Cascaded Tracking with Incrementally Learned Projections

Lianghua Huang(✉)

Institute of Automation, Chinese Academy of Sciences, Beijing, China
huanglianghua@bit.edu.cn

Abstract. A convention in visual object tracking is to only favor the candidate with maximum similarity score and take it as the tracking result, while ignore the rest. However, surrounded samples also provide valuable information for target locating, and the combination of their votes can produce more stable results. In this paper, we have proposed a novel method based on the supervised descent method (SDM). We search for the target from multiple start positions and locate it with their votes. For evaluating each predicted descent direction, we have presented a confidence estimating scheme for SDM. To adapt the tracking model to appearance variations, we have further presented an incremental cascaded support vector regression (ICSVR) algorithm for model updating. Experimental results on a recent benchmark demonstrate the superior performance of our tracker against state-of-the-arts.

1 Introduction

As a fundamental subject in computer vision, visual object tracking plays a critical role in numerous applications including video surveillance, gait recognition, behavior analysis and robotics. Recent years have witnessed great progress in visual tracking [1–4]. Despite decades of studies, tracking is still a challenging task due to large appearance variations such as object deformation, occlusions, illumination variation and background clutter.

There are two main categories of tracking approaches: generative trackers and discriminative trackers. Generative approaches [5–8] take visual tracking as an appearance reconstruction problem. They mainly focus on the reconstruction model and online templates updating. Representative trackers are IVT [5] and sparse representation based trackers [6–8]. On the other hand, discriminative models [9–12] view tracking as a classification or regression task. They learn classifiers online with automatically labeled samples and locate the target with the candidate of maximum classification score. Some discriminative trackers are Struck [13], SCM [9], MEEM [14] and deep learning based methods [15–17]. Generally speaking, discriminative models are more robust against background clutters and thus they usually perform much better than generative ones.

A convention in tracking approaches, generative or discriminative, is to only favor the candidate with maximum similarity score, and afterwards the rest samples have no impact on the tracking result. However, surrounded samples also

Z. Zhang and K. Huang (Eds.): IVS 2016, CCIS 664, pp. 56–64, 2016.
DOI: 10.1007/978-981-10-3476-3_7

provide valuable information for target locating, and the combination of their estimations can produce more stable results without increasing computational burden.

In this paper, we have proposed a novel method for visual tracking. Instead of basing the tracking result on one sample with maximum score, we approach the target from multiple surrounded candidates in a cascaded way by using the Supervised Descent Method (SDM) [18], and locate the target by searching for the most densely voted position. The SDM models the optimization for a non-linear problem with cascaded linear projections, which has been applied in various areas including facial landmark detection [18], extrinsic camera calibration [19] and visual tracking [20]. To provide an evaluation scheme for each predicted offset, we have presented a confidence estimation model for SDM which is learned from samples and updated online. To adapt the model to target appearance variations, we have further proposed an Incremental Cascaded Support Vector Regression (ICSVR) algorithm for model updating.

2 The Proposed Method

This section presents details on the proposed tracking model.

2.1 Cascaded Regression

The observation model in our approach is constructed based on the supervised descent method (SDM) [18], which learns the projection from features to descent directions in a cascaded way.

Specifically, for an object located at $\mathbf{s} \in \mathbb{R}^d$, we draw samples $\{\mathbf{s}_i\}_{i=1}^n$ around $\mathbf{s}$ to obtain training data $\{(\varDelta\mathbf{s}_i, \phi_i)\}_{i=1}^n$, where $\phi_i \in \mathbb{R}^p$ denotes the extracted feature and $\varDelta\mathbf{s}_i = \mathbf{s}_i - \mathbf{s}$ is the offset. The SDM learns the projections $\{\mathbf{R}_k \in \mathbb{R}^{d \times p}\}_{k=1}^C$ in a cascades way by iteratively optimizing the following C problems:

$$\min_{\mathbf{R}_k} \sum_i \|\varDelta\mathbf{s}_i^k - \mathbf{R}_k\phi_i^k\|_2^2 + \lambda\|\mathbf{R}_k\|_2^2,\ k = 1, \cdots, C, \tag{1}$$

where k denotes the cascade index and $\mathbf{s}_i^1 = \mathbf{s}_i$, $\phi_i^1 = \phi_i$, $\varDelta\mathbf{s}_i^k = \mathbf{s}_i^k - \mathbf{s}$, λ is a regularization parameter. With learned matrices $\{\mathbf{R}_k\}_{k=1}^C$, the iterative regression from a start state $\mathbf{s}_i^1$ to the estimated one $\hat{\mathbf{s}}_i = \mathbf{s}_i^{C+1}$ is formulated as:

$$\mathbf{s}_i^{k+1} = \mathbf{s}_i^k + \mathbf{R}_k\phi_i^k,\ k = 1, \cdots, C. \tag{2}$$

