of-plane rotation. The legend contains the AUC score for each tracker. The proposed LCT method performs favorably against the state-of-the-art trackers when evaluating with eight challenging factors.

aspects. We report results for eight main challenging attributes in Figure 6. Among existing methods, the KCF method performs well with overall success in background clutter (53.3%), deformation(53.3%), illumination(49.4%), occlusion(51.3%) and out-of-view (55.0%) while the LCT algorithm achieves success rate of 58.0%, 61.5%, 57.7%, 61.5%, and 59.6% respectively. The MEEM method performs well in fast motion (50.5%) and out-of-plane rotation (49.8%), while the LCT algorithm achieves the success rate of 54.0% and 60.6%. In terms of scale variation, the SCM method achieves the success rate of 51.8% while the LCT algorithm performs well with success rate of 55.8%.

Qualitative evaluation. We compare our algorithm with other four state-of-the-art trackers (KCF [11], STC [28], Struck [9], and TLD [14]) on twelve challenging sequences in Figure 7. The KCF tracker is based on a correlation filter learned from HOG features and thus similar to our baseline implementation CTHOG (See also Figure 4). The KCF tracker performs well in handling significant deformation and fast motion (*Fleetface*, *David*, and *Singer2*) due to the robust representation of HOG features and effectiveness of the temporal context correlation model. However, it drifts when target objects undergo heavy occlusions (*Coke*) and does not re-detect targets in the case of tracking failure (*Tiger2* and *Jogging-2*). In addition, the KCF tracker fails to handle background clutter (*Shaking*), where only HOG features are less effective to discriminate targets from the cluttered background. The STC tracker is also based on a correlation filter and able to estimate scale changes, but does not perform well when both significant scale and rotation occur (*Trellis*) or in the presence of abrupt motion (*Jumping*) as it only learns the filter from brightness channel and estimates the scale change based on a temporal context model rather than a target appearance model. The Struck tracker does not perform well in rotation (*David*), background clutter (*Singer2*), heavy occlusion or out-of-view (*Tiger2* and *Jogging-2*) since it is less effective in handling appearance change caused by multiple factors with one single classifier. The TLD tracker is able to re-detect target objects in the case of tracking failure. However, the TLD method does not fully exploit the temporal motion cues as our approach and therefore does not follow targets undergoing significant deformation and fast motion (*Tiger2*, *Shaking*, and *Singer2*) well. Moreover, the TLD method updates its detector frame-by-frame leading to drifting (*Trellis* and *Skating1*) and false target re-detection (*Jogging-2*). Overall, the proposed LCT tracker performs well in estimating both the scales and positions of target objects on these challenging sequences, which can be attributed to three reasons. First, our temporal context regressor R_c is learned from more robust features rather than only HOG features or simple brightness intensity and it is effective in estimating the translation of target objects. The proposed features are less sensitive to illumination and background clutter (*Shaking* and *Singer2*), rotation (*David*), and partial occlusion (*Coke* and *Tiger2*). Second, the target regressor R_t is conservatively updated and the errors of the scale estimation are not accumulated to affect following frames. Therefore, our method effectively alleviates the scale drifting problem (*Trellis* and *Jumping*). Third, the trained detector effectively re-detects target objects in the case of tracking failure, e.g., with the heavy occlusion (*Coke* and *Jogging-2*) and out-of-view (*Tiger2*).

In addition, we compare the center location error frame-by-frame on the twelve sequences in Figure 8, which shows that our method performs well against existing trackers.

