Robust Tracking via Weakly Supervised Ranking SVM∗

Yancheng Bai and Ming Tang
National Lab of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing 100190, China

Abstract

Appearance model is a key component of tracking algorithms. Most existing approaches utilize the object information contained in the current and previous frames to construct the object appearance model and locate the object with the model in frame $t+1$. This method may work well if the object appearance just fluctuates in short time intervals. Nevertheless, suboptimal locations will be generated in frame $t+1$ if the visual appearance changes substantially from the model. Then, continuous changes would accumulate errors and finally result in a tracking failure. To copy with this problem, in this paper we propose a novel algorithm - online Laplacian ranking support vector tracker (LRSVT) - to robustly locate the object. The LRSVT incorporates the labeled information of the object in the initial and the latest frames to resist the occlusion and adapt to the fluctuation of the visual appearance, and the weakly labeled information from frame $t+1$ to adapt to substantial changes of the appearance. Extensive experiments on public benchmark sequences show the superior performance of LRSVT over some state-of-the-art tracking algorithms.

1. Introduction

Visual object tracking is an important problem in computer vision and has many applications including traffic monitoring, augmented reality and human computer interface (HCI), just to name a few. Although it has been investigated in the past decades, designing a robust tracker to cope with different objects under various situations is still a great challenging task. A very common difficulty is to resist the visual appearance changing frame by frame due to 3D rotation, sudden illumination changing and partial occlusion[34]. Such changes may make a tracker drift away from the target object.

Generally speaking, a typical tracking algorithm is composed of three parts: image representation, the appearance model and the dynamic model [2]. Among them, the appearance model plays a crucial role [2, 25]. In this paper, we will focus on the model-free tracking problem, i.e., no prior knowledge except for the object location is known at the beginning of tracking.

To adapt to the appearance changes of the object during tracking, perhaps a simplest method is to construct and update the appearance model only using the the current frame [8, 14]. As a representative algorithm, the mean shift based tracker [8] represents the appearance model of the object as a weighted feature histogram in the current frame, and locate the object through the maximization of the Bhattacharyya coefficient in the next frame. It is obvious that such a tracker may easily drift away from the target object due to occlusion or substantial changes of the appearance. Therefore, some prior frames are used to model the object appearance to make tracking more robust [2, 12, 25, 13, 26, 23, 19, 20]. Ross et al. [25] incrementally learned a low-dimensional subspace representation of the object frame by frame, efficiently adapting to appearance changes. Babenko et al. [2] learned a classifier as the appearance model via multiple instance boosting [29]. The weak classifiers were online updated by means of a forgetting factor. Mei and Ling [23] constructed the appearance model by means of linear and sparse combination of target templates and trivial ones. The template set was dynamically updated according to the similarity between the tracking result and the template set. More recently, Visual Tracking Decomposition (VTD) [19] proposed by J. Kwon and K. M. Lee used the initial frame and the latest four frames to learn the appearance model by means of sparse principal component analysis (SPCA) of a set of feature templates. By introducing the local sparse appearance model [30], Liu et al. [20] extended the mean shift based tracking algorithm [8] and modeled the appearance in terms of a static sparse dictionary of the object in the initial frame and an online updated histogram of the current frame. In order to overcome the drifting problem during online adaptation of the object model, Grabner et al. [13] formulated the tracking problem as a semi-supervised one, where only the object bounding box in the first frame was considered.

∗This work was supported by Natural Science Foundation of China. Grant Nos. 60835004 and 60572057. Corresponding author: tangm@nlpr.ia.ac.cn
as labeled, and all subsequent tracking results were left unlabeled. However, it was hard for this approach to handle large changes of the object appearance away from the initial frame. In [35], Zeisl et al. proposed an online semi-supervised MILBoost tracker to combine the adaptivity of multiple instance tracking [2] and robustness against drifting of semi-supervised learning based tracking [13].

An implicit assumption in all aforementioned approaches is that the appearance models which are trained till frame $t$ can still work well in frame $t+1$. This assumption means that the appearance of the object can only fluctuate, rather than substantially change, in short time intervals. This, however, may not definitely hold because the appearance in the new frame may change greatly from the recently previous frames, and the model may not catch up with the changes. If the assumption is violated, the appearance model will not work well in frame $t+1$. Moreover, tracking errors will then be accumulated and finally result in the tracker drifting away from the target.