In our method, we use the support vector regression (SVR) algorithm for learning the projection matrices $\{\mathbf{R}_k\}_{k=1}^C$ since it is proven experimentally to be more robust against sample noise. Let $\mathbf{r}_{kj}$ denotes the jth row of $\mathbf{R}_k$, and s_{ij}^k denotes the jth entry of $\mathbf{s}_i^k$, the cascaded SVR is formulated as:

$$
\begin{aligned}
\min_{\mathbf{r}_{kj},\boldsymbol{\xi}_{ki},\boldsymbol{\xi}^*_{ki}} \quad & \frac{1}{2}\|\mathbf{r}_{kj}\|_2^2 + \eta_1 \sum_{i=1}^{n}(\xi_{ki}+\xi^*_{ki}),\\
s.t. \quad & \mathbf{r}_{kj}\cdot\phi_{\boldsymbol{ki}} - \Delta s^k_{ij} \le \varepsilon_1 + \xi_{ki},\\
& \Delta s^k_{ij} - \mathbf{r}_{kj}\cdot\phi_{\boldsymbol{ki}} \le \varepsilon_1 + \xi^*_{ki},\\
& \xi_{ki},\ \xi^*_{ki} \ge 0\\
& i = 1,\cdots,n,\ k = 1,\cdots,C
\end{aligned}
\tag{3}
$$

where η_1 is a regularization factor, ξ_{ki}, ξ^*_{ki} are slack variables and ε_1 is a preset margin which is fixed to $\varepsilon_1 = 5$ empirically in our experiments.

2.2 Confidence Evaluation

Despite the effectiveness of SDM, its main drawback is the lack of a mechanism for indicating how reliable an offset prediction is. In this section, we present a confidence evaluation scheme for SDM.

In training stage, if one regress iteration pulls a sample closer to the groundtruth, we say that the sample is more credible and vice versa. Based on the idea, we propose to learn an extra set of projection matrices $\{\mathbf{Q}_k \in \mathbf{R}^{1\times p}\}_{k=1}^C$ for confidence evaluation. We take the ratio of overlap rates before and after regression $\theta^k_i = (o^{k+1}_i)^2/o^k_i$ (where o^k_i denotes the overlap between $\mathbf{s}^k_i$ and $\mathbf{s}$) as the label to train $\{\mathbf{Q}_k\}_{k=1}^C$:

$$
\begin{aligned}
\min_{\mathbf{Q}_k,\xi_{ki},\xi^*_{ki}} \quad & \frac{1}{2}\|\mathbf{Q}_k\|_2^2 + \eta_2 \sum_{i=1}^{n}(\xi_{ki}+\xi^*_{ki}),\\
\text{s.t.} \quad & \mathbf{Q}_k\cdot\phi_{\boldsymbol{ki}} - \theta^k_i \le \varepsilon_2 + \xi_{ki}\\
& \theta^k_i - \mathbf{Q}_k\cdot\phi_{\boldsymbol{ki}} \le \varepsilon_2 + \xi^*_{ki}\\
& \xi_{ki},\ \xi^*_{ki} \ge 0,\\
& i = 1,\cdots,n,\ k = 1,\cdots,C
\end{aligned}
\tag{4}
$$

When testing, the reliability c_i of each sample is computed as:

$$
c_i = \prod_{k=1}^{C} \theta^k_i,\ k = 1,\cdots,C. \tag{5}
$$

2.3 Target Locating

When locating target in a new frame, we sample around the last estimated position to obtain m candidates $\{\mathbf{s}_i,\phi_i\}_{i=1}^m$. With the learned cascaded model, we iteratively pull each sample $\mathbf{s}_i$ to the target location:

$$
\mathbf{s}^{k+1}_i = \mathbf{s}^k_i + \mathbf{R}_k\phi_i,\ k = 1,\cdots,C. \tag{6}
$$

After C iterations, we obtain all the estimated states $\hat{\mathbf{s}}_i = \mathbf{s}^{C+1}_i$. Intuitively, the most densely voted position is more likely to be the target location. In our method, we use the dominant set [21] algorithm for locating the voting center.

