OBJECT TRACKING VIA ONLINE METRIC LEARNING

Yang Cong[1,2], *Junsong Yuan*[2], *Yandong Tang*[1]

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Science
[2]Nanyang Technological University, Singapore

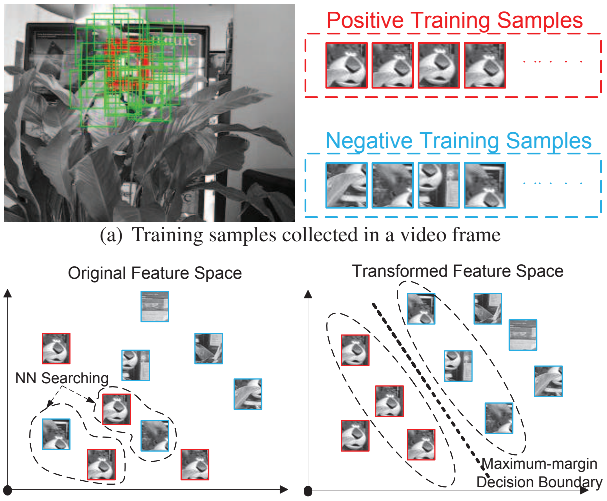
ABSTRACT

By considering visual tracking as a similarity matching problem, we propose a self-supervised tracking method that incorporates adaptive metric learning and semi-supervised learning into the framework of object tracking. For object representation, the spatial-pyramid structure is applied by fusing both the shape and texture cues as descriptors. A metric learner is adaptively trained online to best distinguish the foreground object and background, and a new bi-linear graph is defined accordingly to propagate the label of each sample. Then high-confident samples are collected to self-update the model to handle large-scale issue. Experiments on the benchmark dataset and comparisons with the state-of-the-art methods validate the advantages of our algorithm.

Index Terms— tracking, metric learning, semi-supervised learning, online learning [1]

1. INTRODUCTION

Visual tracking is to find an object in the consecutive image frames, which matches the given template properly, and it is also broadly applied as a key step in many applications, such as video surveillance, Unmanned Aerial Vehicle [1], and human-computer interactions. Without considering the issue of object representation, matching the visual appearances of the target in an image sequence is the most critical problem in video based object tracking, i.e. the selection of distance metric to determine the closet match in the feature space. Most existing tracking methods employ a fixed pre-specified metric, e.g. the Euclidean metric, the Matusita metric [2], the Bhattacharyya coefficient [3], the Kullback-Leibler divergence [4], the information-theoretic similarity measures [5] and a combination of them [6]. However, simply using such a pre-defined metric is problematic in practice, which often leads to a false positive match that fails the tracker. In order to choose a robust metric adaptively, metric learning is incorporated recently [7, 8, 9]. Once the similarity metric is determined, visual tracking can be considered as a Nearest-neighbor (NN) searching problem using metric learning for similarity search. For NN-searching based visual tracking, besides choosing suitable parameters, the accuracy can be severely degraded by the presence of the noisy or irrelevant features, which will affect the performance of tracking in turn. To overcome this, our motivation is why not apply both training samples and testing samples to make a better decision, i.e. incorporate semi-supervised learning.


Positive Training Samples
Negative Training Samples
Original Feature Space
NN Searching
Transformed Feature Space
Maximum-margin
Decision Boundary

(a) Training samples collected in a video frame

(b) Training samples collected in the feature space

Fig. 1. The illustration of the training sample collection process.

Thus, in this paper, we consider tracking as a similarity learning issue and propose a self-supervised online tracking method. In comparison with most existing methods, our proposed method not only learns the metric adaptively but also improves NN-searching into the label propagation using our defined bi-linear graph. The main contributions are listed below: i) Firstly, combining online metric learning and semi-supervised label propagation, we propose a general model for online self-supervised similarity measurement. ii) Secondly, we define a bi-linear graph to measure the pairwise similarity for graph-based label propagation without tuning any parameters. iii) Finally, we propose an object tracking framework, which need less computational resource and can self-update online to address large-scale data.

[1] This work has been supported by NSFC (61105013) to Dr. Yang Cong and by Nanyang Assistant Professorship (SUG M58040015) to Dr. Junsong Yuan.

