2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Visual Tracking by Combining the Structure-Aware Network and Spatial-Temporal Regression

Dezhong Xu,Lifang Wu,Meng Jian,Qi Wang
Faculty of Information Technology
Beijing University of Technology
Beijing, China, 100124
xudezhong@eamils.bjut.edu.cn, lfwu@bjut.edu.cn, mjian@bjut.edu.cn, 14020024wq@emails.bjut.edu.cn

Abstract— **In this paper, we propose a novel visual tracking algorithm by combining the structure-aware network (SA-Net) and spatial-temporal regression model. We first use SA-Net to obtain the initial location proposal, and the deep features are extracted using a fine-tuned convolutional neural network model. Finally, both the location proposal and deep features, including historical information, are input into the long short-term memory (LSTM) for end-to-end spatial temporal regression to adjust the initial location proposal from SA-Net. The experimental results on the challenging OTB dataset demonstrate that the proposed scheme is robust to missing tracking caused by occlusion or object deformation. Additionally, the compared experiments show that the proposed scheme is more competitive than state-of-the-art algorithms.**

Keywords—spatial and temporal regression; LSTM; SA-Net; occlusion; object deformation

I. Introduction

Visual object tracking can be applied in many areas, such as video surveillance, automatic driving, and human computer interaction. Although visual tracking has developed rapidly in recent years, it is still a challenging task in computer vision because of aspects such as occlusion, deformation, and motion blur.

With the development of high-performance computation hardware and the appearance of large-scale datasets, deep learning has caused the rapid development of the object detection task. Compared with hand-designed features, the features from deep learning can extract rich information from objects. They are popularly used in recent years. Nam et al. [4] proposed a multi-domain network (MD-Net) to learn the shared presentation of an object from multiple annotated video sequences used for tracking. However, it is sensitive to similar distracters because the trained convolutional neural network (CNN) model mainly focuses on inter-class classification. To address this problem, Fan et al. proposed the structure-aware network (SA-Net) [3] for visual tracking. Different from traditional CNN models, SA-Net uses RNNs to model the self-structure of an object during learning. It makes the model effective for discriminating not only different objects but also similar distractors in the background. However, SA-Net is a local tracking algorithm: it uses only the information in the previous frame. Whenever objects are missed because of occlusion or object deformation, it is difficult to recover the real object and this will result in a tracking failure gradually. Therefore, tracking accuracy is not very high.

To address the problem of tracking failures, some researchers [7-9] have used generative or discriminative models to discriminate the foreground from the background using handcrafted features. Recently, increasing numbers of trackers have used features learned from deep learning [10-11]. However, these methods pay more attention to the robustness of deep features against handcrafted features. They do not extend the deep neural network to the spatial temporal domain.

It is possible to recover tracking failures by learning from historical visual information and tracking proposals. Inspired by this, we propose a visual tracking algorithm that uses a recurrent CNN. Both the location proposal from SA-Net and deep features, including historical information, are input into the long short time model (LSTM) to adjust the initial location. Extending SA-Net into spatial and temporal domain has not been studied. The experimental results confirm the efficiency of the proposed algorithms.

The contributions of this paper are as follows:

1. We propose an efficient visual tracking algorithm that combines SA-Net, and spatial and temporal information to predict the location of objects. It is robust to occlusion and object deformation.
2. We propose a LSTM-based regression model with feature regulation that combines rich local information and historical information so that missing tracking can be corrected as soon as possible. Additionally, the accuracy of visual tracking can be improved.