To deal with the above problem, Yang et al. [32] incorporated the positive and negative samples in the next frame to model the object appearance till the current frame, and then located the object with the model in the next frame. They assumed that the appearance manifold during a short time interval is linear, therefore, the appearance model had to be updated only with the least several frames. This may result in the model drift problem in the case of occlusion.

In this paper, we formulate the tracking process as a weakly supervised ranking problem to incorporate the information of the object in the next frame. A key idea of our approach is that the target should be ranked higher than others around it. As pointed out by Hare et al. [27], it is hard to assign a class label to a sample (patch) in tracking setting. Whereas, the relative relation between patches is easily figured out. Therefore, we extend the learning to rank algorithm, ranking SVM [17], to learn the relative relation. Another key idea of our algorithm is that a simple and effective tracker is employed to provide rough locations of the target object, i.e., some weakly labeled samples, in the next frame. In our current implementation, the KLT [22] is employed as such a weak labeler. It is probable that patches at the rough locations would be ranked weakly higher than patches around them. It is hopeful that these two sets of patches will reflect different changes, rather than just fluctuations, of the target appearance in the new frame. Based on the above considerations, we propose a weakly supervised ranking SVM algorithm based on the smoothness assumption [37] and the manifold regularization [4]. The overview of the proposed algorithm, online Laplacian ranking support vector tracking (LRSVT), is illustrated in Fig. 1. The labeled higher-ranked samples are composed of patches very close to the ground truth of the initial frame and those very close to the object locations in several most recent frames, and the labeled lower-ranked samples are those around the labeled higher-ranked ones. The weakly labeled higher-ranked samples are composed of patches close to that labeled as object patch by the weak labeler in the new frame. These three sample sets are used to train the novel weakly supervised ranking SVM (Problem (7)). The red patch in Step 3 is the highest scored by $F(x)$, therefore, is accepted as the object location in the new frame. This figure is best viewed in color.
adapt to fluctuation of the object appearance, and incorporates the weakly labeled information in the next frame to adapt to the substantially different changes of the object appearance. Extensive experiments have shown that the LRSVT can outperform some state-of-the-art tracking algorithms.

The rest of this paper is organized as follows. In Sec.2, we briefly overview the related work. In Sec.3, the novel Laplacian ranking SVM is derived. Then the novel tracking algorithm, online Laplacian Ranking Support Vector Tracker (LRSVT), is described in detail in Sec.4. Experimental results and comparison with other state-of-the-art approaches are presented in Sec.5. Sec.6 summarizes our work.

2. Related Work

The topic of learning to rank, which combines relevance problems with prediction problems, has recently attracted considerable attention in machine learning community [21, 11], and a great many of ranking algorithms have been proposed [15, 10, 6, 31]. The main goal of learning to rank is to automatically construct a ranking model based on the partial order of training data.

In the computer vision domain, learning to rank is mainly used in image and video retrieval. For image retrieval, Huang et al. [16] proposed a transductive learning framework, in which the task of image search is casted as the problem of hypergraph ranking. To rank large scale images and video collections, Merler et al. [24] proposed the imbalanced RankBoost which merged RankBoost [10] and iterative thresholding into a unified loss optimization framework.

In recent years, learning to rank has begun to apply to other areas of computer vision. Yang et al [33] developed a variation of the RankBoost algorithm with L1 regularization for facial expression recognition and intensity estimation. Zhang et al. [36] proposed a two stage cascaded ranking SVMs detector to accelerate the detection process without decreasing accuracy.

As a powerful machine learning technique, semi-supervised learning [37] has been applied to cope with the visual object tracking problem. Grabner et al. [13] formulated the tracking problem as a semi-supervised one to overcome the model drift problem caused by serious occlusion. Trying to combine the adaptivity of the multiple-instance based tracker [2] on updating the appearance model and the robustness of the semi-supervised learning based tracker [13] against occlusions, Zeisl et al. [35] proposed an online semi-supervised multiple instance boosting based tracker. Nevertheless, both the above tracking algorithms were based on the tracking-by-detection and only utilized current or past information of the object.