The dominant set algorithm computes sample weights w_i by optimizing:

$$\begin{aligned} \max_{\mathbf{w}} \;\; & \mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w}, \\ \text{s.t.} \;\; & \mathbf{w} \in \mathit{\Lambda}, \end{aligned} \tag{7}$$

where $\mathit{\Lambda} = \{\mathbf{w} \in \mathbb{R}^m : \mathbf{w} > \mathbf{0} \text{ and } \mathbf{e}^{\mathrm{T}}\mathbf{w} = 1\}$, $\mathbf{e} \in \mathbb{R}^m$ is a vector of all 1s, $\mathbf{A} \in \mathbb{R}^{m\times m}$ is an affinity matrix with each entry $A_{ij} = \exp\big(\frac{\|\hat{\mathbf{s}}_i - \hat{\mathbf{s}}_j\|_2^2}{2\sigma_A^2}\big)$ representing the similarity between $\mathbf{s}_i^{C+1}$ and $\mathbf{s}_j^{C+1}$, σ_A is a scaling factor which is set to the median value of all entries in $\mathbf{A}$. Finally, the estimated target location is obtained by:

$$\hat{\mathbf{s}} = \sum_i w_i \hat{\mathbf{s}}_i. \tag{8}$$

Taking sample confidences c_i into consideration, we slightly modify the affinity matrix $\mathbf{A}$ as:

$$A_{ij}^* = c_i \cdot c_j \cdot A_{ij}. \tag{9}$$

The rest voting process is the same as described before.

3 Updating Scheme

To adapt the model to target appearance variations, we propose an Incremental Cascaded Support Vector Regression (ICSVR) algorithm for online model updating.

Note that the Support Vector Regression (SVR) problem with training samples $\{\mathbf{x}_i, y_i\}_{i=1}^l$ and preset margin ε is equivalent to a Support Vector Classification (SVC) problem with modified training data $\{(\mathbf{z}_i, 1)\}_{i=1}^l$ and $\{(\mathbf{z}_i, -1)\}_{i=l+1}^{2l}$, where $\mathbf{z}_i = (\mathbf{x}_i^{\mathrm{T}}, y_i{+}\varepsilon)^{\mathrm{T}}$ for $i = 1, \cdots, l$ and $\mathbf{z}_i = (\mathbf{x}_i^{\mathrm{T}}, y_i{-}\varepsilon)^{\mathrm{T}}$ for $i = l{+}1, \cdots, 2l$:

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\xi}} \;\; & \frac{1}{2}\|\mathbf{w}\|_2^2 + \eta \sum_{i=1}^{2l} \xi_i, \\ \text{s.t.} \;\; & (\mathbf{w} \cdot \boldsymbol{z}_i) \geq 1 - \xi_i, \;\; i = 1, \cdots, l \\ & -(\mathbf{w} \cdot \boldsymbol{z}_i) \geq 1 - \xi_i, \;\; i = l+1, \cdots, 2l, \\ & \xi_i \geq 0, \;\; i = 1, \cdots, 2l \end{aligned} \tag{10}$$

where η is a regularization parameter. In this way, the incremental learning of SVR can also be implemented by online SVC with slightly modified training samples. We use the work proposed in [22] as the SVC updater in our approach.

As for the cascaded process, in training stage, we collect samples and overlap rates accross C cascades, and train SVRs with samples in corresponding cascades.

4 Experiments

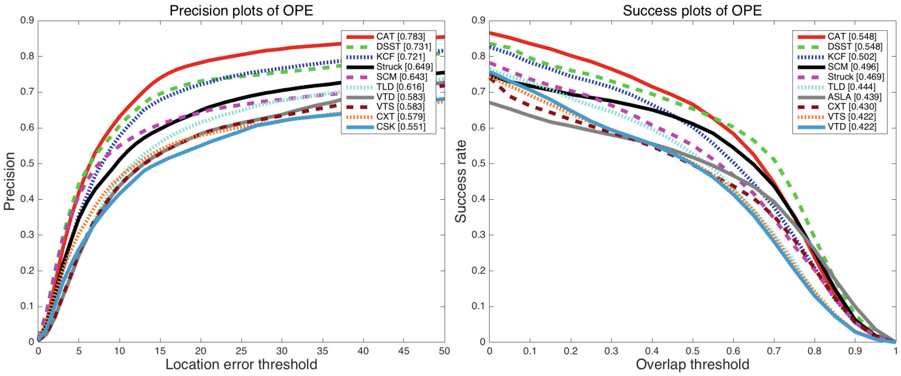
We evaluate our tracking approach on a publicly available benchmark [23], which contains 51 challenging sequences, and compare the performance with 30 trackers, where 28 of which are recommended by [23] including Struck [13], SCM [9], TLD [24], VTD [25], CT [26] and ALSA [27], while the KCF [28] and DSST [29] are recent state-of-the-art trackers.