978-1-5386-3788-3/18/$31.00 ©2018 IEEE

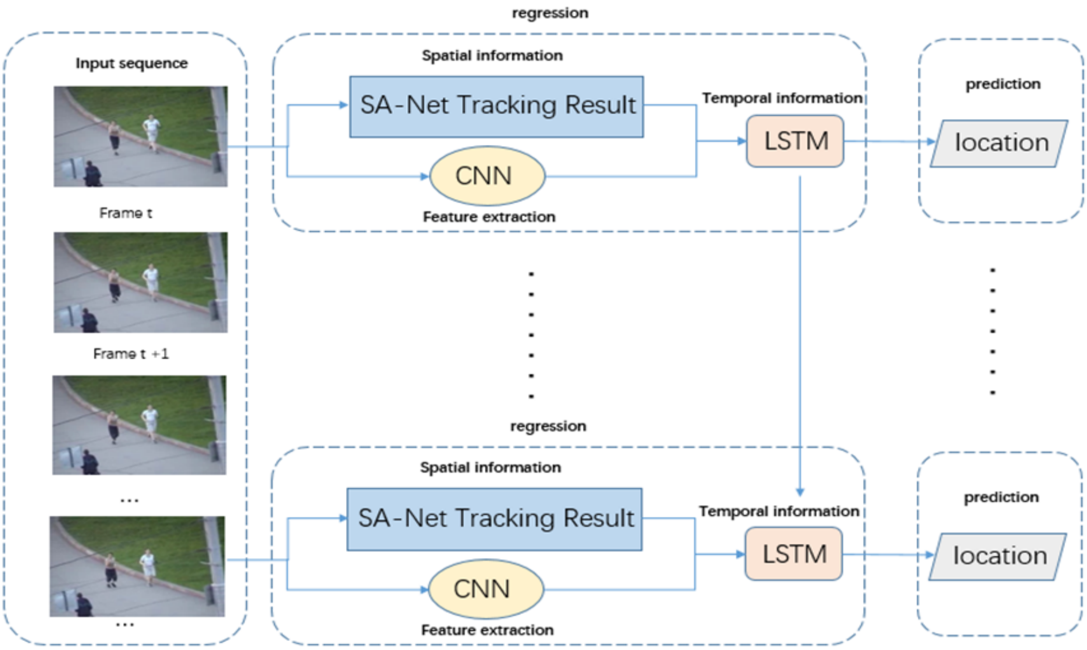


Fig. 1. Framework of the proposed scheme

II. Proposed Scheme

The framework of the proposed scheme is shown in Fig. 1. We first extract the features of each frame using the CNN network. Then, we obtain the initial location of each object using SA-Net. Finally, the LSTM is used for spatial and temporal regression based on the initial location from SA-Net, the deep features from CNN, and historical information.

A. Feature Extraction using the CNN model

Many deep neural network models exist for feature extraction, such as VGG [18], AlexNet[19], and GoogLeNet[20]. GoogLeNet is typically used in object detection; therefore, it is more suitable for the bounding box regression task than other models. Therefore, GoogLeNet is selected for our study. It is first pre-trained using the ImageNet dataset, and then fine-tuned using the VOC dataset. The output of the first full connection layer is used as the feature vector; the dimension of the feature vector is 4,096. Finally, it is regularized. The extracted spatial features are denoted by B_t.

B. Initialize the Object Location using SA-Net

In this section, the structure of the object is represented as four directed cyclic graphs: $G^u = \{G_1, G_2, G_3, G_4\}$.Additionally, the summary of the hidden layers are fed into the output layer. It is represented as follows Eq.(1)-(2):

$$\begin{cases} h_m^{(v_i)} = \phi'\left(U_m \chi^{(v_i)} + \sum_{v_i \in P_{G_m(v_i)}} W_m h_m^{(v_i)} + b_m\right) & (1) \\ y^{(v_i)} = \sigma\left(\sum_{G_m \in G^u} V_m h_m^{(v_i)} + \phi'\left(h_m^{(vi)}\right)\right) & (2) \end{cases}$$

where $\Phi(\cdot)$ and $\sigma(\cdot)$ are non-linear activation functions; U_m, W_m, and V_m are matrix parameters; b_m is the bias term for G_m; $P_{Gm}(v_i)$ is the predecessor set of v_i in G_m; and $h_m^{(v_i)}$, $y^{(vi\ )}$are hidden layer and output layer at v_i respectively. The error is back-propagated to the previous convolutional layer v_i and computed as follows Eq.(3):

$$\nabla_x^{(vi)} = \sum_{Gm \in G^u} U_m^T d_{hm}^{(vi)} \circ \phi'\left(h_m^{(v_i)}\right) \tag{3}$$

where ∘ is Hadamard product.

Using the trained SA-Net, we obtain the bounding box of the object for each frame, which is represented as Eq.(4)

$$L_t = (x, y, w, h) \tag{4}$$

where $(x,\ y)$ represents the coordinates of the center of the bounding box, w is the width of the bounding box, and h is the height of the bounding box. To be in accordance with the spatial features extracted from the CNN model, x, y, w, and h are normalized in the range of [0,1].

