Visual Target Tracking using Robust Information Interaction between Single Tracker and Online Model

Yeyi Gu[1], Xinmin Zhou[2,a], Minjie Wan[1,3] and Guohua Gu[1]

[1]*School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China*
[2]*Criminal Police Headquarters of Jiangsu Provincial Security Bureau, Nanjing, China*
[3]*Department of Electrical and Computer Engineering, Laval University, Quebec City, Canada*

Abstract. In this paper, a novel tracking algorithm based on the cooperative operation of online appearance model and typical tracking in contiguous frames is proposed. First of all, to achieve satisfactory performances in challenging scenes, we focus on establishing a robust discriminative tracking model with linear Support Vector Machine (SVM) and use the particle filter for localization. Intended to fit the particle filter, the outputs of SVM classifier are mapped into probabilities with a sigmoid function so that the posterior of candidate samples is estimated. Then, the tracking loop starts with median flow method and the coordinated operation of the two trackers is mediated by the maximum a posteriori (MAP) estimate for the target probability of negative samples, which is defined during the sigmoid fit. Lastly, for the purpose of model update, we sum up the optimal SVM using a prototype set with the predefined budget, and the classifier is updated on both the prototype set and the updated data from the tracking results every few frames. A number of comparative experiments are conducted on real video sequences and both qualitative and quantitative evaluations demonstrate a robust and precise performance of our method.

1 Introduction

Visual object tracking is aimed to estimate the states of a moving target in an image sequence. It serves as an important tool in a variety of vision-based systems designed for video surveillance, autonomous vehicles, human computer interaction, etc. Generally, an excellent tracking strategy should be able to remain stable tracking in complex conditions. For an uncertain tracking task, only the initial location of object given, the limited prior knowledge makes it a challenge to overcome drastic object appearance variations, e.g., illumination change, occlusion, abrupt object motion, and disturbance caused by background clutters.

In recent years, the framework of tracking-by-detection[1-3] has become the mainstream scheme for visual object tracking, where the key is to find the candidate sample that most closely matches the online model. One issue with such a framework is the updating rate[1]. for one thing, highly adaptive online models easily result in drifting in the case of noisy updates. For another thing, stable update will lead to the loss of information from the former contiguous frames and it is difficult to perform well.

To deal with the problems presented above, we present an approach in which a temporary tracker of the median flow algorithm[4] and the online appearance model are independently implemented to exchange information so that a more robust tracking performance can be obtained. An online SVM classifier is built to restrict the temporary tracking by the median flow method and the tracker can provide new samples for the model update. The proposed method combines the context model information with the contiguous appearance information and it can effectively alleviate the model update problems, which is closely related to the drifting problem and the model adaptability to appearance change.

2 Online model embedded particle filter

2.1 Object appearance model using online linear SVM

To adapt to the appearance variations, it is necessary to construct an online object model. In our paper, we use the SVM algorithm[5] to train a classifier. For SVM models, new examples are predicted with a signed distance as

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \Phi(\boldsymbol{x}) + b \qquad (1)$$

where, $\boldsymbol{w}$ is the weight vector; b is the bias threshold; $\boldsymbol{x}$ is an M-dimensional input vector, and Φ is the mapping from the original input space to the feature space $\mathcal{H}$.

For linear SVM, the model is trained with a dataset $\mathcal{D}=\{(\boldsymbol{x}_i, y_i), i=1,...,l\} \in \mathbb{R}$, where, l denotes the number of samples and a binary label $y \in \{-1,+1\}$, by optimizing the following function:

[a] Corresponding author: zhouxinminjs@sina.com

$$\min_{w,b}\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_i L[y_i, f(\boldsymbol{x}_i)] \qquad (2)$$

Where, L is the hinge loss function and C is the penalty factor user-specified that balances the model complexity and the loss on datasets.