2. FORMULATION AND A GENERAL MODEL

We propose an online learning framework for visual tracking. Two key issues need to be considered here: 1) the pairwise similarity measurement depending on metric learning; 2) discriminative criterion using graph-based semi-supervised learning to propagate the label of each testing samples via our new bi-linear graph. For specific, we first learn a matrix W for similarity measurement, then classify new unlabeled data using W, lastly adds those new labeled data with high confidence scores to update W accordingly. Such a process iterates for online processing.

2.1. Online Metric Learning

The goal of Online Metric Learning (OML) is to learn a similarity function with a bi-linear form as:

$$s_W(p_i, p_j) \equiv p_i^T W p_j, \tag{1}$$

where $p_i, p_j \in \mathbb{R}^d$ are the feature vector, $W \in \mathbb{R}^{d\times d}$ and s_W assigns higher scores to more similar pairs of feature vectors. To estimate W, we have the following convex model with a soft margin:

$$\begin{aligned} W^i = \arg\min_W \frac{1}{2}\|W - W^{i-1}\|_{Fro}^2 + C\xi \\ \text{s.t. } l_W(p_i, p_i^+, p_i^-) \le \xi \text{ and } \xi \ge 0, \end{aligned} \tag{2}$$

where $\|\cdot\|_{Fro}$ is the Frobenius norm (point wise L_2 norm), C is the tuning parameter, p_i and p_i^+ belong to the same class and p_i and p_i^- vice verse. In the i^{th} iteration, W^i is updated to optimized a trade-off between staying close to the previous parameter W^{i-1} and minimizing the loss on the current triplet $l_W(p_i, p_i^+, p_i^-) = \max\big(0, 1 - s_W(p_i, p_i^+) + s_W(p_i, p_i^-)\big)$. The passive-Aggressive algorithm [10, 11] is adopted to solve the above model iteratively ($C = 0.2$):

$$\begin{cases} W = W^{i-1} + \tau V_i \\ \tau = \min\Big\{C, \dfrac{l_{W^{i-1}}(p_i, p_i^+, p_i^-)}{\|V_i\|^2}\Big\}. \end{cases} \tag{3}$$

Depending on this, we define the bi-linear graph as:

Definition 1 *Bi-linear Graph: Assume the similarity of pairwise points* $\forall\ i,\ j,\ \ 1 \le i, j \le N,\ i \ne j$ *is defined as*

$$S_{i,j} \ = \ \max(0,\ S_w(i,j)) \ = \ \max(0,\ p_i^T W p_j). \tag{4}$$

For $p_i \in P,\ i \in [1, \dots, N]$, *we obtain a matrix* $\{S_{ij},\ 1 \le i, j \le N\}$, *where symmetric version is* $S_{i,j} = (S_{i,j} + S_{j,i})/2$.

In comparison with other graph models, e.g. $k-$NN or $\varepsilon-$NN graph, our bi-linear graph can maintain the accuracy without tuning parameters or prior knowledge of the topology graph.

2.2. Label Propagation via Bi-linear Graph

To detect the object, we use the graph-based semi-supervised learning, also called label propagation. We first define graph $\mathrm{G} = (\mathscr{V}, \mathscr{E})$, where nodes $\mathscr{V}$ denotes $N = n + m$ feature vector (n and m are the number of training and testing samples and $m = 1$ in our case); and $\mathscr{E}$ is the similarity of pairwise nodes of bi-linear graph S. We define a $N \times N$ probability transition matrix $P_{ij} = \frac{S_{ij}}{\sum_{k=1}^n S_{ik}}$, which can be split into labeled and unlabeled sub-matrices $P = \begin{bmatrix} P_{ll} & P_{lu} \\ P_{ul} & P_{uu} \end{bmatrix}$. Let $F = \binom{F_l}{F_u}$, where $F_l = [f_1, f_2, \dots, f_n]$ denotes the labeled data, and $F_u = [f_{n+1}, f_{n+2}, \dots, f_{n+m}]$ is the unlabeled data. We have $F_u \leftarrow P_{uu}F_u + P_{ul}F_l$, which leads to

$$F_u = \lim_{t\to\infty}(P_{uu})^t F_u^0 + (\sum_{i=1}^{t}(P_{uu})^{(i-1)})P_{ul}Y_l, \tag{5}$$