C. Regression Model Training on LSTM

Long short-term memory (LSTM) networks were described in [17]. The main innovation of the LSTM model is a new structure called a memory cell, which can store information for a long period of time. A memory cell is composed of four main elements: an input gate, neuron with a self-recurrent connection, forget gate, and output gate. Whenever a new input arrives, if the input gate is activated, its information is remembered. Meanwhile, if the forget gate is activated, the past cell status c_{t-1} could be forgotten. The final state h_t is controlled by the output gate o_t, and the latest cell c_t is output. The nonlinear sigmoid equation is denoted by σ. The sigmoid layer outputs numbers between zero and one, which describe how much of each component should be let through. x_t is the current frame, h_{t-1} is the hidden status at time *t-1*, c_{t-1} is the cell status at time *t-1* and . The LSTM network at time *t* is updated as follows Eq.(5)-(9):

$$i_t = \sigma\left(W_{x_i} X_t + W_{h_i} h_{t-1} + b_i\right) \quad (5)$$

$$f_t = \sigma\left(W_{x_f} X_t + W_{h_f} h_{t-1} + b_f\right) \quad (6)$$

$$o_t = \sigma\left(W_{x_o} X_t + W_{h_o} h_{t-1} + b_o\right) \quad (7)$$

$$c_t = f_t c_{t-1} + i_t \sigma\left(W_{x_c} X_t + W_{h_c} h_{t-1} + b_c\right) \quad (8)$$

$$h_t = o_t \circ tan(c_t) \quad (9)$$

In the proposed method, the spatial features *Bt* and initial location L_t are connected to form vector $X_t=\{B_t, L_t\}$. To make the position of the target easier to regress, a regularization factor for b_t is introduced. Both X_t and the output of the previous state S_{t-1} are input into the LSTM model, which is trained using the mean square error and adds the L_2 regularization term as follows Eq.(10):

$$Loss_{MSE} = \frac{1}{n}\left(\sum_{i=1}^{n}\left\|L_{t\arg et} - L_{pred}\right\|_2^2 + \lambda\varphi\left(B_t\right)\right) \quad (10)$$

where *n* is the number of training samples in each batch, L_{pred} is the prediction model, L_{target} is the ground truth of the location, ||.|| is the square of the Euclidean distance, $\varphi(B_t)$ is the regularization item, and λ is the penalty factor.

D. Regression by Combining Spatial and Temporal Information

Based on the tracking task, we implement the regression model in spatial and temporal domains using LSTM's characteristics. The regression is implemented in single and multiple neural units. For the regression in a single neural unit, the visual features and local representation are linked using LSTM. Then, the location is inferred based on visual features. For regression in multiple neural units, sequential features from video sequences are used to predict the visual features of the next frame. In the procedure, LSTM can efficiently present spatial and temporal information completely using the visual features and location of the object.

In general object detection algorithms [5,6], the regression model was used for all candidate bounding boxes. The initial location was used as spatial supervised information for prediction. It is helpful to regress the visual features to the location of an object. In our work, LSTM can learn sequential information in a long time period, and temporal information can be preserved efficiently so that the range of the prediction location is constrained.

III. EXPERIMENTS

A. Experimental Environment

The OTB dataset was used to evaluate the proposed algorithms. OTB-100 [15] is a public dataset for object tracking. It includes 100 challenging videos; the objects in these videos are labeled in a bounding box, frame by frame. Images from example videos are shown in Fig. 2.

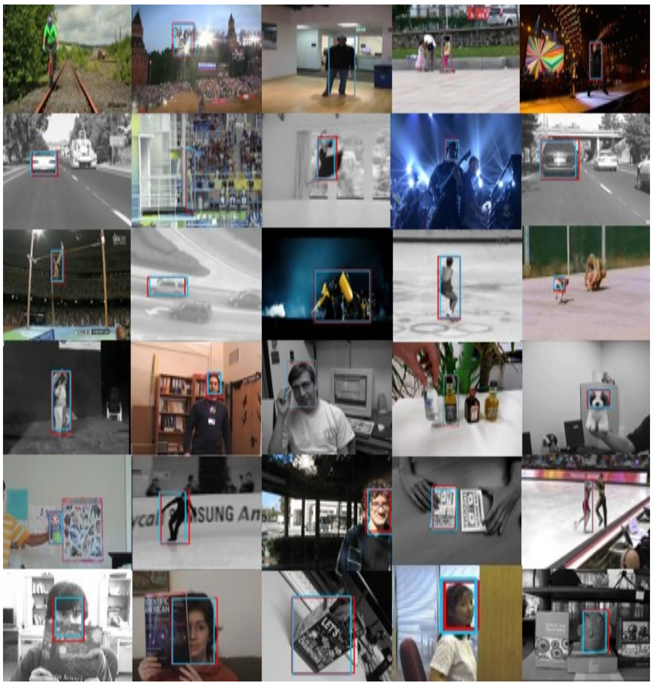


Fig. 2. Images from example videos in the OTB-100 dataset

The visual tracking results were evaluated using precision plots (PP) and success plots (SP).