To initialize the classifier, we sample N_{object} positive examples and $N_{background}$ negative examples randomly from patches spreading all over the image. In addition, those image patches are represented by color and texture histograms. The color histogram is calculated in the RGB space under 3*3*3 bins, which is defined as $\boldsymbol{p}$={$p^{(n)}$, n=1,...,27} . To provide robustness for drastic illumination variations, local binary pattern (LBP) histogram in (8,1) neighbourhood using uniform rotation invariant LBP is combined. The histogram vector of LBP is defined as $\boldsymbol{q}$={$q^{(n)}$, n=1,...,10} , and the new feature vector of each instance is denoted as $\boldsymbol{x}=\{\boldsymbol{p},\boldsymbol{q}\}$. To obtain nonlinear decision boundaries with linear SVM, the mapping method mentioned in paper[6] is applied to approximate the min kernel SVM, and a single 1850-dimensional vector is calculated for the classifier training.

2.2 State estimation by particle filter algorithm

Particle filter[7], also known as condensation algorithm, has been proved an effective framework in non-linear and non-Gaussian tracking problems. In this paper,the particle filter algorithm works as the motion model to determine the position.

In our study, $\boldsymbol{s}_t^* = [x, y, s, r]^T$ is defined to describe the state of target of interest, consisting of the 2D image position (x, y), the lateral scale s, and the aspect ratio r. $Z_t = \{z_1,..., z_t\}$ denotes all the observations. A weighted sample set $S_t = \{(\boldsymbol{s}_t^{(n)}, \pi_t^{(n)}), n = 1,..., N_{particle}\}$ is used to approximate the probability distribution, where $N_{particle}$ denotes the total number of particles. Each $\boldsymbol{s}$ is a hypothetical state and π is the corresponding importance weight. Each state is weighted in terms of the likelihood of the observation as

$$\pi_t^{(n)} = p(z_t \mid \boldsymbol{s}_t^* = \boldsymbol{s}_t^{(n)}) \qquad (3)$$

The particle set is generated by sampling from the proposal distribution $S_t^{(n)} \sim p(S_t^{(n)} \mid S_{0:t\text{-}1}^{(n)})$. In our case, a multivariate Gaussian distribution is employed to draw the particle set, and a Brownian motion is used as the state dynamics model. Especially, the scale term is replaced with a uniform distribution following the power function with L_1 measurement, reducing the algorithm complexity and usage of particles to a certain extent.

Moreover, with the assumed distribution above, the state of the object is finally estimated by the mean value E[S]:

$$\mathrm{E}[S] = \sum_{n=1}^{N}\pi^{(n)}\boldsymbol{s}^{(n)} \qquad (4)$$

Generally, the likelihood $p(z_t \mid \boldsymbol{s}_t^* = \boldsymbol{s}_t^{(n)})$ is measured by the Bhattacharyya distance between the target template and the particle patches. In our case, the decision value output $f(\cdot)$ from the classifier is associated with the weight π . Nevertheless, the output of the SVM is not an actual probability, but the distance of the candidate patches to the separating hyperplane in the feature space. Platt et. al [8] proposed to scale the output to the range of [0,1] by a sigmoid function and a further revision is carried on in [9], aiming at flaws such as the "catastrophic collapse" problem. Overall, the posterior $Pr(y=1 \mid \boldsymbol{x})$ is built as

$$Pr(y=1 \mid \boldsymbol{x}) \approx \begin{cases} \dfrac{\exp(-Af-B)}{1+\exp(-Af-B)} & if \quad Af+B \ge 0 \\ \dfrac{1}{1+\exp(Af+B)} & \text{else} \end{cases} \qquad (5)$$

where, f is evaluated in three-fold cross-validation of the SVM, and the optimum parameters A^* , B^* are calculated by solving the following regularized maximum likelihood problem:

$$\min_{A,B} -\sum_{i=1}^{l}(t_i \log(p_i) + (1-t_i)\log(1-p_i)) \qquad (6)$$

where,

$$-(t_i \log(p_i) + (1-t_i)\log(1-p_i))$$
$$= \begin{cases} (t_i-1)(Af_i+B)+\log(1+\exp(Af_i+B)) & if \quad Af_i+B \\ t_i(Af_i+B)+\log(1+\exp(-Af_i-B)) & \text{else} \end{cases}$$
$$(7)$$