Kalal et al. [18] proposed a robust detector/tracker, TLD (tracking, learning, detection), which combined the random forest [5] based detector and KLT tracker [22]. In their approach, KLT provided positive and negative training samples for updating the random forest based detector. Therefore, TLD was based on tracking-by-detector and supervised learning. And like most of existing tracking algorithms, TLD only utilized the past and current information of the object.

3. Laplacian Ranking SVM

In this section, firstly, the ranking SVM algorithm [17] is briefly reviewed. Then the Laplacian Ranking SVM is derived by introducing the manifold regularization [4].

3.1. Ranking SVM

Denote by \( X^1 = \{x_i : i = 1, ..., N_1\} \) and \( X^0 = \{x_j : j = N_1 + 1, ..., N_1 + N_0\} \) two sets of feature vectors. \( \forall i, j, x_i > x_j, \) i.e., \( x_i \) is ranked higher than \( x_j \), where \( x_i \in X^1 \), and \( x_j \in X^0 \). Suppose that the ranking function \( F(x) = w^T \Phi(x) \) satisfies the following conditions:

\[
x_i > x_j \iff F(x_i) > F(x_j).
\]

where \( \Phi(x) \) the implicit mapping of the feature \( x \) imposed by a kernel function. Training ranking SVM is a quadratic programming to balance the maximization of ranking margin and the minization of the ranking error of the training pairs. Its objective function is given by

\[
\min_{w, \eta} \|w\|^2 + C \sum_{i,j} \eta_{ij}
\]

\[
s.t. \quad w^T(\Phi(x_i) - \Phi(x_j)) \geq 1 - \eta_{ij}, \eta_{ij} \geq 0, \quad i = 1, ..., N_1, j = N_1 + 1, ..., N_1 + N_0.
\]

where \( C \) is a trade-off parameter between margin and training error.

3.2. Laplacian Ranking SVM

In weakly supervised ranking learning, we have two sets of labeled samples \( X^1_t \) and \( X^0_t \) till frame \( t \) and a set of weakly labeled samples \( X^w_{t+1} = \{x_k : k = N_t + 1, ..., N_t + N_w\} \) coming from frame \( t + 1 \), where \( N_t = N_1 + N_0 \). A typical smoothness assumption used in semi-supervised learning is that close samples in a high-density region should correspond to approximative values of the trained function \( f \) [7, 37]. When the labeled and weakly labeled samples lie on a manifold, they are represented as a weighted adjacency graph. The graph’s weight matrix \( W = [w_{ij}] \) encodes similarities between samples. The smoothness assumption over the graph is enforced by the manifold regularizer which can be empirically approximated by

\[
\sum_{i=1,j=1}^{N_t+N_w} w_{ij}(f(x_i) - f(x_j))^2 = f^T L f,
\]
where \( \mathbf{f} = [F(x_1), ..., F(x_{N_l+N_u})]^T \) is the scores of all samples, \( L \) is the graph Laplacian given by \( L = D - W \), and \( D \) is the diagonal matrix with \( D_{ii} = \sum_{j=1}^{N_l+N_u} w_{ij} \).

We plug the manifold regularizer in Eq.(3) into the supervised ranking problem (2), obtaining the following optimization problem which corresponds to the manifold regularization for ranking outputs.

\[
\begin{align*}
\min_{\alpha, \eta} & \quad \|w\|^2 + C \sum_{ij} \eta_{ij} + \gamma_1 \mathbf{f}^T L \mathbf{f} \\
\text{s.t.} & \quad \mathbf{w}^T (\Phi(x_i) - \Phi(x_j)) \geq 1 - \eta_{ij}, \eta_{ij} \geq 0, \quad i = 1, ..., N_l, j = N_l + 1, ..., N_l + N_u,
\end{align*}
\]

(4)

where \( \gamma_1 \geq 0 \) is the tradeoff parameter for the manifold regularizer. Obviously, (4) will be reduced to the normal ranking SVM (2) if \( \gamma_1 = 0 \).

By taking a similar orthogonality argument to that in [4], it is easy to prove the following Representer Theorem.