PP is used to measure the overall tracking performance. It shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth.

For SP, given the tracked bounding box l_t and ground truth box l_a, the overlap score is defined as S = $|l_t \cap l_a| / |l_t \cup l_a|$. The number of successful frames whose overlap S is larger than the given threshold is counted, and SP is the ratio of successful frames to the total number of frames.

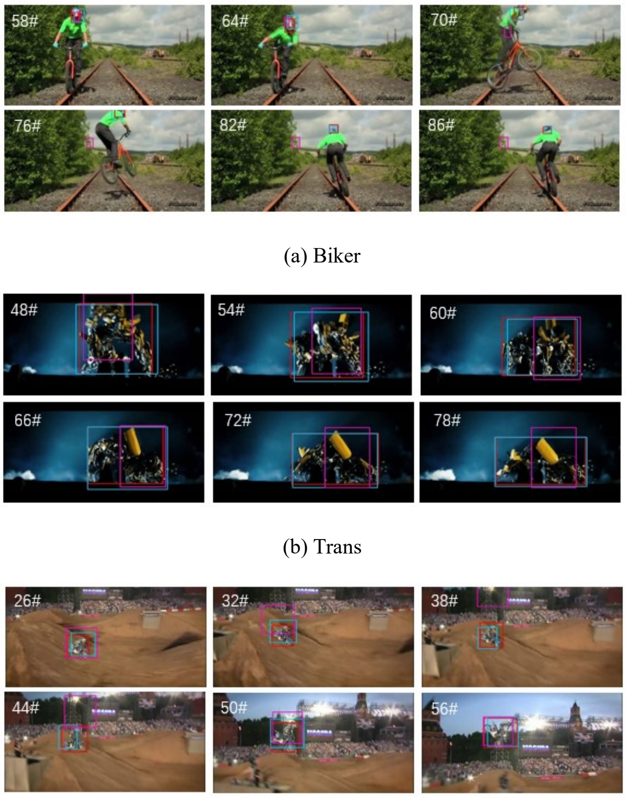
Deep learning algorithms were implemented on the TensorFlow framework using Python. They were run on a computation server that included two NVIDIA Tesla K80 GPUs.

B. For Missing Tracking

In this section, we compare the efficiency of the proposed scheme and SA-Net for missing tracking. Fig. 3 shows images from five example videos. In this figure, the ground truth is indicated by a red rectangle, the results of SA-Net are indicated a pink rectangle, and our results are indicated by a blue rectangle. In Fig. 3(a)–(c), there are object deformations because the motion direction or object changes, whereas in Figs. 3(d) and (e), the objects are occluded by other moving or static objects.

In Fig. 3(a), a person is riding toward the camera at the beginning. From frame 1–58–64, both SA-Net and our scheme tracked the object well. In frame 70, the rider turned around and rode backward far from the camera. As the appearance of the object varied more, SA-Net gradually missed the object, whereas our algorithm continued tracking the object stably. In Fig. 3(b), the object changes from a robot to a car; there is also a large appearance variation for the object. Based on local similarity, Sa-Net tracked the local region, whereas our scheme tracked the object completely. Fig. 3(c), shows a person riding on a soil slope. In frame 26, both methods were good. From frame 32–44, affected by lighting and background changes, SA-Net missed the object, but our method could still track the object accurately