Meanwhile, in consideration of the sparsity of the sigmoid function, the MAP estimate for the target probability of positive examples and negative are defined as

$$t_+ = \frac{N_+ +1}{N_+ +2} \quad t_- = \frac{1}{N_- +2} \qquad (8)$$

According to Eq.(4), the state of the target can be estimated as :

$$\mathrm{E}[S] = \sum_{n=1}^{N} Pr(y=1 \mid \boldsymbol{x}_t^{(n)})\boldsymbol{s}_t^{(n)} \qquad (9)$$

3 Cooperative operation of double trackers

As is stated above, a robust algorithm integrating the particle filter algorithm and the online SVM classifier is given, key of which is an appearance model summarizing previous observations. Generally, the online update is an important part. To balance the model drift problem and the model adaptability, a new target localization and model update strategy is proposed.

3.1 Target localization

The proposed algorithm is started by training an initial classifier as introduced in section 2.1. Then, the tracking loop starts with the median flow method. The Median Flow tracker is proved to be stable in temporary tracking and the calculation is faster. What is more, the quality of point prediction is then estimated and each point is assigned with an error value, e.g., Forward-Backward (FB) Error[4], Sum of Squared Differences (NCC), normalized cross correlation (SSD). Only half of the points with low error values are used to estimate the whole bounding box:

$$\Delta \boldsymbol{x}_n = median(\Delta X_n) \tag{10}$$

$$s_n = median(d_n ./ d_{n-1}) \tag{11}$$

where $\Delta \boldsymbol{x}=(\Delta x, \Delta y)^T$ is the predicted displacement of the whole bounding box, defined from the remaining points in n-th frame using median over each spatial dimension $\Delta X_n=\{(\Delta \mathrm{x}_n^{(i)}, \Delta \mathrm{y}_n^{(i)}), i=1,...,N_{point}\}$, and s_n is the scale change computed as the median ratio between the current point distance d_n and the previous point distance d_{n-1} for each pair of remained point.

However, the problem of the native strategy is the drifting. Once there is a failure, the tracking will end up. Our tracking loop starts with the Median Flow tracking, and tracking result for every frame is evaluated by the trained classifier. During the trajectory of Median Flow, results with low FB error and probabilistic output higher than λt_- are adopted as new positive samples. t_- is the MAP estimate for the target probability of negative examples in Eq.(8) and the parameter λ is defined as the compensation for incompleteness of samples.

When the error of Median Flow is higher than the given threshold and the probabilistic output closely approximates that of the negative examples, the learning embedded particle filter is used for reliable tracking.

3.2 Model update

The tracking results will be added to the new dataset and N_{nneg} patches are randomly sampled from the surrounding region outside of the bounding box as the new negative examples. Besides, the model will be updated once the learning embedded particle filter will be used for reliable tracking or the interval reaches the predefined value Δ_{update} .

To update the model, we expand the SVM by combining a prototype set $\mathcal{Q}=\{\Phi(\boldsymbol{x}_i), \omega_i, s_i\}_1^B$ with new datasets $\mathcal{L}=\{\boldsymbol{x}_i, y_i\}_1^J$, where B is a predefined bound of prototype sets and J is the number of new samples.

