Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Learning Patch-Based Dynamic Graph for Visual Tracking

Chenglong Li,[1,2] Liang Lin,[2*] Wangmeng Zuo,[3] Jin Tang[1]
[1]School of Computer Science and Technology, Anhui University, Hefei,China
[2]School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[3]School of Computer Science and Technology, Harbin Institute of Technology, China
lcl1314@foxmail.com, linliang@ieee.org, cswmzuo@gmail.com, tj@ahu.edu.cn

Abstract

Existing visual tracking methods usually localize the object with a bounding box, in which the foreground object trackers/detectors are often disturbed by the introduced background information. To handle this problem, we aim to learn a more robust object representation for visual tracking. In particular, the tracked object is represented with a graph structure (i.e., a set of non-overlapping image patches), in which the weight of each node (patch) indicates how likely it belongs to the foreground and edges are also weighed for indicating the appearance compatibility of two neighboring nodes. This graph is dynamically learnt (i.e., the nodes and edges received weights) and applied in object tracking and model updating. We constrain the graph learning from two aspects: i) the global low-rank structure over all nodes and ii) the local sparseness of node neighbors. During the tracking process, our method performs the following steps at each frame. First, the graph is initialized by assigning either 1 or 0 to the weights of some image patches according to the predicted bounding box. Second, the graph is optimized through designing a new ALM (Augmented Lagrange Multiplier) based algorithm. Third, the object feature representation is updated by imposing the weights of patches on the extracted image features. The object location is finally predicted by adopting the Struck tracker (Hare, Saffari, and Torr 2011). Extensive experiments show that our approach outperforms the state-of-the-art tracking methods on two standard benchmarks, i.e., OTB100 and NUS-PRO.

Introduction

Existing successful visual tracking methods mainly adopt the tracking-by-detection paradigm, i.e., separating the foreground object from its background over time by maintaining a classifier on the fly. These methods usually localize the object using a bounding box, and draw positive (negative) samples from inside (outside) of the bounding box for the classifier updating. Since the ground-truth object labelling is only available at the initial frame, incrementally updating the object classifier in subsequent frames often undertakes the risk of model drifting due to introducing outlier samples.

In literature of visual tracking, many efforts have been devoted to alleviate the effects of outlier samples (Comaniciu, Ramesh, and Meer 2003; Hare, Saffari, and Torr 2011; He et al. 2013; Zhang, Ma, and Sclaroff 2014; Kim et al. 2015). For example, the methods in (Comaniciu, Ramesh, and Meer 2003; Hare, Saffari, and Torr 2011; He et al. 2013) update the object classifiers by considering the distances of samples with respect to the bounding box center, e.g., the samples close to the center receiving higher weights. Some other methods (Duffner and Garcia 2013; Yang, Lu, and Yang 2014) performs object segmentation during the tracking process to exclude background information. However, these methods are limited in dealing with cluttered backgrounds (e.g., unreliable segmented object masks). To improve the robustness, Kim et al. (Kim et al. 2015) proposed to define an image patch based 8-neighbor graph to represent the tracked object, in which the 8-neighbor graph denote that if two nodes are 8-neighbors, they are connected by an edge, and the edge weight is computed by their low-level feature distance. This approach has two main shortcomings: i) It only considers the spatial neighbors, and cannot capture the intrinsic relationship between patches; ii) It directly uses low-level features, which are easily contaminated by video noises.

To handle this problem, we aim to learn a more robust object representation for visual tracking. Given one bounding box of the target object, we partition it into non-overlapping local patches, which are described by color and gradient histograms. We take these patches as graph nodes, and the bounding box can thus be represented with a graph structure, in which the weight of each node describes how likely it belongs to the target object, and the edge weight between two neighboring patches indicates their appearance compatibility. In this work, we propose a novel weighted low-rank and sparse representation model to dynamically learn the graph for each frame that infers the edges and the node weights in a joint fashion.

According to Wright et al. (Wright et al. 2010), an informative graph should have three characteristics: high discriminative power, enhanced sparsity and adaptive neighborhood. Therefore, we represent each patch descriptor as a linear combination of other patch descriptors, and employ the non-negativeness, sparsity, and low-rank constraints to suppress

*Corresponding author: Liang Lin. This work was in part supported by State Key Development Program under Grant 2016YFB1001004, in part by the National Natural Science Foundation of China under Grant 61622214 and Grant 61472002, and in part by the Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase).

the effects of noises and/or corruptions of low-level features in computing edges. Simultaneously, we optimize the node weights in a semi-supervised way. i) The target bounding box is shrunk and expanded to obtain the initial foreground and background nodes, whose weights are set to be 1 and 0, respectively. ii) The node weights are diffused along the computed edges.

To improve the tracking efficiency, we design a new Augmented Lagrange Multiplier (ALM) based algorithm to efficiently seek the solution of the proposed model. In particular, we employ the linearized Alternating Direction Method (ADM) with adaptive penalty (Lin, Liu, and Su 2011) to separate the objective function into several convex subproblems while avoiding some matrix inversions, and then optimize them iteratively. To further reduce computational burden, we utilize the randomized singular value thresholding (SVT) method (Oh et al. 2015) to avoid direct computation on Sigular Value Decomposition (SVD). Finally, we incorporate the optimized patch weights into the struck tracker (Hare, Saffari, and Torr 2011) for object tracking and model updating.