where F_u^0 is the initial value of F_u. Since the sum of each row of P equals to 1, we have $(P_{uu})^n$ converge to zero, i.e. $(P_{uu})^n F_u^0 \to 0$. Using the Taylor Equation, the second item can be written as $F_u = (I - P_{uu})^{-1} P_{ul} Y_l$. Due to P_{uu} is a fixed real number in our case, $(I - P_{uu})^{-1}$ is also a real number and invertible, so $F_u \propto P_{ul}Y_l$. Thus, F_u can be calculated using the largest values of each row, which is also consistent with the simplified function:

$$c_x^\star = \arg\max_c E_c(x), \quad E_c(x_i) \ = \ \sum_{j=1}^{n} \delta_c(j) S_{i,j}, \tag{6}$$

where $c \in \{1, \dots, \mathrm{K}\}$ (K is the number of class, $\mathrm{K} = 2$ here), x_i is the query sample and $\delta_c(i)$ is a indicate function. $E_c(x)$ is the energy function, which measures the cost of x belonging to class c. Thus, given x, the optimal solution of c is the one with maximize the cost of $E_c(x)$.

3. OUR TRACKING FRAMEWORK

3.1. Object Representation

We use a two-level spatial pyramid to reserve spatial context, where the first level is the whole object candidate and the second level is to split each object into 2×2 sub-regions, then we concatenate them into a whole feature vector. The appearance of each sub-region is represented by combining both Edge Orientation Histogram (EOH) and Local Binary Pattern (LBP) histogram. For EOH, we quantize each pixel into 9 bins with the first bin for non-edge regions and the others for 8 directions. Then, for LBP histogram, we quantize it into 32-bin histogram. Therefore, the total dimension is $d = 205$ in this paper. Moreover, we adopt the modified version of integral histogram to make the computational complexity of the histogram calculation for linear.

Algorithm 1 Online Metric Learning Tracker (OMLTrack)

```
Input: Object $p$, Query sample $q$
Output: $W^\star$
Initialization: Get samples $p_i^+$ and $p_i^-$
Train the metric learner $W$
for each testing sample $q$ do
    Generate Bi-linear Graph $S$
    $c_q^* = \arg\max_{c \in C} E_c(q)$
    Refine Localize $q$ by mean-shift
    if $E(c_q)/E(\bar{c}_q) > T_\xi$ then
        W = Update (q)
    end if
end for
return $W^\star = W$

Function W = Update (q)
while $i <$ ITER-MAX $\bigcap \|W^i - W^{i-1}\|_{Fro} < T_w$ do
    Get sample $q_i^+ \in c_{q_i}$ and $q_i^- \in \bar{c}_{q_i}$
    Update W by Eq. (3)
    $i = i + 1$
end while
```

3.2. Algorithm Details

For initialization, we extract the foreground object and its nearest neighborhood to generate the positive objects $p_i^+, i \in [1 \cdots N^+]$ ($N^+ = 10$ in this paper) and its surrounding background to generate the negative objects $p_i^-, i \in [1 \cdots N^-]$ ($N^- \approx 30$), as the green and red rectangles in Fig. 1. Then, we train the metric learner of matrix W by Eq. 3 through iterative random sample p_i^+ and p_i^-.

For testing, we calculate the similarity between the testing sample q^* and templates $\{p_i^+, p_i^-\}$ by Eq.1, and propagate its label and estimate its confident score by Eq.6. The samples with higher confident scores are used to self-update the model by a rough criterion:

$$E_{c^\star}(q) > T_\xi \times E_{\bar{c}}(q), \quad \forall \bar{c}, \ \bar{c} \notin c^\star. \tag{7}$$

In this paper $T_\xi = 1.2$. To refine the object position, we pursuit the final object position with the maximum likelihood by mean shift algorithm on the confident score map.

Our tracking method can online self-update itself. Thus, if Eq.7 is satisfied, we extract foreground objects $\{q_j^+\}$ and background objects $\{q_j^-\}$ around q, and combine them with $\{p_i^+, p_i^-\}$ to update the metric learner. This selp-supervised online procedure is processed step-by-step. Alternately, we can also change this strategy to update the metric learner every few frames, i.e. mini-batch updating. The computational complexity of the entire procedure is $O(d^2)$, which is very low and can run fast. The details of testing and online learning procedure is shown in Alg. 1.