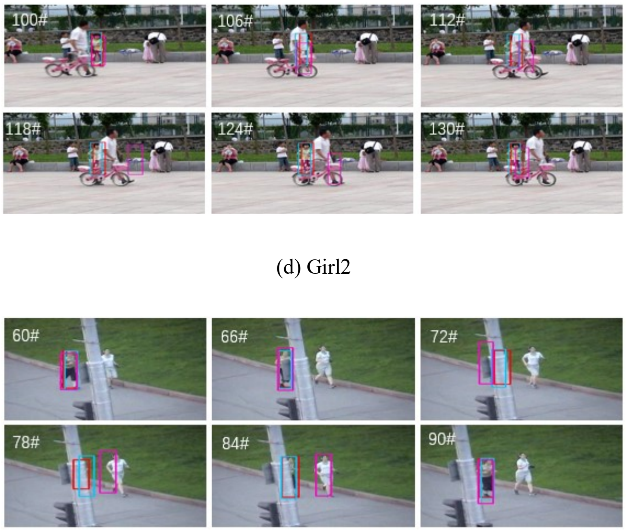
In Fig. 3(d), a girl is occluded by a person riding a bike in frame 106. The features of the person were extracted as the object features for SA-Net. From this frame, SA-Net falsely tracked the person riding the bike and not the girl. By comparison, our algorithm always tracked the object accurately. A similar scenario is shown in Fig. 3(e), in which the tracking object is a running person that is occluded by a column in frame 72. From then, SA-Net missed the tracking object completely, whereas our algorithm recovered the true object once the person appeared.



(a) Biker

(b) Trans

(c) MotorRolling



(d) Girl2

(e) Jogging_1

Fig. 3. Comparison of tracking results for object deformation and occlusion

SA-Net represented the rich structure of the object, but it did not consider more long-term information. Our algorithm used the LSTM to combine the historical information of the object. It obtained robust sequential features and smoothed spatial supervised information. It connected the location and visual features in the spatial and temporal domains, and effectively used the historical location of the object. It had the advantage of both SA-Net and LSTM. It was robust to missing tracking caused by occlusion or object deformation, as demonstrated in Fig. 3.

C. *Comparison Experiments*

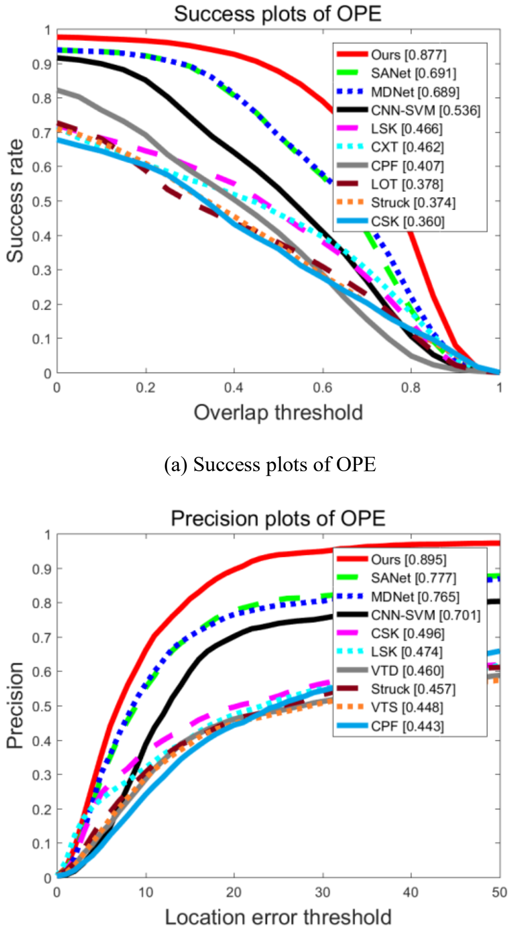
In this section, we compare the proposed algorithm with state-of-the-art algorithms: SA-Net[3], MD-Net[4], Struck[12], TLD[9], OAB[13], CXT[7], CSK[8], VTD[14], and CNN-SVM[10].

Experiment C-1

In the visual tracking algorithms, limited data was one of challenging problems. To evaluate the generality of the proposed algorithms, we selected 60 videos from OTB-100 as the training set. Additionally, we selected 10 other videos as a testing set. The SPs and PPs of OPE are shown in Fig. 4, in which we observe that the proposed scheme was better than other algorithms in all cases. The performance of SA-Net and MD-Net were slightly worse than that of our scheme, but much better than that of the other algorithms. Our algorithm improved the average SP and PP by 0.186 and 0.118, respectively, compared with SA-Net the best state-of-the-art algorithms.