This work makes the following three major contributions. First, we propose an effective approach to mitigate the effects of background information in visual tracking. Extensive experiments show that the proposed method outperforms the state-of-the-art trackers on two standard benchmarks, validating the effectiveness of the proposed approach. Second, we present a novel weighted low-rank and sparse representation model to learn a dynamic graph for each frame by considering non-negativeness, sparsity and low rank constraints among image patches. The proposed model provides a general solution that jointly infers the graph edges and the graph node weights for visual tracking and related problems. It can effectively exploit the intrinsic relationship of data, and thus is robust to data noises and/or corruptions. Third, we design a new ALM based algorithm to efficiently seek the solution of the associated optimization problem. Thanks to the proposed optimization algorithm, our tracker performs nearly real-time.

Related Work

Various tracking methods have been proposed to improve the robustness to nuisance factors including label ambiguity, background cluttering, corruption and occlusion. Grabner et al. (Grabner, Grabner, and Bischof 2008) presented a tracker which limits the drifting problem while still being adaptive to various appearance changes. The knowledge from labeled data was used to build a fixed prior for online classifier while unlabelled data was explored in a principled manner during tracking. Babenko et al. (Babenko, Yang, and Belongie 2011) employed a bag of multiple samples, instead of a single sample, to update the classifier reliably. To avoid the label ambiguity, Hare et al. (Hare, Saffari, and Torr 2011) employed structured samples instead of binary-labeled samples when training the classifier in the structured SVM framework (Tsochantaridis et al. 2005).

To improve the robustness to background cluttering, one representative strategy is to assign weights to different pixels (or patches) in the bounding box. Comaniciu et al. (Comaniciu, Ramesh, and Meer 2003) employed the kernel-based method to assign smaller weights to boundary pixels during the histogram construction. He et al. (He et al. 2013) also assumed that pixels far from a box center should be less important. These methods may fail when a target object has a complicated shape or is occluded. Some works (Duffner and Garcia 2013; Yang, Lu, and Yang 2014) integrated segmentation results into tracking to alleviate the effects of background. These algorithms, however, are sensitive to segmentation results. Kim et al. (Kim et al. 2015) developed a random walk restart algorithm on 8-neighbor graph to compute patch weights within target object bounding box. But the constructed graph may fail to capture the relationship between patches.

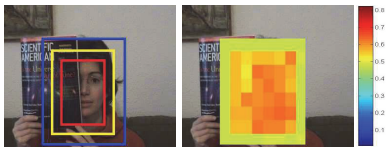

Figure 1: Illustration of the original, shrunk and expanded bounding boxes, which are represented by the yellow, red and blue colors, respectively. The optimized patch weights are also shown visually for clarity, in which the hotter color indicates the larger weight.

Patch-based Graph Learning

Given one bounding box of the target object, we partition it into non-overlapping local patches, and then assign each patch with a weight that reflects its importance in describing the target object to mitigate the effects of background information. We concatenate these weighting patch descriptors into a feature vector to represent the target object robustly, and then combine this feature vector with Struck (Hare, Saffari, and Torr 2011) to carry out object tracking. This section will introduce a novel weighted low-rank and sparse representation model to compute the patch weights, and then design a new ALM based algorithm to optimize the proposed model efficiently.

Representation

Each bounding box of the target object is partitioned into n non-overlapping patches, and a set of low-level appearance features are extracted and further combined into one single d-dimensional feature vector $\mathbf{x}_i$ for characterizing the i-th patch. We take these patches as graph nodes, and the bounding box can thus be represented with a graph structure, in which the weight of each node describes how likely it belongs to the target object, and the edge weight between two neighboring patches indicates their appearance compatibility.

On one hand, some patches in the target bounding box may belong to background due to irregular shape, scale variation and partial occlusion of the target object, as shown in Fig. 1. Therefore, we assign a weight for each graph node to mitigate the effects of background information in object tracking and object model updating. On the other hand, instead of constructing spatially adjacent graph in conventional methods (Yang et al. 2013;

Kim et al. 2015), the edges are dynamically learnt for capturing the intrinsic relationship of data, including global structures of the whole data (Zhuang et al. 2012; Liu et al. 2013) and datum-adaptive neighborhoods (Yan and Wang 2009; Yang et al. 2015). In this work, we propose a novel weighted low-rank and sparse representation model to infer the edges and the node weights in a joint manner, which can also provide a general solution for related problems.

All the feature vectors of n patches in one bounding box form the data matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \in \mathbb{R}^{d\times n}$. We assume that object or background patches are all drawn from the same low-rank subspace, and all patches lie on a union of multiple subspaces. Similar assumptions have been justified in some works on image and video segmentation (Cheng et al. 2011; Li et al. 2016b). Thus, each patch descriptor can be represented as a linear combination of remaining patch descriptors, and the non-negative low-rank and sparse representation of all patch vectors can then be formulated in a joint fashion: $\mathbf{X} = \mathbf{XZ},\ \mathbf{Z} \geq 0$, where $\mathbf{Z} \in \mathbb{R}^{n\times n}$ is the low-rank and sparse representation coefficient matrix. Sparse constraints can automatically select most informative neighbors for each patch (higher-order relationships), making the graph more powerful and discriminative (Yan and Wang 2009). Low-rank constraints can capture global structure of whole patches, and thus preserve membership of patches that belong to same subspace (Liu et al. 2013). We employ these two constraints to better capture intrinsic relationship among patches in constructing graph. Since the patch feature matrix is often noisy or grossly corrupted, the non-negative low-rank and sparse representation can be obtained by solving the objective function:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} rank(\mathbf{Z}) + \alpha||\mathbf{Z}||_0 + \lambda||\mathbf{E}||_{2,0}, \\ &s.t.\ \mathbf{X} = \mathbf{XZ} + \mathbf{E},\ \mathbf{Z} \geq 0, \end{aligned} \tag{1}$$

where $||\cdot||_0$, $||\cdot||_{2,0}$ are the l_0 norm and the $l_{2,0}$ norm, respectively. $rank(\cdot)$ denotes the $rank$ function, and $\mathbf{E} \in \mathbb{R}^{d\times n}$ denotes the sparse sample-specific corruptions, i.e., some patches are corrupted and others are clean. α and λ are the balanced parameters.