4. EXPERIMENTS AND COMPARISONS

In this section, we systematically apply our proposed algorithm to several published challenging video sequences to justify the effectiveness. We also compare our online metric learning tracking method (OMLTrack) with two recent prominent tracking methods, i.e. Multiple Instance Learning Tracking (MILTrack) [12] and L_1 tracking [13].

Evaluation criterion: The Average Tracking Precision (ATP) is used to evaluate the performance, which is extended by the Average Precision (AP) used in the PASCAL grand challenge. Assume T_j and G_j are the bounding boxes of predicted target and ground truth in frame j, respectively. The ATP for a tracker of an object in a video clip is defined as: ATP $= \frac{1}{t}\sum_{j=1}^{t} r_j = \frac{1}{t}\sum_{j=1}^{t} |G_j \bigcap T_j| / |G_j \bigcup T_j|$, where $r_j \in [0,1]$, t is the frame number. So the greater the ATP value, the better the tracker performs, and ATP $\equiv 1$ ideally.

Experiments: We perform our experiments on several public video sequences. All the sequences are labeled the center of the groundtruth object for every 5 frames. Each frame is gray scale and resized to 320×240 pixels. Fig.3 demonstrates some tracking results from our OMLTrack, L_1Track and MILTrack. In the Coke Can video, our OMLTracker tracks the Coke Can robustly in spite of severe occlusions in the 4th image. The Surf and Sylvester video data are often used in tracking papers as they present difficult tracking scenarios, such as challenging lighting, changes in scale and poses, and occlusions. Nevertheless, our algorithm consistently produces good results even in these challenging examples. The Tiger2 video contains frequent fast motions, which lead to motion-blur sometimes. Yet our algorithm again tracks the object well in this case. The quantitative results are summarized in Table 1 and Fig. 2, where the average ATP of our OMLTrack is 0.612 higher than MILTrack (0.60185) and L_1Track (0.2445).

Table 1. The quantitative comparison of Average Tracking Precision (ATP) of each video data by different methods

Video	#Frames	L_1Track	MILTrack	OMLTrack
CokeCan	291	0.0618	0.3173	0.4405
Sylvester	855	0.6208	0.7556	0.7941
Surf	375	0.0823	0.7106	0.7004
Tiger	364	0.2143	0.6239	0.5141

5. CONCLUSION

We propose a self-supervised object tracking method via online metric learning and semi-supervised learning. Given a number of labeled data followed by a sequential input of unseen testing samples, the similarity metric is firstly learnt by our model to maximize the margin between foreground and background samples. The pair-wise similarity is then measured by our new bi-linear graph for online label propagation of the new data. With the most confident samples adopted to update the model, our model can be improved incrementally

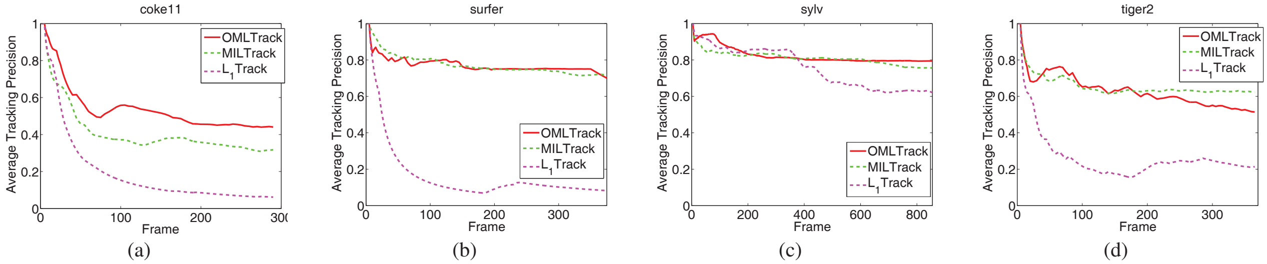


Fig. 2. The quantitative comparison results of our OMLTrack to MILTrack and L_1Track on different video data measured by Average Tracking Precision (ATP) criterion.