The prototype set is positioned near the decision boundary to summarize the previous training data, where $\boldsymbol{x}_i$ is the input vector; ω_i is the binary label, and s_i is the counting number of the support vectors represented. Further, the classifier optimal function in Eq.(2) can be improved as

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\{\sum_{i=1}^{B} \frac{s_i}{W_{\omega_i}} L_h(\omega_i, \boldsymbol{x}_i; \boldsymbol{w}) + \frac{\alpha_{y_i}}{W_{y_i}} \sum_{i=1}^{J} L_h(y_i, \boldsymbol{x}_i; \boldsymbol{w})\} \tag{12}$$

$$W_+ = \sum_{w_i=+} s_i + \sum_{y_i=1} \alpha_+ \quad W_- = \sum_{w_i=-} s_i + \sum_{y_i=-1} \alpha_- \tag{13}$$

where, α_+ and α_- are the learning rate of the new positive samples and that of the new negative samples; W_+ and W_- are the total weight of the positive samples and that of the negative samples.

In this scheme, the imbalance of the negative and positive instances is considered by the total weight W . The instances of prototype set and new dataset have different influence on the training which can be adjusted by the learning rate α .

New members of the prototype set will arise after updating with counting number 1. Once, the size of the prototype set is over the predefined budget B , the components of the same label with the minimal distance are merged by:

$$\Phi(\boldsymbol{x}^*) = \frac{s_{i_1}\Phi(\boldsymbol{x}_{i_1}) + s_{i_2}\Phi(\boldsymbol{x}_{i_2})}{s_{i_1} + s_{i_2}} \quad s^* = s_{i_1} + s_{i_2} \tag{14}$$

In our experiment, we use $C=100$, $B=50$, $\alpha_- = 1$, and the learning rate of positive sample are adaptive by the counting number of the positive ones in the prototype set.

4 Experiments and results

Intended to demonstrate the effectiveness of the proposed method, both qualitative and quantitative experiments are implemented on 8 publicly available challenging image sequences. These sequences contain complex scenes with challenging factors for visual tracking. For comparison, we run 7 state-of-the-art algorithms. These algorithms[10] include: the online multiple instance learning (MIL) , L1 accelerated proximal gradient (L1APG), online Adaboost boosting (OAB), tracking by detection (TLD), Struck, Semi-supervised online boosting (SemiT) , sparsity-based collaborative model (SCM).

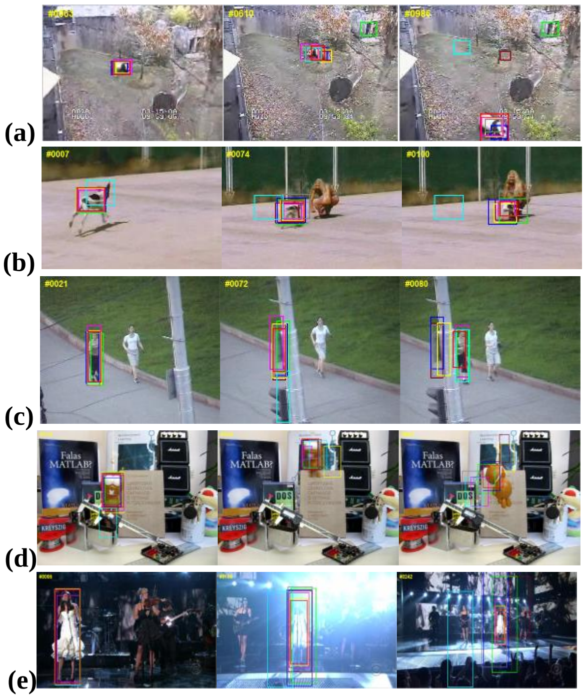
For our tracker, 20 positive examples ($N_{object}=20$, $IOU > 0.8$) and 600 negative examples ($N_{background}=600$ $IOU < 0.2$) are sampled to initialize the classifier, and 100 negative examples ($N_{nneg}=100$) are added to update the classifier from the slightly overlapped patches ($IOU < 0.2$) around the estimated position within 100 steps. In addition, we set $\Delta_{update}=10$, $\lambda=3$ for the whole process, and the sample number of particle filter is set as 300 ($N_{particle}=300$). All the experiments are implemented by Matlab 2012a and Visual Studio 2010 softwares on a PC with a 2.59 GHz INTEL CPU and 8 GB memory.