To mitigate the effects of background information, we assign a weight $\mathbf{w}_i$ for each patch i, and optimize them in a semi-supervised way. Let $\mathbf{q} = \{q_1, q_2, ..., q_n\}^T$ be an initial weight vector, in which $q_i = 1$ if q_i is a target object patch, and $q_i = 0$ indicating a background patch. $\mathbf{q}$ is computed by the initial ground truth (for first frame) or the previous tracking result (for subsequent frames) as the follows: for i-th patch, if it belongs to the shrunk region of the bounding box then q_i is 1, and if it belongs to the expanded region of the bounding box then q_i is 0. Fig. 1 shows the details. Although we adopt this simple initialization strategy, the promising results on standard benchmarks in our experiments have demonstrated its effectiveness. The remaining patches are non-determined, and will be diffused by other patches. To this end, we define an indication vector Γ that $\Gamma_i = 1$ indicates the i-th patch is foreground or background patch, and $\Gamma_i = 0$ denotes the i-th patch is non-determined patch. We integrate the patch weights into Eq. (1), and obtain

Algorithm 1 Optimization Procedure to Eq. (3)

Input: The patch feature matrix $\mathbf{X}$ and the initial weight vector $\mathbf{q}$, the parameters α, λ, β, γ and ξ;
Set $\mathbf{Z}_0 = \mathbf{P}_0 = \mathbf{Q}_0 = \mathbf{Y}_{2,0} = \mathbf{Y}_{3,0} = \mathbf{0}$, $\mathbf{E}_0 = \mathbf{Y}_{1,0} = \mathbf{0}$, $\mathbf{w} = \mathbf{1}$, $\eta = 0.01 * ||\mathbf{X}||_F^2$, $\mu_0 = 0.1$, $\mu_{max} = 10^{10}$, $\rho = 1.1$, $\varepsilon_1 = 10^{-6}$, $maxIter = 35$, and $k = 0$.
Output: $\mathbf{Z}$, $\mathbf{E}$ and $\mathbf{w}$.

while not converged **do**
  Update $\mathbf{Z}_{k+1}$, $\mathbf{P}_{k+1}$ and $\mathbf{Q}_{k+1}$ by Eq. (6);
  Update $\mathbf{E}_{k+1}$ by APG algorithm;
  Update $\mathbf{w}_{k+1}$ by solving Eq. (6);
  Update Lagrangian multipliers as follows: $\mathbf{Y}_{1,k+1} = \mathbf{Y}_{1,k} + \mu_k(\mathbf{W}_{k+1} \circ (\mathbf{X} - \mathbf{XZ}_{k+1} - \mathbf{E}_{k+1}))$,
  $\mathbf{Y}_{2,k+1} = \mathbf{Y}_{2,k} + \mu_k(\mathbf{Z}_{k+1} - \mathbf{P}_{k+1}),\ \mathbf{Y}_{3,k+1} = \mathbf{Y}_{3,k} + \mu_k(\mathbf{Z}_{k+1} - \mathbf{Q}_{k+1})$;
  Update μ_{k+1} by $\mu_{k+1} = \min(\mu_{max}, \rho\mu_k)$;
  Update k by $k = k + 1$;
  Check the convergence condition: the maximum element changes of $\mathbf{Z}$, $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{E}$ and $\mathbf{w}$ between two consecutive iterations are less than ε_1 or the maximum number of iterations reaches $maxIter$.
end while

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E},\mathbf{w}} rank(\mathbf{Z}) + \alpha||\mathbf{Z}||_0 + \lambda||\mathbf{E}||_{2,0} + \beta\sum_{i,j}\mathbf{Z}_{ij}(\mathbf{w}_i - \mathbf{w}_j)^2 \\ &+ \frac{\gamma}{2}||\Gamma \circ (\mathbf{w} - \mathbf{q})||^2 + \frac{\xi}{2}||\mathbf{w}||^2, \\ &s.t.\ \mathbf{W} \circ \mathbf{X} = \mathbf{W} \circ (\mathbf{XZ} + \mathbf{E}),\ \mathbf{Z} \geq 0,\ \mathbf{w} \geq 0, \end{aligned} \tag{2}$$

where $\mathbf{W} = [\mathbf{w}^T; \mathbf{w}^T; ...; \mathbf{w}^T] \in \mathbb{R}^{d\times n}$, and $\circ$ indicates the element-wise product. β, γ and ξ are the balanced parameters. The fourth and fifth terms are the smoothness constraint and the fitting constraint, respectively. Since the indication vector Γ removes fitness constraint of non-determined patch weights, we introduce the last term to avoid overfitting. In Eq. 2, $\mathbf{Z}$ indicates the graph affinity matrix, and larger $\mathbf{Z}_{ij}$ will encourage $\mathbf{w}_i$ is closer to $\mathbf{w}_j$ by minimizing the fourth term.