**Theorem 1** For Problem (4), the optimal \( w^* \) can be expressed as the following form:

\[
\begin{align*}
w^* & = \alpha^T \Phi,
\end{align*}
\]

(5)

where \( \alpha = [\alpha_1, ..., \alpha_N]^T, \Phi = [\Phi(x_1), ..., \Phi(x_N)]^T, N = N_l + N_u \).

By introducing Eq.(5) into Problem (4), the primal problem of Laplacian Ranking SVM can be written as

\[
\begin{align*}
\min_{\alpha, \eta} & \quad \alpha^T K \alpha + \gamma_1 \alpha^T K L K \alpha + C \sum_{ij} \eta_{ij}, \\
\text{s.t.} & \quad \sum_{k=1}^{N_l+N_u} \alpha_k (K(x_k, x_i) - K(x_k, x_j)) \geq 1 - \eta_{ij}, \\
& \quad \eta_{ij} \geq 0, \quad i = 1, ..., N_l, j = N_l + 1, ..., N_l + N_u,
\end{align*}
\]

(6)

where \( K(x_i, x_j) \) is the kernel function and \( K \) is Gram matrix \( K_{i,j} = K(x_i, x_j) \) over the labeled and weakly labeled points.

By using Lagrange multipliers, we get the dual of Laplacian Ranking SVM:

\[
\begin{align*}
\beta^* & = \max_{\beta} \quad \sum_{i=1}^{M} \beta_i - \frac{1}{2} \beta^T Q \beta, \\
\text{s.t.} & \quad 0 \leq \beta_i \leq C, i = 1, ..., M,
\end{align*}
\]

(7)

and \( \alpha^* = (2I + 2\gamma_1 L K)^{-1} J \beta^* \), where \( M = N_l \cdot N_u \), \( \beta = [\beta_1, ..., \beta_M]^T \), \( Q = J^T K (2I + 2\gamma_1 L K)^{-1} J \), and \( J = [(A \odot B)^T, (C \odot D)^T, (E \odot B)^T]^T \), \( A = I_{n_l}, B = 1_{n_u \times n_l}, C = 1_{n_l \times 1}, D = -1_{n_l \times 1}, E = 0_{n_u \times n_l}, \odot \) denotes the Kronecker product. This quadratic optimization problem (7) can be easily solved by standard optimization softwares such as MATLAB, CPLEX. The resulting ranking function can be expressed as

\[
\begin{align*}
F(x) & = \sum_{i=1}^{N_l+N_u} \alpha_i^* K(x_i, x).
\end{align*}
\]

(8)

### 4. Tracking with Laplacian Ranking SVM

#### 4.1. Preparation of Training Sets

As stated in Sec.1, our purpose is to combine the most credible information, the recent fluctuation, and possible future different changes of the object appearance to locate the object in the new frame. Therefore, we extract image patches from the initial and several the most recent frames to construct the labeled training sets, and patches from the next frame for the weakly labeled data.

##### 4.1.1 Preparation of Labeled Training Sets

The labeled training data consist of two sets, \( X^1_l \) and \( X^1_l \), and \( x_i \succ x_j \) if \( x_i \in X^1_l \) and \( x_j \in X^0_l \). Because, in general, there exists the semantical uncertainty of the exact location of the target object even if in the first frame, it is reasonable to assume that the patches around the located object within a couple of pixels could be accepted as object patches and should be ranked higher than other patches. Therefore, we define \( X^1_l = \{ x : ||l_s(x) - l^*_t|| \leq \alpha_l, s = 1, t - \Delta t, ..., t \} \) and \( X^0_l \subset Z = \{ x : \gamma_l < ||l_s(x) - l^*_t|| < \beta_l, s = t - \Delta t, ..., t \} \), where \( x \) is an image patch and \( l_s(x) \) is the column vector of its center position in frame \( s \), \( \Delta t \) expresses how many the most recent frames are taken to embody the appearance fluctuation, \( \gamma_l > \alpha_l \) and \( \beta_l \) control the area of sampling, and we randomly crop out patches from \( Z \) to produce \( X^0_l \).

It has been observed in the experiments that setting \( \alpha_l > 1 \) makes our tracker more robust.

##### 4.1.2 Preparation of Weakly Labeled Training Set

Sampling weakly labeled patches from the next frame before locating the object by our tracker is an important step. If the weakly labeled set, \( X^w_l \), were constructed by randomly sampling image patches from frame \( t + 1 \) for training the appearance model (8), \( F(x) \) would be easily degraded or even not better than the supervised ranking model. Therefore, it would be better if we could know the exact location of the object.