4.1 Implementation Details

The proposed approach is implemented on MATLAB R2015b and run on a 2.6 GHz Intel Core i5 CPU with 8 GB memory. The code without optimization runs at 3.5 fps in average. Each sampled image is converted to grayscale and normalized to 32 × 32, then HOG feature is extracted on it with bin size 4. For simplicity, we only estimate the target position $\mathbf{s} = \{x, y\}$ and assume the scale and angle of the target stay the same during tracking. In training stage, we sample 200 images around the target with sample radius $r_1 = 8$. $C = 3$ cascades of SVR are trained with regularization parameters $\eta_1 = 0.001, \eta_2 = 0.001$. ε_1 is set to 5 and ε_2 is set to 1. When testing, 400 images are sampled around the last estimated target location with sample radius $r_2 = 20$. The model updating is performed each $T = 5$ frames. All the parameters are fixed for different sequences for fair comparison.

4.2 Overall Performance

The overall performance of our method on the benchmark [23] is illustrated in Fig. 1. We apply the precision plot and the success plot for comparing



(a) **(b)**

Fig. 1. Overall performance of 30 state-of-the-art trackers and our tracker on the benchmark. For clarity, only top 10 trackers are illustrated. (a) Precision scores. (b) Success scores.

performance between different trackers. The precision plot indicates the percentage of frames whose estimated location is within the given threshold distance to the ground truth, while the success plot demonstrates the ratios of successful frames whose overlap rate is larger than the given threshold. The precision score is decided by the score on a selected threshold (20 pixel), and the success score is evaluated by the Area Under Curve (AUC) of each plot. For clarity, only top 10 trackers are illustrated on both plots.

As can be seen from Fig. 1(a) and (b), our method obtains superior performance against others. In the precision plot, our tracker outperforms DSST by 5.2 % and outperforms KCF by 6.2 %. In the success plot, our tracker performs as good as DSST and 4.6 % better than KCF. The DSST employs an accurate scale estimation scheme while our tracker does not estimate the target scale, which makes the DSST obtains competitive performance in the success plot. Overall, our tracker performs competitive or better than state-of-the-arts in terms of both the location accuracy and overlap precision.

The superior performance of our tracker validates the effectiveness of sample voting and the cascaded support vector regression scheme. The cascaded process models the non-linear mapping from features to offsets with iterative linear regressions. In addition, the proposed Incremental Cascaded Support Vector Regression (ICSVR) algorithm provides an effective way for robust model updating, which contributes greatly to the stability of long term tracking.

4.3 Component Validation

This section carries out experiments for verifying the contributions of different components in our method. Three components are evaluated in this section: the dominant set voting, the sample confidence evaluation and the incremental learning of cascaded SVR.

Figure 2(a) compares precision scores among trackers using different voting methods. CAT-DS, CAT-MED and CAT-MS denote the trackers using dominant set voting, (weighted) median voting and (weighted) mean shift voting schemes respectively, where the weights are computed as described in Sect. 2.2.

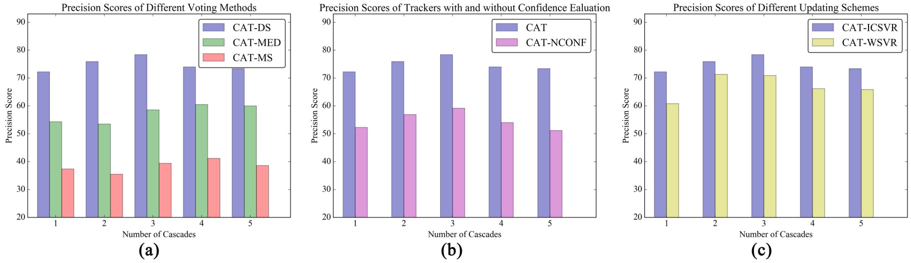


Fig. 2. Validations of different components. (a) Precision scores with and without confidence evaluation. (b) Precision scores of different updating schemes. (c) Precision scores of different voting methods.

As can be seen from Fig. 2(a), CAT-DS significantly outperforms CAT-MED and CAT-MS, which indicates that the dominant set voting is more stable in finding the most densely voted place.