Optimization

Due to the non-convexity of the $rank$ function and the l_0 norm, it is difficult to directly minimize Eq. (2). To overcome these obstacles, we will use convex surrogates for all the non-convex low rank and sparsity terms. Through convex relaxation, we replace $rank$ function and the l_0 norm with the nuclear norm and the l_1 norm, respectively. Thus, Eq. (2) can be relaxed as:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E},\mathbf{w}} ||\mathbf{Z}||_* + \alpha||\mathbf{Z}||_1 + \lambda||\mathbf{E}||_{2,1} + \beta\sum_{i,j}\mathbf{Z}_{ij}(\mathbf{w}_i - \mathbf{w}_j)^2 \\ &+ \frac{\gamma}{2}||\Gamma \circ (\mathbf{w} - \mathbf{q})||^2 + \frac{\xi}{2}||\mathbf{w}||^2, \\ &s.t.\ \mathbf{W} \circ \mathbf{X} = \mathbf{W} \circ (\mathbf{XZ} + \mathbf{E}),\ \mathbf{Z} \geq 0,\ \mathbf{w} \geq 0, \end{aligned} \tag{3}$$

where $||\cdot||_*$, $||\cdot||_1$, $||\cdot||_{2,1}$ are the nuclear norm, the l_1 norm and the $l_{2,1}$ norm, respectively. Next, we present an efficient algorithm to solve Eq. (3).

We first use the linearized ADM with adaptive penalty (LADMAP) (Lin, Liu, and Su 2011) to avoid some matrix inversions in optimization. Two auxiliary variables $\mathbf{P} \in \mathbb{R}^{n\times n}$ and $\mathbf{Q} \in \mathbb{R}^{n\times n}$ are introduced to make Eq. (3) separable:

$$\begin{aligned}&\min_{\mathbf{Z},\mathbf{E},\mathbf{w},\mathbf{P},\mathbf{Q}} ||\mathbf{Z}||_* + \alpha||\mathbf{P}||_1 + \lambda||\mathbf{E}||_{2,1} \\ &+ \beta\sum_{i,j}\mathbf{Q}_{ij}(\mathbf{w}_i - \mathbf{w}_j)^2 + \frac{\gamma}{2}||\Gamma\circ(\mathbf{w}-\mathbf{q})||^2 + \frac{\xi}{2}||\mathbf{w}||^2, \\ &s.t.\ \mathbf{W}\circ\mathbf{X} = \mathbf{W}\circ(\mathbf{XZ}+\mathbf{E}),\ \mathbf{Z}=\mathbf{P},\ \mathbf{Z}=\mathbf{Q}, \mathbf{Q}\geq 0, \\ &\mathbf{w}\geq 0,\end{aligned} \tag{4}$$

The augmented Lagrangian function of (4) is

$$\begin{aligned}&L(\mathbf{Z},\mathbf{P},\mathbf{Q},\mathbf{E},\mathbf{w}) \\ &= ||\mathbf{Z}||_* + \alpha||\mathbf{P}||_1 + \lambda||\mathbf{E}||_{2,1} + \beta\sum_{i,j}\mathbf{Q}_{ij}(\mathbf{w}_i-\mathbf{w}_j)^2 \\ &+ \frac{\gamma}{2}||\Gamma\circ(\mathbf{w}-\mathbf{q})||^2 + \frac{\xi}{2}||\mathbf{w}||^2 \\ &+ f(\mathbf{Z},\mathbf{P},\mathbf{Q},\mathbf{E},\mathbf{w},\mathbf{Y}_1,\mathbf{Y}_2,\mathbf{Y}_3,\mu) - \frac{1}{2\mu}(||\mathbf{Y}_1||_F^2 \\ &+ ||\mathbf{Y}_2||_F^2 + ||\mathbf{Y}_3||_F^2),\end{aligned} \tag{5}$$

with $\mathbf{Q}\geq 0$, and $\mathbf{w}\geq 0$. $\mu > 0$ is the penalty parameter, and $f(\mathbf{Z},\mathbf{P},\mathbf{Q},\mathbf{E},\mathbf{w},\mathbf{Y}_1,\mathbf{Y}_2,\mathbf{Y}_3,\mu) = \frac{\mu}{2}(||\mathbf{W}\circ(\mathbf{X}-\mathbf{XZ}-\mathbf{E})+\mathbf{Y}_1/\mu||_F^2 + ||\mathbf{Z}-\mathbf{P}+\mathbf{Y}_2/\mu||_F^2 + ||\mathbf{Z}-\mathbf{Q}+\mathbf{Y}_3/\mu||_F^2)$. $\mathbf{Y}_1$, $\mathbf{Y}_2$ and $\mathbf{Y}_3$ are the Lagrangian multipliers. LADMDP alternatively updates one variable by minimizing L with fixing other variables. With simple algebra, the updating schemes of $(k+1)$-th iteration are as follows,

$$\begin{aligned}&\mathbf{Z}_{k+1} = \arg\min_{\mathbf{Z}}\ ||\mathbf{Z}||_* + \frac{\eta\mu_k}{2}||\mathbf{Z}-\mathbf{Z}_k||_F^2 + \langle\nabla_{\mathbf{Z}}f_k, \mathbf{Z}-\mathbf{Z}_k\rangle, \\ &\mathbf{P}_{k+1} = \arg\min_{\mathbf{P}}\ \alpha||\mathbf{P}||_1 + \frac{\mu_k}{2}||\mathbf{Z}_{k+1}-\mathbf{P}+\mathbf{Y}_{2,k}/\mu_k||_F^2, \\ &\mathbf{Q}_{k+1} = [\mathbf{Z}_{k+1} + (\mathbf{Y}_{3,k}-\beta\mathbf{W}'_k)/\mu_k]_+, \\ &\mathbf{E}_{k+1} = \arg\min_{\mathbf{E}}\ \lambda||\mathbf{E}||_{2,1} + \frac{\mu_k}{2}||\mathbf{W}_k\circ(\mathbf{X}-\mathbf{XZ}_{k+1}-\mathbf{E}) \\ &+ \mathbf{Y}_{1,k}/\mu_k||_F^2, \\ &\mathbf{w}_{k+1} = [(2\beta(\mathbf{D}_{k+1}-\mathbf{Q}_{k+1}-\mathbf{Q}_{k+1}^T) + \gamma\Gamma' + \xi\mathbf{I} \\ &+ \mu_k\mathbf{D}_{1,k+1})^{-1}(\gamma\Gamma\circ\mathbf{q} - \mu_k\mathbf{d}_{2,k+1})]_+,\end{aligned} \tag{6}$$