In reality, we can approximately and roughly evaluate parameters of the object state in frame \( t + 1 \) by using another tracker and bear its inaccuracy in location. We call it a weak tracker. There are a great many of trackers to be able to incorporate into our algorithm, for instance, normalized cross-correlation (NCC), KLT tracker [22], or even mean shift based tracker [8]. In our current implementation, the KLT tracker is employed to provide a rough and possible location of the object in the next frame.

Let \( x(l; l^*_t) \) denote the image patch centered at \( l^*_t \) in frame \( t \). Let \( \mathbf{W}(l; p) \) be the parameterized set of allowed warps, where \( p = (p_1, ..., p_n)^T \) is the vector of parameters. The warp \( \mathbf{W}(l; p) \) maps the pixel \( l \) of the patch \( x(l; l^*_t) \)
to the sub-pixel location $W(l; p)$ in frame $t + 1$. In our current implementation, the warp $W(l; p)$ only contains scaling and translation, i.e.,

$$W(l; p) = \begin{bmatrix} p_1 & 0 & p_2 \\ 0 & p_1 & p_3 \end{bmatrix} \begin{bmatrix} l_x \\ l_y \\ 1 \end{bmatrix},$$

(9)

where $l = (l_x, l_y)^T$, $p_1$ is the scaling factor, and $p_2, p_3$ are translation factors.

By minimizing the sum of squared errors between two image patches, $x(l; l^*_t)$ and $x(l; W(l^*_t; p))$ in frame $t + 1$,

$$e_{KLT} = \min_p \sum_{l \in \ell(l^*_t)} \|x(l; W(l^*_T; p)) - x(l; l^*_T)\|^2,$$

(10)

we get the optimal $p_\ast$. If $e_{KLT} < \Delta_w$, then it is probable that the KLT provides a rough location of the object.

Considering the inaccuracy of the aforementioned locations, we set $X^w_{t+1} = \{x(l; W(l^*_T; p))||e_{KLT} < \Delta_w, \|p - p_\ast\|^2 \leq \Delta_p^2\}$ as the set of weakly labeled patches in frame $t + 1$. If $e_{KLT} \geq \Delta_w$, then $X^w_{t+1} = \emptyset$.

4.2. Location of the object

The ranking function $F_{t+1}(x)$ of Eq.(8) is trained through solving the Problem (7) with the labeled sample sets $X_t^1$ and $X_t^0$ and the weakly labeled set $X^w_{t+1}$. If $X^w_{t+1} = \emptyset$, then $\gamma = 0$. That is, $F_{t+1}(x)$ is trained through the normal ranking SVM (Problem (2)) with $X_t^1$ and $X_t^0$ in this case.

Let $X_{t+1}$ be the set of location candidates of the object in the new frame. Set $X_{t+1} = \{x||\|x - W(l^*_T; p^\ast)\| \leq \gamma\}$ if $e_{KLT} < \Delta_w$, $X_{t+1} = \{x||\|x - l^*_T\| \leq \gamma\}$ otherwise, where $\gamma$ is the search radius. That is, if the KLT can provide a rough location of the object in the new frame, then we believe that the true location should be near to the rough one in the parameter space. Otherwise, the true location will be supposed to be near to the last location in frame $t$.

It is obvious that more complicated dynamic models can easily be combined into our tracking algorithm in this step.

Then, the patch highest scored by $F_{t+1}(x)$ is accepted as the location of the object in frame $t + 1$, i.e.,

$$l^*_t = l(\arg \max_{x \in \ell_{t+1}} F_{t+1}(x)).$$

(11)

The entire procedure of the proposed tracking algorithm, LRSVT, is summarized in Algorithm 1.

5. Experiments

We implement our online Laplacian ranking SVM tracker (LRSVT) in C++ language and evaluate its performance on public challenging image sequences.\footnote{LRSVT ran at about 8 fps on the desktop (Intel Core Dual CPUs, E8300, 2.83GHz, 2G RAM)} LRSVT is also quantitatively compared with some state-of-the-art tracking algorithms, FragTracker [1], Online Boosting tracker (OAB) [12], MILTracker [2], L1 tracker [23], and Ranking Support Vector Tracker (RSVT) [3]. Their codes are publicly available and the parameters are turned finely. All algorithms are compared in terms of the same initial positions in the first frame.