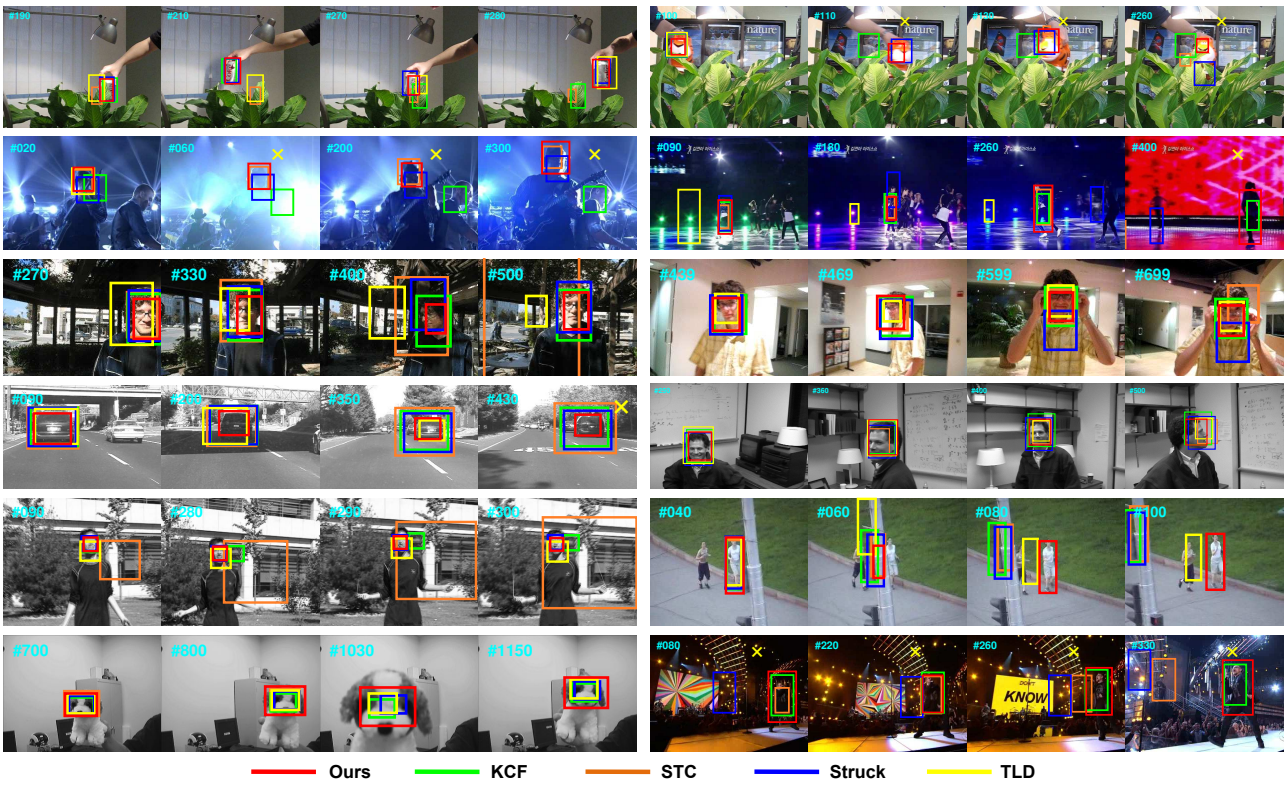

Figure 7. Tracking results of our LCT algorithm, KCF [11], STC [28], Struck [9] and TLD [14] methods on twelve challenging sequences (from left to right and top to down are *Coke*, *Tiger2*, *Shaking*, *Skating1*, *Trellis*, *David*, *Car4*, *Fleetface*, *Jumping*, *Jogging-2*, *Dog1*, and *Singer2*, respectively).