where $\nabla_{\mathbf{Z}}f$ is the partial derivative of f with respect to $\mathbf{Z}$, and $\eta = 0.01 * ||\mathbf{X}||_F^2$. f_k is the abbreviation of $f(\mathbf{Z}_k,\mathbf{P}_k,\mathbf{Q}_k,\mathbf{E}_k,\mathbf{w}_k,\mathbf{Y}_{1,k},\mathbf{Y}_{2,k},\mathbf{Y}_{3,k},\mu_k)$. The operator $[\mathbf{u}]_+$ turns negative elements in $\mathbf{u}$ to 0 while keeps the rest. $\mathbf{I}$ is the identity matrix, and $\mathbf{W}'$ is the matrix with the element $\mathbf{W}'_{ij} = (\mathbf{w}_i - \mathbf{w}_j)^2$, and $\mathbf{D}$ is the degree matrix of $(\mathbf{Q}+\mathbf{Q}^T)$ that $\mathbf{D} = \mathrm{diag}\{d_{11}, d_{22}, ..., d_{nn}\}$, where $d_{ii} = \sum_j(\mathbf{Q}_{ij}+\mathbf{Q}_{ji})$. Similarly, $\mathbf{D}_1$ is the degree matrix of $(\mathbf{X}-\mathbf{XZ}-\mathbf{E})^T\circ(\mathbf{X}-\mathbf{XZ}-\mathbf{E})^T$. $\mathbf{d}_2$ is the vector that its i-th element equals to summing all elements of i-th column in $(\mathbf{X}-\mathbf{XZ}-\mathbf{E})\circ(\mathbf{Y}_1/\mu)$, and $\Gamma' = \mathrm{diag}\{\Gamma_1,\Gamma_2,...,\Gamma_n\}$. We present the complete derivations of above subproblems in the **supplementary file**.

Since each subproblem of Eq. (6) is convex, we can guarantee that the solution by our algorithm satisfies the Nash equilibrium conditions (Xu and Yin 2013). Alg. 1 summarizes the optimization procedure. Note that: i) $\mathbf{P}_{k+1}$ can be solved by the soft-thresholding (or shrinkage) method (Liu et al. 2013) with closed-form solution. ii) The subproblem of $\mathbf{E}_{k+1}$ does not have a closed-form solution, and we employ the accelerated proximal gradient (APG) algorithm (Parikh and Boyd 2014) to solve it. iii) The solution of $\mathbf{Z}_{k+1}$ can be obtained by the singular value thresholding (SVT) method (Liu et al. 2013), which involves one SVD operation at each iteration, and thus also suffers from high computational cost. Therefore, we employ the randomized SVT method (Oh et al. 2015) to approximate the solution of original SVT while preserving its accuracy.

Structured Output Tracking

In this section, we incorporate the optimized patch weights into the conventional tracking-by-detection algorithm, Struck (Hare, Saffari, and Torr 2011). Struck selects the optimal target bounding box $\mathbf{c}_t^*$ in the t-th frame by maximizing a classification score:

$$\mathbf{c}_t^* = \arg\max_{\mathbf{c}} \langle\mathbf{h}_{t-1}, \mathbf{x}_{t,\mathbf{c}}\rangle, \tag{7}$$

where $\mathbf{h}_{t-1}$ is the normal vector of a decision plane of $(t-1)$-th frame, and $\mathbf{x}_{t,\mathbf{c}} = [\mathbf{x}_{t,1}; \mathbf{x}_{t,2}; ...; \mathbf{x}_{t,n}]$ denotes the descriptor representing a bounding box $\mathbf{c}$ in t-th frame. Instead of using binary-labeled sample, Struck employs structured sample that consists of a target bounding box and nearby boxes in the same frame to prevent the labelling ambiguity in training the classifier. Specifically, it constrains that the confidence score of an target bounding box is larger than that of a nearby box by a margin determined by the overlap ratio between two boxes. By this way, Struck can reduce adverse effects of false labelling.

Given the bounding box of the target object in previous frame $t-1$, we first set a searching window in current frame t. For i-th candidate bounding box within the searching window, we weight its patch feature descriptor $\mathbf{x}_{t,i}$ by the weight $\mathbf{a}_{t-1,i} = 1/(1+\exp(-\sigma\hat{\mathbf{w}}_{t-1,i}))$, and concatenate them into a vector as the feature representation: $\hat{\mathbf{x}} = [\mathbf{a}_{t-1,1}\mathbf{x}_{t,1}; \mathbf{a}_{t-1,2}\mathbf{x}_{t,2}; ...; \mathbf{a}_{t-1,n}\mathbf{x}_{t,n}]$. Herein, $\hat{\mathbf{w}}$ is the normalized vector of $\mathbf{w}$, and the parameter σ is fixed to be 35 in this work. The optimal bounding box $\mathbf{c}_t^*$ can be selected to update the object location by maximizing the classification score:

$$\mathbf{c}_t^* = \arg\max_{\mathbf{c}} (\omega\langle\mathbf{h}_{t-1}, \hat{\mathbf{x}}_{t,\mathbf{c}}\rangle + (1-\omega)\langle\mathbf{h}_0, \hat{\mathbf{x}}_{t,\mathbf{c}}\rangle), \tag{8}$$

where $\mathbf{h}_0$ is learnt in initial frame, which can prevent it from learning drastic appearance changes, and ω is a balance parameter. Given the tracked bounding box $\mathbf{c}_t^*$, we compute the patch weights $\mathbf{a}_t$ by Eq. (3), and then update the classifier $\mathbf{h}_t$. To prevent the effects of unreliable tracking results, we update the classifier only when the confidence score of tracking result is larger than a threshold θ. In this paper, the confidence score of tracking result in t-th frame is defined as the average similarity between the weighted descriptor of the tracked bounding box and the positive support vectors: $\frac{1}{|\mathbb{S}_t|}\sum_{\mathbf{s}\in\mathbb{S}_t}\langle\mathbf{s}, \hat{\mathbf{x}}_{t,\mathbf{c}_t^*}\rangle$, where $\mathbb{S}_t$ is the set of the positive support vectors at time t.