We concatenate $N$ generalized Haar-like features to form a vector to represent each image patch. Each feature is composed of 4-8 rectangles which are generated randomly, and each rectangle has a random real valued weight within the range of $[-1, 1]$ [9].

In our testing and comparison, all the parameter settings were fixed in our experiments. $N = 100$, $\gamma_t = 0.2$, $\alpha_t = 2$, $\gamma_l = 8$, $\beta_l = 40$, and $C = 5$. When a new frame arrives, the search radius $\gamma = 25$.

To evaluate the performance among the above algorithms and ours, the center errors between the tracking results and the ground truth are calculated and reported on image sequences. Due to the randomness in our tracking algorithm, we run our tracker five times on every sequence and average the results.

Throughout the experiments, we evaluated all approaches on the following sequences: Sylvester, David, Face Occ, Face Occ2, Tiger 1, Tiger 2, Snack Bar, Coke Can and

Algorithm 1: Online Laplacian RSVT

- **Input:** Frames 1, 2, 3, ..., and the initial location, $l^*_1$, of the object in frame 1.
- **Output:** object locations, $l^*_2, l^*_3, ..., $ in the subsequent frames.

1. $t = 1$,
2. Construct two sets of samples, $X^1_t$ and $X^0_t$ (Sec. 4.1.1),
3. Use the KLT tracker to find the rough location of the object in frame $t + 1$,
4. If $e_{KLT} < \Delta_w$ (Eq.(10))
   - Construct the set of weakly labeled samples, $X^w_{t+1}$ (Sec. 4.1.2),
   - Construct data adjacency graph and calculate edge weight matrix $W = [w_{ij}]$,
   - Learn the ranking function $F(x)$ using Laplacian Ranking SVM (Problem 7 and Eq.(8)),
   - else
   - Learn the ranking function $F(x)$ with $\gamma = 0$, (Problem 7 with $\gamma = 0$),
5. Locate the object in frame $t + 1$, producing $l^*_t + 1$,
6. $t = t + 1$, go to 2.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Main Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sylvester</td>
<td>3D-rotation, varying illumination, large scale change</td>
</tr>
<tr>
<td>David</td>
<td></td>
</tr>
<tr>
<td>Face Occ</td>
<td>heavy occlusion</td>
</tr>
<tr>
<td>Face Occ2</td>
<td>heavy occlusion, rotation in plane</td>
</tr>
<tr>
<td>Tiger1</td>
<td>heavy occlusion, scale change</td>
</tr>
<tr>
<td>Tiger2</td>
<td>3D-rotation, fast appearance change</td>
</tr>
<tr>
<td>Coke Can</td>
<td>varying illumination, occlusion, textureless</td>
</tr>
<tr>
<td>Lemming</td>
<td>3D-rotation, occlusion, scale change</td>
</tr>
<tr>
<td>Snack Bar</td>
<td>scale change, background clutter</td>
</tr>
</tbody>
</table>

Table 1. The main challenges of each image sequence.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>OAB</th>
<th>Frag</th>
<th>MIL</th>
<th>L1</th>
<th>RSVT</th>
<th>LRSVT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sylvester</td>
<td>25</td>
<td>11</td>
<td>11</td>
<td>15</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>David</td>
<td>49</td>
<td>46</td>
<td>23</td>
<td>51</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Face Occ</td>
<td>43</td>
<td>6</td>
<td>27</td>
<td>8</td>
<td>28</td>
<td>12</td>
</tr>
<tr>
<td>Face Occ2</td>
<td>21</td>
<td>45</td>
<td>20</td>
<td>34</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Tiger1</td>
<td>35</td>
<td>40</td>
<td>15</td>
<td>30</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Tiger2</td>
<td>34</td>
<td>38</td>
<td>17</td>
<td>48</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>Coke Can</td>
<td>25</td>
<td>63</td>
<td>20</td>
<td>53</td>
<td>17</td>
<td>12</td>
</tr>
<tr>
<td>Lemming</td>
<td>149</td>
<td>178</td>
<td>73</td>
<td>167</td>
<td>38</td>
<td>23</td>
</tr>
</tbody>
</table>

Table 2. Average center errors between the tracking results and their ground truth. The bold represents the best result, and the underlined the second best.

