Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 4427945, 9 pages
https://doi.org/10.1155/2021/4427945

Research Article

Visual Object Tracking with Online Updating for Car Sharing Services

Zhou Zhu,[1] **Haifeng Zhao**,[1,2] **Fang Hui**,[1] **and Yan Zhang**[1]

[1]*School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China*
[2]*Jiangsu HopeRun Software Co., Ltd, Nanjing 210012, China*

Correspondence should be addressed to Zhou Zhu; zhuzhou@jit.edu.cn

Received 26 May 2021; Revised 15 August 2021; Accepted 27 August 2021; Published 21 September 2021

Academic Editor: Pengpeng Jiao

In this paper, we address the problem of online updating of visual object tracker for car sharing services. The key idea is to adjust the updating rate adaptively according to the tracking performance of the current frame. Instead of setting a fixed weight for all the frames in the updating of the object model, we assign the current frame a larger weight if its corresponding tracking result is relatively accurate and unbroken and a smaller weight on the contrary. To implement it, the current estimated bounding box's intersection over union (IOU) is calculated by an IOU predictor which is trained offline on a large number of image pairs and used as a guidance to adjust the updating weights online. Finally, we imbed the proposed model update strategy in a lightweight baseline tracker. Experiment results on both traffic and nontraffic datasets verify that though the error of predicted IOU is inevitable, the proposed method can still improve the accuracy of object tracking compared with the baseline object tracker.

1. Introduction

Car sharing services provide customers access to shared vehicles for short-term use. They can reduce inner-city traffic, trip cost, congestion, and environmental pollution and have developed rapidly in recent years. To achieve better safety and operating efficiency, more and more intelligent vehicle technologies have been utilized in car sharing services [1, 2]. Visual object tracking is a fundamental component of them, by which given an object's initial location in the first frame its locations in subsequent frames can be estimated continually. Moreover, the object's trajectories and velocities can be calculated simultaneously from the tracking results and used for augmented or automatic driving of shared vehicles. Compared with radar tracking, visual tracking technology is cheaper and can perceive richer semantic information about the traffic scene. However, its disadvantage is that there exist several factors such as the real-time variation of illumination, weather condition, and interaction between traffic elements which may usually reduce the object tracking performance in complex traffic scenes. Therefore, there is still huge room for the development of visual object tracking for car sharing services.

A typical visual object tracking method consists of five components, namely, feature extraction, motion model, appearance model, model updating, and integration process [3]. Most studies focus on feature extraction and appearance model. The features used for object tracking include handcraft features such as Color, HOG, LBP, and CN and autolearned convolution features. The main appearance models can be classified to generative and discriminant ones and receive much attention. By contrast, the model updating component is less studied. Most object trackers use the simplest linear weighting for model updating, in which a new appearance model is obtained by weighting the old one and the tracking result of the current frame. The drawback of this method is that the weight factor of the current frame is set unchanged and has no connection with the tracking performance of the current frame during the updating process. In fact, if the tracking result of the current frame is reliable and the object is not occluded, a small weight factor of current tracking result may cause the appearance model not to be updated adequately. On the contrary, if the tracking result of the current frame is inaccurate or the object is occluded, a large weight factor of current tracking

result may cause the appearance model to be updated improperly. Under both situations, some errors may be introduced into the appearance model, and as the updating proceeds, the errors may accumulate and make the appearance model drift away from the object. From the above analysis, we can find that it is necessary to assign a suitable weight factor according to the evaluation of current tracking performance. Nonetheless, how to update the tracking model online based on the analysis of current tracking performance is still an open problem. This study tries to bridge this research gap, and the main contributions are follows:

(1) Introduce an object-specific IOU predictor which trained offline on a large number of image pairs to estimate the performance of current tracking result for object model updating.

(2) Propose a dynamic updating mechanism based on IOU prediction. The updating principle is to assign the current tracking result a larger weight if it is relatively accurate and unbroken and a smaller weight on the contrary.

(3) Integrate the IOU predictor into a lightweight correlation filter tracker and update the tracker online using the proposed updating mechanism.

This paper is organized as follows: Section 2 provides a scan of related works. Section 3 introduces a baseline object tracker and the IOU predictors used in computer vision and proposes our visual object tracker with online updating. Section 4 shows the experimental results and corresponding analysis. Finally, Section 5 presents the conclusions and future research directions.