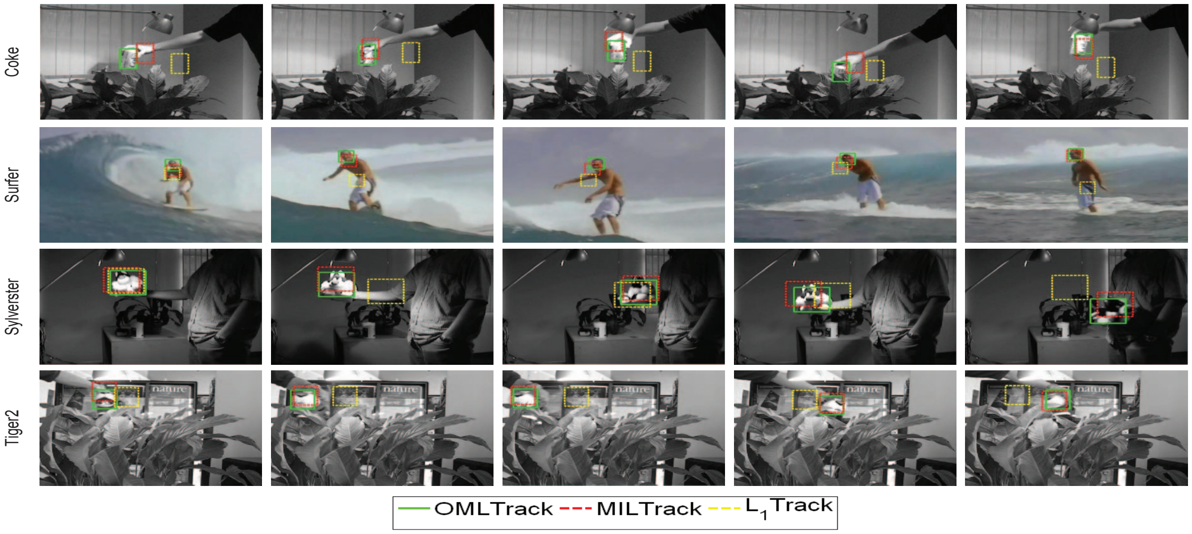


Fig. 3. The comparison of tracking results on different video data, where the green line, the red dash line and the yellow dash line correspond to the result of our Online Metric Learning tracking (OMLTrack), Multi Instance Learning Tracking (MILTrack) and L_1 tracking, respectively.

and is also computationally efficient. Experiments on various benchmark datasets and comparisons with other state-of-the-art methods demonstrate the effectiveness and efficiency of our algorithm.

6. REFERENCES

[1] Baojie Fan, Yingkui Du, Linlin Zhu, Jing Sun, and Yandong Tang, "A robust template tracking algorithm with weighted active drift correction," in *Pattern Recognition Letters*, 2011, vol. 32, pp. 1317–1327.

[2] G.D. Hager, M. Dewan, and C.V. Stewart, "Multiple kernel tracking with ssd," in *CVPR*. IEEE, 2004, vol. 1, pp. I–790.

[3] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[4] A. Elgammal, R. Duraiswami, and L.S. Davis, "Probabilistic tracking in joint feature-spatial spaces," in *CVPR*. IEEE, 2003, vol. 1, pp. I–781.

[5] P. Viola and W.M. Wells III, "Alignment by maximization of mutual information," *International journal of computer vision*, vol. 24, no. 2, pp. 137–154, 1997.

[6] C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," in *CVPR*, 2005, vol. 1, pp. 176–183.

[7] G. Tsagkatakis and A. Savakis, "Online distance metric learning for object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1810–1821, 2011.

[8] N. Jiang, W. Liu, and Y. Wu, "Learning adaptive metric for robust visual tracking," *IEEE Transactions on Image Processing*, pp. 2288–2300, 2010.

[9] X. Wang, G. Hua, and T. Han, "Discriminative Tracking by Metric Learning," *ECCV*, pp. 200–214, 2010.

[10] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "An online algorithm for large scale image similarity learning," *NIPS*, vol. 21, pp. 306–314, 2009.

[11] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *The Journal of Machine Learning Research*, vol. 7, pp. 585, 2006.

[12] B. Babenko, Ming-Hsuan Yang, and S. Belongie, "Visual Tracking with Online Multiple Instance Learning," in *CVPR*, 2009.

[13] X. Mei and H. Ling, "Robust visual tracking using 11 minimization," in *ICCV*, 2009, pp. 1–8.