Lemming. The first eight sequences are from [2] and the last one from [28]. The main challenges of these sequences on tracking are summarized in Table 1.

In comparison with FragTracker, OAB, MILTracker and L1 tracker, our tracker is more robust against illumination and visual appearance changes. It can be seen in sequence David (Fig.2, the 1st row) that MILTracker drifted at around frame 39 and more from frame 392 on due to the varying illumination. Similar drifts can be observed in sequence Coke Can (Fig.2, the 4th row).

L1 tracker is based on the error of sparse reconstruction, in which the candidate with the smallest error is considered as the tracking target. Therefore, it may deal with the appearance fluctuation well. Nevertheless, if the object appearance undergoes different or substantial changes from the previous, L1 tracker will easily drift away from the true location. Another reason to lower its tracking accuracy is that its appearance model is not discriminative, therefore, may be confused by background regions similar to the object. Drifting away from true locations caused by the above two reasons can be easily observed in sequence David (Fig.2, the 1st row) around frame 295 and sequence Face Occ2 (Fig.2, the 3rd row) around frame 649.

Occlusion is a very challenging problem for robust tracking. L1 tracker and FragTracker worked slightly better than ours on Sequence Face Occ. This is because the target shows little appearance change except for occlusion and both L1 tracker and FragTracker include special mechanisms to handle occlusion. On other sequences, i.e., Face Occ2 (Fig.2, the 3rd row), Tiger1 (Fig.2, the 2nd row), and Tiger2, however, there exist changes of target appearances. Therefore, our LRSVT outperformed both of them.

3D-rotation often causes a tracker drifting because of loss of depth information in the 2D image [34]. This phenomenon is apparent in sequences Tiger1 (Fig.2, the 2nd row) and Tiger2, in which a little 3D rotation can lead to large changes in appearances. In fact, 3D-rotation may introduce changes of the appearance different from the previous. It can be seen from Fig.2 and Table 2 that our tracker worked better than others on sequences Tiger1 (Fig.2, the 2nd row), Tiger2, and Lemming (Fig.2, the 5th row).

We also compared our tracker with MILTracker [2] and L1 tracker in the case of scale changes of the object bounding boxes on sequences David (Fig.3, the 1st and 2nd rows) and Snack Bar (Fig.3, the 3rd row). In Fig.3, we directly copied their published tracking results. It can be seen, from Fig.2, that our tracker handled the scale change of the object well. In Fig.3, it is observed that the results of MILTracker were really inconsistent in the cases of different initial random seeds for generating the Haar-like features [9], while our tracker was not affected at all.

The whole quantitative comparisons are shown in Table 2 and Fig.4. It can be concluded that our track was almost always more robust than others.

6. Conclusions

A novel tracking algorithm - the online Laplacian ranking support vector tracking (LRSVT) - has been presented in this paper. When constructing the appearance model, unlike most of the existing tracking approaches, we incorporate the weakly labeled data in the next frame to learn different or substantial changes of the visual appearance, while utilizing the labeled information in the initial and the most recent frames to resist the full occlusion and adapt to rapid fluctuation of the object appearance. Through experiments on public benchmark sequences, we have demonstrated that our novel algorithm can track objects very well under large pose, scale variation and heavy occlusion, and that our LRSVT can almost always outperform the state-of-the-art algorithms.

References

Figure 2. Representative frames of 5 sequences with cyan, yellow, and red object bounding boxes generated by L1 tracker [23], MIL-Tracker [2], and our LRSVT, respectively. This figure is best viewed in color.

Figure 3. Representative frames on sequences David and Snack Bar under scale changes. Cyan, yellow, and red object bounding boxes were generated by L1 tracker [23], MILTracker [2], and our LRSVT, respectively. In David sequence, both MILTracker and LRSVT used two different random seeds, and MILTracker was affected seriously, whereas LRSVT was not. This figure is best viewed in color.


Figure 4. The pixel errors for every sequence tested in the experiments. This figure is best viewed in color.