4.1 Qualitative Comparison with other methods

In this section, we qualitatively compare our performance with the other 7 state-of-the-art trackers.

As shown in Figure 1, the tracking results are displayed using bounding boxes with different colour.In the sequences ‘Panda’, ‘Dog’ and ‘Jogging-1’, the targets undergo severe pose variety; In the sequences ‘Panda’, ‘Singer1’ and ‘Carscale’, there are distinct scale variations; In the sequences ‘Dudek’, ‘Rubik’ and ‘Panda’, the appearances of the targets change with drastic in-plane rotations and out-of-plane rotations; In the sequences ‘Singer1’, ‘Panda’, ‘Rubik’ and ‘Carscale’, the tracking is disturbed by partial occlusion. These proposed factors have serious impacts throughout the sequences and our scheme succeeds in all the experiments, showing good robustness in severe pose changes, large scale variations, rotation, and partial occlusion and it results from the robust discriminative model using the global features of RGB and LBP histograms.

On the other hand, the other 7 state-of-the-art trackers have different performances. When light shines on the object just like the video clip of ‘Singer1’, LIAPG and SemiT methods fail to distinguish the singer; In the sequence ‘Jogging-1’, SCM, LIAPG, Struck and MIL lose the target when occlusion occurs while others can keep tracking the jogger; The ‘Panda’ video contains significant pose change and rotation. OAB, SemiT and L1APG trackers obviously cannot run so well, and TLD fails in some cases; In ‘Lemming’, TLD, SCM, L1APG, SemiT lose the object when abrupt motion occurs and when there is drastic out-of-plane rotation, only our scheme and MIL method perform well; In the respect of scale, our algorithm and SCM, L1APG realize the scale adaptation under the particle filter framework. Struck and TLD adopt the approach of multiscale traversal search. OAB, MIL and SemiT do not take the scale into consideration.



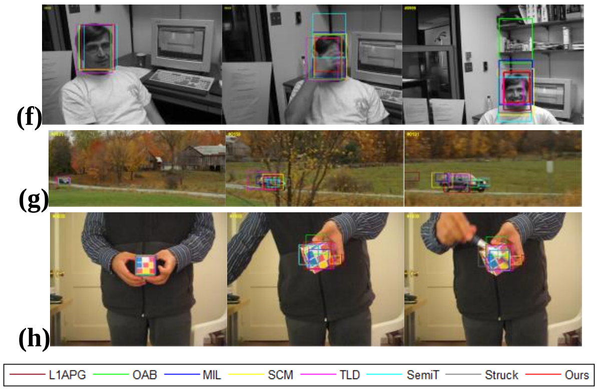
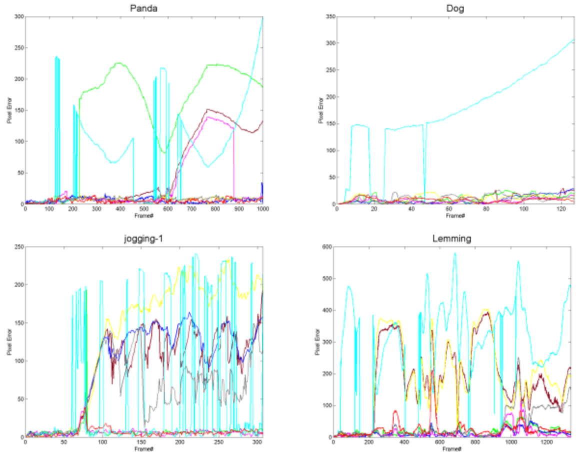


Figure 1. Tracking results of 8 trackers on 8 test sequences, (a) “Panda”; (b) “Dog”; (c) “Jogging-1”; (d) “Lemming”; (e) “Singer1”; (f) “Dudek”; (g) “Carscale”; (h) “Rubik”. (Results are best viewed on high-resolution displays.)

4.1 Quantitative Comparison with other methods

In this part, we adopt the centre location error to quantitatively evaluate the performance in each frame for all test sequences.