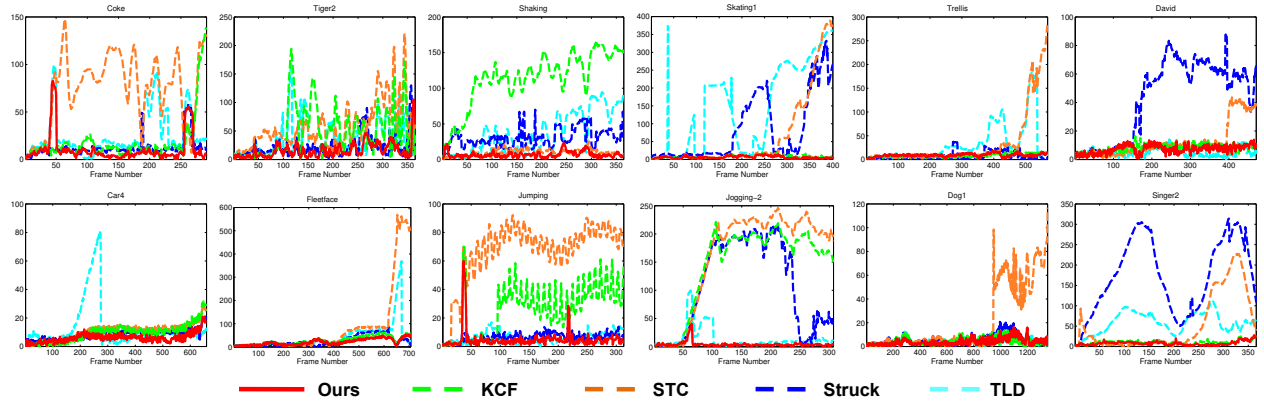

Figure 8. Fame-by-frame comparison of center location errors (in pixels) on twelve challenging sequences in Figure 7. Generally, our method is able to track targets accurately and stably. In particular on the *Coke*, *Jumping* and *Jogging-2* sequences, our tracker drifts in the 40th, 42ed and 60th frames respectively due to heavy occlusion or out-of-view, but manages to re-detect the targets subsequently in a short period.

6. Conclusions

In this paper, we propose an effective algorithm for long-term visual tracking. Our method learns discriminative correlation filters for estimating the translation and scale variations of target objects effectively and efficiently. The translation is estimated by modeling the temporal context correlation and the scale is estimated by searching the target appearance pyramid exhaustively. We further develop a robust online detector using random ferns to re-detect targets in case of tracking failure. Extensive experimental results show that the proposed algorithm performs favorably against the state-of-the-art methods in terms of efficiency, accuracy, and robustness.

Acknowledgment

C. Ma is sponsored by China Scholarship Council. X. Yang and C. Zhang are supported in part by the NSFC Grants #61025005, #61129001 and #61221001, STCSM Grants #14XD1402100 and #13511504501, 111 Program Grant #B07022, and the CNKT R&D Program Grant #2012BAH07B01. M.-H. Yang is supported in part by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

References

[1] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8), 2011.

[2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[3] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[5] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *Proceedings of British Machine Vision Conference*, 2014.

[6] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer. Adaptive color attributes for real-time visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[7] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014.

[8] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian processes regression. In *Proceedings of the European Conference on Computer Vision*, 2014.

[9] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *Proceedings of IEEE International Conference on Computer Vision*, 2011.

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proceedings of the European Conference on Computer Vision*, 2012.

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[12] Y. Hua, K. Alahari, and C. Schmid. Occlusion and motion reasoning for long-term tracking. In *Proceedings of the European Conference on Computer Vision*, 2014.

[13] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[14] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

[15] B. V. K. V. Kumar, A. Mahalanobis, and R. D. Juday. *Correlation Pattern Recognition*. Cambridge University Press, 2005.

[16] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):810–815, 2004.

[17] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[18] M. Özuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[19] F. Pernici. Facehugger: The ALIEN tracker applied to faces. In *Proceedings of the European Conference on Computer Vision*, 2012.

[20] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: parallel robust online simple tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[21] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, 2014.

[22] J. S. Supancic and D. Ramanan. Self-paced learning for long-term tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[23] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[24] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[25] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006.

[26] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision*, 1994.

[27] J. Zhang, S. Ma, and S. Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *Proceedings of the European Conference on Computer Vision*, 2014.

[28] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang. Fast visual tracking via dense spatio-temporal context learning. In *Proceedings of the European Conference on Computer Vision*, 2014.

[29] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *Proceedings of the European Conference on Computer Vision*, 2012.

[30] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparse collaborative appearance model. *IEEE Transactions on Image Processing*, 23(5):2356–2368, 2014.