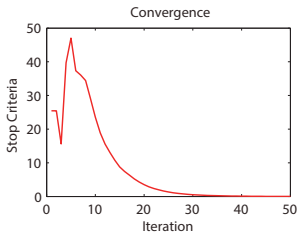

Figure 2: Convergence of the proposed algorithm on all video sequences of OTB100.

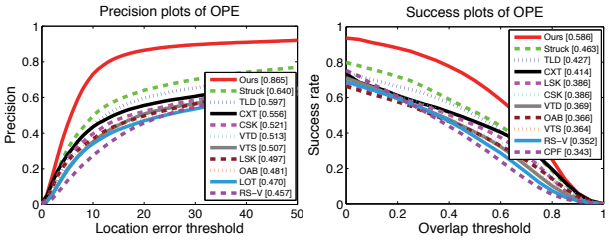

Figure 3: Evaluation results on OTB100 with 9 conventional trackers. The representative score of PR/SR is presented in the legend. OPE denotes the one-pass evaluation.

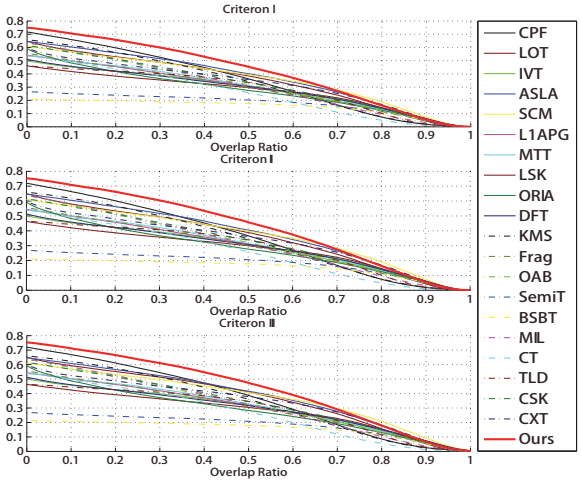

Figure 4: TRR curves on NUS-PRO, where twenty trackers are shown here.

Difference with Related Work

It should be noted that the proposed tracking algorithm is significantly different from the recently proposed approaches that use low-rank or sparse representation for object tracking (Zhong, Lu, and Yang 2012; Zhang et al. 2014). In these methods, reconstruction errors or representation coefficients are usually adopted to compute the confidence of candidates in the Bayesian filtering framework. While we employ the low-rank and sparse representation to learn a dynamic graph for each frame, in which the node weights are used to suppress the effects of background information in the tracking-by-detection framework.

In addition, our approach is also significantly different from the recently proposed tracker, SOWP (Kim et al. 2015), in several aspects. First, it learns a dynamic graph for each frame by considering the nonnegativeness, sparsity, and low-rank constraints to suppress the effects of noises and/or corruptions of low-level features in computing edges. Second, it optimizes the edges and the node weights in a joint fashion, while SOWP first computes the edge weights and then the node weights. Third, it simultaneously integrates the initial foreground and background information into an unified model, while SOWP requires two calculations on its model to obtain the final patch weights, one for foreground and another for background. Fourth, it designs an efficient algorithm to optimize the proposed model, and achieves comparable efficiency with SOWP. Finally, it improves the Struck tracking algorithm by considering the initial classifier in the current classification process, which can prevent it from learning drastic appearance changes.

Experiments

The experiments are carried out on a PC with an Intel i7 4.0GHz CPU and 32GB RAM, and implemented in C++. The proposed tracker performs at about 10 frames per second, and the convergence curve of the proposed algorithm is presented in Fig. 2.

Evaluation Settings

Parameters. For fair comparisons, we fix all parameters and other settings in experiments. In Eq. (2), we empirically set $\{\alpha, \lambda, \beta, \gamma, \xi\} = \{1, 0.1, 5, 18, 1\}$. In Struck, we empirically set $\{\omega, \theta\} = \{0.67, 0.2\}$. Besides, we partition all bounding box into 64 non-overlapping patches to balance accuracy-efficiency trade-off, and extract RGB and gradient histograms for each patch, where the dimensions of gradient and each color channel is set to be 8. To improve efficiency, each frame is scaled so that the minimum side length of a bounding box is 32 pixels, and the side length of a searching window is fixed to be $2\sqrt{WH}$, where W and H are the width and height of the scaled bounding box, respectively.

It is worth noting that our approach is insensitive to parameters. When we slightly adjust some parameters, tracking performance only changes a little, e.g., setting β as 2 and setting ξ as 2, we find the results are slightly different from the current settings ($\beta = 2$ obtains 0.7% lower in PR, 0.1% better in SR, and $\xi = 2$ obtains 0.1% lower in both PR and SR). Although Eq. (3) involves many parameters, we can easily tune them as some of them are correlated, e.g., empirically $\gamma = 1.6\beta$.