Figure 2(b) compares precision scores between trackers with and without confidence evaluation, namely the CAT and the CAT-NCONF trackers. There's a striking disparity between their scores, which indicates that the sample confidence evaluation is an indispensable part in our method.

Figure 2(c) compares precision scores between trackers using different updating schemes. CAT-ICSVR denotes the tracker using the proposed Incremental Cascaded Support Vector Regression (ICSVR) updating scheme while the CT-WSVR denotes the one using weighted parameter updating scheme (with forgetting factor $\lambda = 0.1$). As illustrated in Fig. 2(c), CAT-ICSVR outperforms CAT-WSVR by 8.8 % when the cascade number is set to 3, which indicates the significant contribution of ICSVR updating algorithm on the tracking performance.

Besides, we can see from the figures that, as the cascade number increases, the performance of our tracker (CAT) steadily rises and reaches the top at 3 cascade, then slightly declines when the number gets larger. This trend reflects the mechanism of SDM and its effectiveness. The SDM models the optimization for a non-linear problem with cascaded linear projections. When the cascade number grows from 1 to 3, the precision score rises since the model fits the data better. Whereas the performance decreases afterwards, which indicates that the SDM gets overfitting on the training data when the model becomes more complex.

References

1. Smeulders, A.W., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: an experimental survey. IEEE Trans. Patt. Anal. Mach. Intell. **36**, 1442–1468 (2014)
2. Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking: a review. Neurocomputing **74**, 3823–3831 (2011)
3. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surv. (CSUR) **38**, 13 (2006)
4. Wang, N., Shi, J., Yeung, D.Y., Jia, J.: Understanding and diagnosing visual tracking systems. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3101–3109 (2015)
5. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**, 121–141 (2008)
6. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1436–1443. IEEE (2009)
7. Zhang, T., Liu, S., Xu, C., Yan, S., Ghanem, B., Ahuja, N., Yang, M.H.: Structural sparse tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 150–158 (2015)
8. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2042–2049. IEEE (2012)

9. Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1838–1845. IEEE (2012)
10. Yao, R., Shi, Q., Shen, C., Zhang, Y., Hengel, A.: Part-based visual tracking with online latent structural learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2363–2370 (2013)
11. Liu, B., Huang, J., Kulikowski, C., Yang, L.: Robust visual tracking using local sparse appearance model and k-selection. IEEE Trans. Patt. Anal. Mach. Intell. **35**, 2968–2981 (2013)
12. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1822–1829. IEEE(2012)
13. Hare, S., Saffari, A., Torr, P.H.S.: Struck: structured output tracking with kernels. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 263–270 (2011)
14. Zhang, J., Ma, S., Sclaroff, S.: MEEM: robust tracking via multiple experts using entropy minimization. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 188–203. Springer, Cham (2014). doi:10.1007/978-3-319-10599-4_13
15. Wang, N., Yeung, D.Y.: Learning a deep compact image representation for visual tracking. In: Advances in Neural Information Processing Systems, pp. 809–817 (2013)
16. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
17. Wang, N., Li, S., Gupta, A., Yeung, D.Y.: Transferring rich feature hierarchies for robust visual tracking. arXiv preprint arxiv:1501.04587 (2015)
18. Xiong, X., Torre, F.: Supervised descent method and its applications to face alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 532–539 (2013)
19. Xiong, X., la Torre, F.D.: Global supervised descent method. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2664–2673 (2015)
20. Wang, X., Valstar, M., Martinez, B., Khan, M.H., Pridmore, T.: Tric-track: tracking by regression with incrementally learned cascades. In: IEEE International Conference on Computer Vision, pp. 4337–4345 (2015)
21. Massimiliano, P., Marcello, P.: Dominant sets and pairwise clustering. IEEE Trans. Patt. Anal. Mach. Intell. **29**, 167–172 (2007)
22. Wang, Z., Vucetic, S.: Online training on a budget of support vector machines using twin prototypes. Statistical Analysis and Data Mining **3**, 149–169 (2010)
23. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2411–2418 (2013)
24. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**, 1409–1422 (2012)
25. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1269–1276 (2010)
26. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7574, pp. 864–877. Springer, Berlin (2012). doi:10.1007/978-3-642-33712-3_62

27. Jia, X.: Visual tracking via adaptive structural local sparse appearance model. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1822–1829 (2012)
28. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Patt. Anal. Mach. Intell. **37**, 583–596 (2014)
29. Danelljan, M., Hger, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference (2015)