Experiment C-2

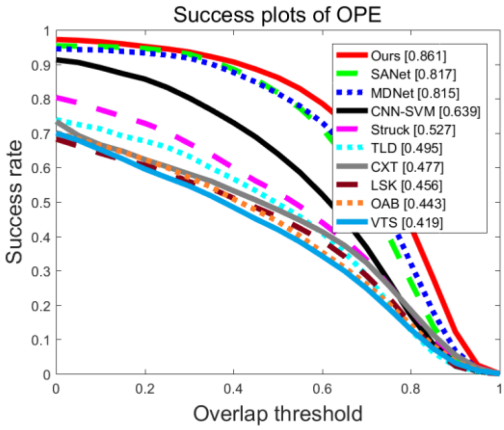
We further evaluated how dynamic auxiliary frame training influenced performance. Referring to the experiments in Ref [16], we randomly selected 1/6 frames from 60 videos (the same as the training set in Experiment C-1) for training, and the remaining 5/6 frames from the 60 videos were used for testing. The compared experimental results are shown in Fig. 5, in which we observe that the dynamic information further improved the performance of visual tracking. This demonstrates that the introduction of dynamic auxiliary information further improved performance in the supervised training environment.



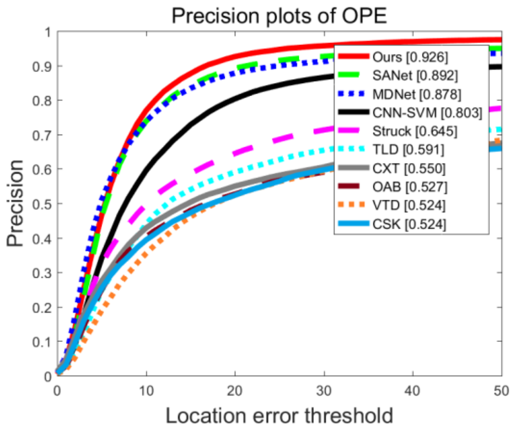
(a) Success plots of OPE

(b) Precision plots of OPE

Fig. 4. Compared results on generality



(a) Success plots of OPE

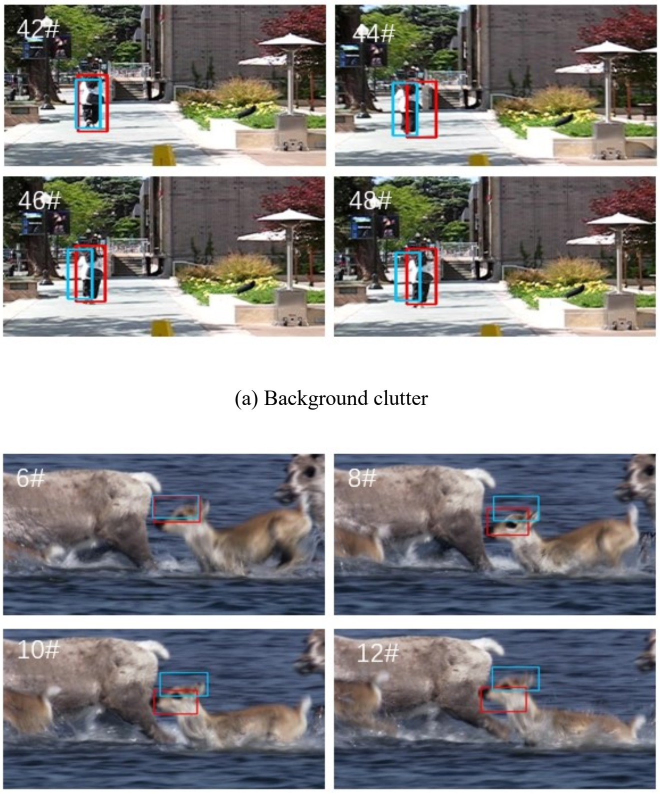


(b) Precision plots of OPE

Fig. 5. Compared results on the influence of dynamic auxiliary information

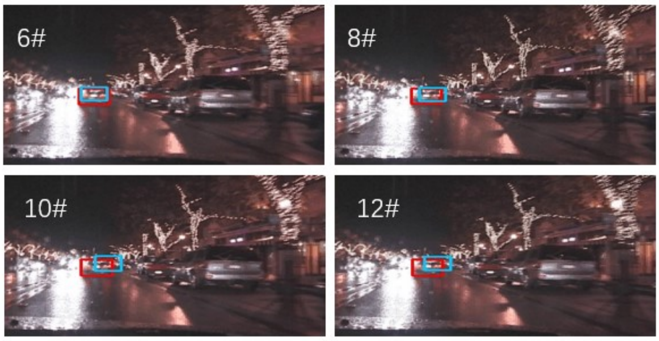
D. *Failure Cases*

Experiments in Sections B and C demonstrated that the proposed algorithm improved a great deal regarding tracking performance for occlusion and object deformation, However, the proposed algorithm did not work in some cases, such as background clutter, motion blur, and low-resolution objects. Some examples are shown in Fig. 6.