OTB100 benchmark. We evaluate the proposed tracking method on the OTB100 benchmark dataset (Wu, Lim, and Yang 2015). OTB100 is a large dataset, which includes 100 image sequences with ground-truth object locations, and is the extension of OTB50 (Wu, Lim, and Yang 2013). We employ precision rate (PR) and success rate (SR) to measure the quantitative performance of various trackers.

NUS-PRO challenge. We also compare our approach with other tracking approaches on another large-scale benchmark

Table 1: Attribute-based PR/SR scores on OTB100 benchmark (Wu, Lim, and Yang 2015) compared with recent 8 leading trackers. The best and the second best results are in red and green colors, respectively.

	HCF	SOWP	MEEM	MUSTer	KCF	LCT	DSST	DLT	Ours
IV	0.817/0.540	0.777/0.554	0.740/0.517	0.782/0.600	0.708/0.474	0.746/0.566	0.723/0.489	0.522/0.408	0.838/0.573
SV	0.802/0.488	0.750/0.478	0.740/0.474	0.715/0.518	0.639/0.399	0.686/0.492	0.667/0.413	0.542/0.399	0.813/0.504
OCC	0.767/0.525	0.754/0.528	0.741/0.504	0.734/0.554	0.622/0.438	0.682/0.507	0.615/0.426	0.454/0.335	0.820/0.562
DEF	0.791/0.530	0.741/0.527	0.754/0.489	0.689/0.524	0.617/0.436	0.689/0.499	0.568/0.412	0.451/0.295	0.857/0.582
MB	0.797/0.573	0.710/0.557	0.722/0.545	0.699/0.557	0.617/0.456	0.673/0.532	0.636/0.465	0.427/0.353	0.815/0.591
FM	0.797/0.555	0.719/0.542	0.735/0.529	0.691/0.539	0.628/0.455	0.675/0.527	0.602/0.440	0.426/0.345	0.777/0.549
IPR	0.854/0.559	0.828/0.567	0.794/0.529	0.773/0.551	0.693/0.465	0.782/0.557	0.724/0.485	0.471/0.348	0.856/0.573
OPR	0.810/0.537	0.790/0.549	0.798/0.528	0.748/0.541	0.675/0.454	0.750/0.541	0.675/0.453	0.517/0.376	0.855/0.577
OV	0.677/0.474	0.633/0.497	0.685/0.488	0.591/0.469	0.498/0.393	0.558/0.452	0.487/0.374	0.558/0.384	0.753/0.533
BC	0.847/0.587	0.781/0.575	0.752/0.523	0.786/0.579	0.716/0.498	0.740/0.553	0.708/0.481	0.509/0.373	0.867/0.614
LR	0.787/0.424	0.713/0.416	0.605/0.355	0.677/0.477	0.545/0.306	0.490/0.330	0.595/0.311	0.615/0.422	0.732/0.417
All	0.837/0.562	0.803/0.560	0.781/0.530	0.774/0.577	0.692/0.475	0.762/0.562	0.695/0.475	0.526/0.384	0.865/0.586

dataset, NUS-PRO (Li et al. 2016a). This large-scale database contains 365 challenging image sequences of pedestrians and rigid objects, and most of them are captured from moving cameras. Each sequence is annotated target location and occlusion level for evaluation. We employ the threshold-response relationship (TRR) with three criteria (Criterion I, II and III) of occlusion annotations on entire dataset to evaluate our method.

Comparison Results

We first present the evaluation results on OTB100 against 9 conventional trackers (Wu, Lim, and Yang 2015) in Fig. 3. The comparison curves show that our tracker significantly outperforms the Struck tracker, achieving 35.2% gain in PR and 26.6% gain in SR over Struck.

Tab. 1 presents the attribute-based comparison results of our tracker with recent 8 leading trackers on OTB100, including HCF (Ma et al. 2015a), SOWP (Kim et al. 2015), MEEM (Zhang, Ma, and Sclaroff 2014), MUSTer (Hong et al. 2015), KCF (Henriques et al. 2015), LCT (Ma et al. 2015b), DSST (Danelljan et al. 2014) and DLT (Wang and Yeung 2013). The superior results over other methods demonstrate the effectiveness of the proposed approach in handling sequences with IV, SV, OCC, DEF, MB, IPR, OPR, OV and BC. Moreover, for FM, we achieve comparable results against HCF. However, unsatisfying results are usually generated in low-resolution video sequences. It may attribute to the weakness of our used features (color and gradient) in representing the target object with less appearance information. In addition, the qualitative comparisons and analysis of our approach with several typical trackers are presented in the **supplementary file** due to space limitation.

For comprehensive comparisons, we further evaluate our method on NUS-PRO against 20 conventional trackers (Li et al. 2016a) in Fig. 4. We can see that the proposed tracker also achieves superior performance than other trackers. The results of the top 3 performing methods (ASLA (Jia, Lu, and Yang 2012), SCM (Zhong, Lu, and Yang 2012) and LOT (Oron et al. 2012)) show that the combination of the local feature representation and the particle filter framework can obtain promising tracking performance. Although adopting the local feature representation only, we achieve the best performance on NUS-PRO.

Table 2: The performance of three versions of the proposed method.

	Ours	Ours-noW	Ours-8nG	Ours-ovF
PR	0.865	0.795	0.829	0.840
SR	0.586	0.558	0.569	0.574

Component Analysis

To justify the significance of the main components using OTB100, we implement three versions of our approach for empirical analysis. The three versions are: 1) Ours-noW, that removes the patch weights in our tracking algorithm. 2) Ours-8nG, that substitutes the dynamic graph by an 8-neighbor graph to compute the patch weights. 3) Ours-OvF, that removes the last term in Eq. (2). Tab. 2 presents the evaluation results. It can be seen that the performance achieved by our versions demonstrate the significance of the main components. Furthermore, introducing patch weights into the tracking algorithm benefits to suppress the effects of background by observing Ours and Ours-8nG superior to Ours-noW. In addition, Ours outperforms Ours-8nG, which suggests that the dynamic graph is beneficial to optimize the patch weights. Finally, Ours outperforms Ours-ovF, justifying the effectiveness of avoiding overfitting in Eq. (3).