The overall performance of different algorithms for sequences are shown in Figure 2 by the graph of centre offset value across all frames of each image sequence. The centre location error is defined as the Euclidean distance between the centre of the tracking result and the ground truth for each frame. As we can see from the Figure 2, the proposed tracker performs remarkably almost in the whole frames for all sequences. The TLD tracker also performs well on most videos, because it uses a detector integrated with a cascade of three classifiers (i.e., patch variance, random ferns, and nearest neighbour classifiers) for tracking and the lost tracking can be restarted via redetection. Furthermore, other trackers merely perform well in part of the test sequences.



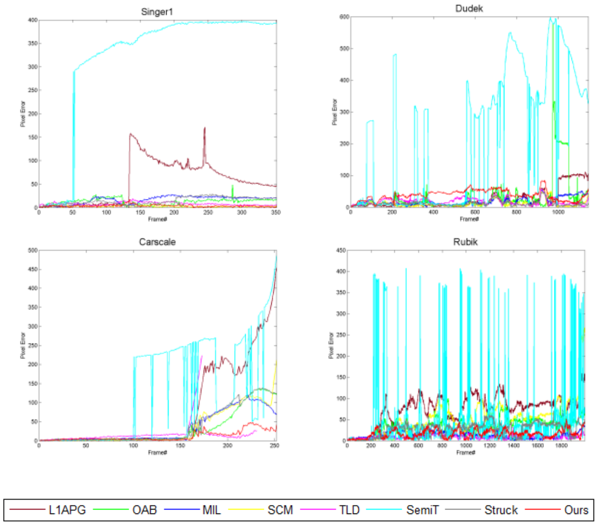


Figure 2. the graphs of pixel error.

5 Conclusion

In this paper, a robust visual tracker based on an integrated framework is introduced. Our approach combines a typical tracker based on optical flow and the online learning embedded particle filter tracker. The two trackers work in coordination and exchange information, balanced by the MAP estimate value for the target probability of negative examples. Furthermore, the appearance model is updated every few frames with a prototype set and the new dataset. The prototype set sums up the previous model and the new dataset provides updated information. The combination of these ideas leads to a precise and flexible tracker that is able to quickly applicable to the tracking of arbitrary objects in unknown environments. Both the qualitative and quantitative evaluations demonstrate that the proposed tracker can overcomes various challenging interferes and achieves stable and robust performance in long-term tracking.

In future work, much attention should be paid to setting up a more reliable appearance model, such as the combination of integral images and local histograms or a more distinct feature, so that the tracking accuracy can be further improved.

Acknowledgments

This paper is financially supported by the National Natural Science Foundation of China (61675099), Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2016-2-07) and China Scholarship Council.

References

1. C. Ma, X. Yang, C. Zhang, M.H. Yang, *CVPR*(2015)
2. M. Wan, G. Gu, W. Qian, K. Ren, Q. Chen, H. Zhang, X. Maldague, Remote Sensing, **10**, 510(2018)
3. Z. Kalal, K. Mikolajczyk, J. Matas, IEEE trans. Patt. Anal. Mach. Int., **34**, 1409-1422(2012)
4. Z. Kalal, K. Mikolajczyk, J. Matas, *ICPR*(2010)
5. S. Avidan, IEEE trans. Patt. Anal. Mach. Int., **26**, 1064-1072(2004)
6. S. Maji, A.C. Berg, *ICCV* (2009)
7. K. Nummiaro, E. Koller-Meier, L. Van Gool, Image Vis. Comput., **21**, 99-110(2003)
8. J. Platt, Advances in Large Margin Classifiers, **10**, 61-74(1999)
9. H.T. Lin, C.J. Lin, R.C. Weng, Machine Learning, **68**, 267-276(2007)
10. Y. Wu, J. Lim, M.H. Yang, IEEE trans. Patt. Anal. Mach. Int., **37**, 1834-1848(2015)