(a) Background clutter

(b) Motion blur



(c) Low-resolution object

Fig. 6. Failure examples

In Fig. 6 (a), the background is similar to the color of the object's clothes; therefore, the background features became increasingly dominant, which caused missing tracking. In Fig. 6(b), motion blur causes inaccurate features, which resulted in missing tracking. In Fig. 6(c), the tracking object has low-resolution. In our algorithm, GoogLeNet with 22 layers was used for feature extraction. It obtained more information, but the multi-level pooling operation caused a decrease in resolution. Therefore, our algorithms were easily missing tracking for low-resolution objects.

IV. Conclusion

In this paper, we proposed a novel visual tracking algorithm by combining SA-Net and a spatial temporal regression model using LSTM. The proposed method extends SA-Net for combining long-time spatial temporal information. It also proved the presentation ability of the LSTM model for long-time sequences and its regression capability for high-level visual features. The experimental results show that the proposed algorithm was robust to shift and scale variation of the object and occlusion.

In future, we will further improve the algorithm by introducing more features and information so that it is robust to more types of videos.

ACKNOWLEDGMENT

We thank Maxine Garcia, PhD, from Liwen Bianji, Edanz Group China (www.liwenbianji.cn/ac) for editing the English text of a draft of this manuscript.

We thank Beijing Municipal Education Commission Science and Technology Innovation Project (KZ201610005012), National Natural Science Foundation of China(61702022), Beijing excellent young talent cultivation project(2017000020124G075), Beijing university of technology "Ri Xin" cultivation project and China Postdoctoral Science Foundation funded project (2017M610027) for partially supporting this work.

REFERENCES

[1] Samira Ebrahimi Kahou, Vincent Michalski, and Roland Memisevic. Ratm: Recurrent attentive tracking model. arXiv preprint arXiv:1510.08660, 2015.

[2] Quan Gan, Qipeng Guo, Zheng Zhang, and Kyunghyun Cho. First step toward model-free, anonymous object tracking with recurrent neural networks. arXiv preprint arXiv:1511.06425, 2015.

[3] Heng Fan and Haibin Ling. Sanet: Structure-aware network for visual tracking. arXiv:1611.06878, 2016

[4] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. arXiv preprint arXiv:1510.07945, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Computer Vision and Pattern Recognition* IEEE, 2014:580-587.

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015.

[7] Thang Ba Dinh, Nam Vo, and Gérard Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. *Computer Vision and Pattern Recognition* IEEE, 2011:1177-1184.

[8] Joao F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. *European Conference on Computer Vision*, pages 702–715. Springer, 2012.

[9] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. *Computer Vision and Pattern Recognition* IEEE, 2010: 49–56.

[10] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, pages 597-606. 2015.

[11] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. *IEEE International Conference on Computer Vision* IEEE Computer Society, 2015:3119-3127.

[12] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng, S. Hicks, and P. Torr. Struck: Structured output tracking with kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence, PP(99):1–1, 2015.

[13] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. *British Machine Vision Conference 2006, Edinburgh, Uk*, September DBLP, 2013:47-56.

[14] Junseok Kwon and Kyoung Mu Lee. Visual tracking decomposition. *Computer Vision and Pattern Recognition*, pages 1269–1276. IEEE, 2010.

[15] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, 37(9):1834–1848, 2015.

[16] Guanghan Ning, Zhi Zhang, Chen Huang, Zhihai He.Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking.arXiv:1607.05781v1 [cs.CV] 19 Jul 2016.

[17] Hochreiter, Sepp, and J. Schmidhuber. Long Short-Term Memory. *Neural Computation* 9(8):1735-1780,1997.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*,2015

[19] Krizhevsky, Alex, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems* Curran Associates Inc. 2012:1097-1105.

[20] Szegedy, Christian, et al. Going deeper with convolutions.*IEEE Conference on Computer Vision and Pattern Recognition* IEEE Computer Society, 2015:1-9.