Conclusion

In this paper, we have proposed an effective approach for visual tracking by suppressing the effects of background information. A patch-based graph has been learnt dynamically by capturing the global structure and local linear relationship among patches. To reduce the computational complexities, we have presented an efficient algorithm for the proposed model by solving several convex subproblems. Finally, the optimized patch weights are incorporated into Struck tracker to carry out object tracking. In future work, we will replace hand-crafted features with deep feature learning for more robust object representation, and integrate fast feature pyramids into our framework to achieve scale adaptation without increasing much computational cost.

References

Babenko, B.; Yang, M.-H.; and Belongie, S. 2011. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(7):1619–1632.

Cheng, B.; Liu, G.; Wang, J.; Huang, Z.; and Yan, S. 2011. Multi-task low-rank affinity pursuit for image segmentation. In *ICCV*.

Comaniciu, D.; Ramesh, V.; and Meer, P. 2003. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Danelljan, M.; Hager, G.; Khan, F.; and Felsberg, M. 2014. Accurate scale estimation for robust visual tracking. In *BMVC*.

Duffner, S., and Garcia, C. 2013. Pixeltrack: A fast adaptive algorithm for tracking non-rigid objects. In *ICCV*.

Grabner, H.; Grabner, M.; and Bischof, H. 2008. Semi-supervised on-line boosting for robust tracking. In *ECCV*.

Hare, S.; Saffari, A.; and Torr, P. H. S. 2011. Struck: Structured output tracking with kernels. In *ICCV*.

He, S.; Yang, Q.; Lau, R.; Wang, J.; and Yang, M.-H. 2013. Visual tracking via locality sensitive histograms. In *CVPR*.

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

Hong, Z.; Chen, Z.; Wang, C.; Mei, X.; Prokhorov, D.; and Tao, D. 2015. MUlti-Store Tracker (MUSTer): a cognitive psychology inspired approach to object tracking. In *CVPR*.

Jia, X.; Lu, H.; and Yang, M.-H. 2012. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*.

Kim, H.-U.; Lee, D.-Y.; Sim, J.-Y.; and Kim, C.-S. 2015. Sowp: Spatially ordered and weighted patch descriptor for visual tracking. In *ICCV*.

Li, A.; Li, M.; Wu, Y.; Yang, M.-H.; and Yan, S. 2016a. Nus-pro: A new visual tracking challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(2):335–349.

Li, C.; Lin, L.; Zuo, W.; Wang, W.; and Tang, J. 2016b. An approach to streaming video segmentation with sub-optimal low-rank decomposition. *IEEE Transactions on Image Processing* 25(5):1947–1960.

Lin, Z.; Liu, R.; and Su, Z. 2011. Linearized alternating direction method with adaptive penalty for low rank representation. In *NIPS*.

Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; and Ma, Y. 2013. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(1):171–184.

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*.

Oh, T.-H.; Matsushita, Y.; Tai, Y.-W.; and Kweon, I. S. 2015. Fast randomized singular value thresholding for nuclear norm minimization. In *CVPR*.

Oron, S.; Bar-Hillel, A.; Levi, D.; and Avidan, S. 2012. Locally orderless tracking. In *CVPR*.

Parikh, N., and Boyd, S. 2014. Proximal algorithms. *Foundations and Trends in Optimization* 1(3):123–231.

Tsochantaridis, I.; Joachims, T.; Hofmann, T.; and Altun, Y. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* 6:1453–1484.

Wang, N., and Yeung, D.-Y. 2013. Learning a deep compact image representation for visual tracking. In *NIPS*.

Wright, J.; Ma, Y.; Mairal, J.; Sapiro, G.; Huang, T. S.; and Yan, S. 2010. Sparse representation for computer vision and pattern recognition. *Proceedings of IEEE* 98(6):1031–1044.

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

Wu, Y.; Lim, J.; and Yang, M.-H. 2015. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xu, Y., and Yin, W. 2013. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences* 6(3):1758–1789.

Yan, S., and Wang, H. 2009. Semi-supervised learning by sparse representation. In *ICDM*.

Yang, C.; Zhang, L.; Lu, H.; Ruan, X.; and Yang, M.-H. 2013. Saliency detection via graph-based manifold ranking. In *CVPR*.

Yang, Y.; Feng, J.; Yang, J.; and Huang, T. S. 2015. Learning with l^0-graph: l^0-induced sparse subspace clustering. *arXiv:1510.08520*.

Yang, F.; Lu, H.; and Yang, M.-H. 2014. Robust superpixel tracking. *IEEE Transactions on Image Processing* 23(4):1639–1651.

Zhang, T.; Liu, S.; Ahuja, N.; Yang, M.-H.; and Ghanem, B. 2014. Robust visual tracking via consistent low-rank sparse learning. *International Journal of Computer Vision* 111(2):171–290.

Zhang, J.; Ma, S.; and Sclaroff, S. 2014. MEEM: robust tracking via multiple experts using entropy minimization. In *ECCV*.

Zhong, W.; Lu, H.; and Yang, M.-H. 2012. Robust object tracking via sparsity-based collaborative model. In *CVPR*.

Zhuang, L.; Gao, H.; Lin, Z.; Ma, Y.; Zhang, X.; and Yu, N. 2012. Non-negative low rank and sparse graph for semi-supervised learning. In *CVPR*.