2. Related Work

As mentioned above, existing visual object tracking methods can be divided into two categories: generative ones and discriminant ones. In the generative methods, the appearance model contains only the object's information and object tracking is achieved by searching for the optimal candidate region that best match the appearance model. Template tracking is the earliest generative tracking method, which takes the original spatial intensity distribution of the object region as the template and tracks the object by template matching. Aiming at the drift problem caused by inadequate updating of templates in tracking, Matthews et al. [4] kept the first template around, used it to align the current template, and finally reduced the possible drift phenomenon to a certain extent. As another classical generative tracking method, the mean shift method [5] takes the object's kernel histogram in the first frame as the appearance model and employs a metric which derived from the Bhattacharyya coefficient as the similarity measure to perform the matching. Throughout the whole tracking process above, the appearance model remained unchanged. To update the appearance model dynamically, Peng et al. [6] employed Kalman filter to filter the kernel histogram using the previous appearance model and current candidate region. The modified method could partly keep up with the changing of object appearance, but the hidden assumption that the object appearance obeyed the Gaussian distribution may not hold in many practical situations. Besides the intensity template and the kernel histogram, low-dimensional linear subspace is also a generative appearance model and first introduced into object tracking by Hager and Belhumeur [7] to handle the object appearance's variation caused by illumination. To update the linear subspace model adaptively, Ross et al. [8] proposed an incremental learning-based tracking method. It collected the object locations in previous frames and employed incremental PCA to update the linear subspace model. Through the updating operation, the linear subspace model could adapt to the variation of object appearance even more.

Different from the generative object tracking methods, the discriminant methods consider not only the objects' information but also the backgrounds' information for tracking. They take object tracking as a binary classification problem, train a classifier to separate the object from the background, and have attracted more attention due to their strength to deal with the objects under complex environments. Most traditional tracking-by-detection methods train their binary classifiers online to update the appearance model, and the updating process always has two steps: (i) the generation and labelling of samples based on the estimated object locations in previous frames and (ii) the online updating of the classifiers [9]. However, the generated samples' labels are often noisy. To increase the classifier' robustness to the poorly labelled samples, several improvements such as robust loss functions [10, 11], semi-supervised learning [12, 13], and multiple instance learning [14, 15] have been proposed.

With the fast development of deep learning, modern visual object trackers such as correlation filtering-based trackers and siamese trackers generally use deep features to build their appearance models, and the corresponding model updating mechanisms have also been studied. MOSSE filter [16] the first correlation filtering-based tracker updates the object model by weighting the current estimated object region and the previous object model linearly, and the linear weighting method has been also used in many other correlation filtering-based trackers [17–20]. Siamese tracker is another kind of modern object tracker, whose basic principle is to learn a similarity metric offline and search online for an optimal candidate region which best matches the object appearance template. SiamFC is the original siamese tracker, in which the object template is initialized in the first frame and then kept fixed during the remainder of the video [21]. Most siamese trackers [22–24] implement the same model updating strategy as the one in MOSSE, and there are two problems in the updating of these trackers. First, the weight factors of current frame are set fixed and cannot change adaptively in the updating process. Second, only the object information is updated, and the updating of the background information is ignored. Aiming at the second limitation, Huang et al. [25] modeled the context between the object and its surroundings by an object-aware weight vector and took the spatial-temporal context into

account in the updating process. Besides the above, there are some learning-based model updating methods. Taking the initial template, the accumulated template, and the template of the current frame as inputs, Zhang et al. [26] utilized a convolutional neural network to learn the optimal template of the next frame in an offline way. Li et al. [27] learned a RNN-based model updater on offline videos by metal-earning. In general, to make the learned mechanism adapt to arbitrary targets, a large number of samples with different kinds of appearance variation are needed for these learning-based model updating methods.

To consider the feedback from tracking results in object model updating, Wang et al. [28] used the response map's peak value and average peak-to-correlation energy (APCE) to measure the confidence of current tracking result. The object model was only updated if these indexes were greater than certain thresholds and remained unchanged if not. Similar to the above method, Sun et al. [29] calculated peak-to-sidelobe ratio (PSR) of response map to evaluate the quality of tracking result in each frame and used it to update the template of a siamese tracker. In addition, Zhu et al. [30] took peak-versus-noise ratio (PNR) as an evaluation index. When the PNR and the max value of response map exceeded certain thresholds simultaneously, one-step stochastic gradient descent with a small learning rate was used to update the object model.

In summary, most modern object trackers update their appearance models without considering whether the estimated object location is accurate or not. Actually, once the object is estimated inaccurately, severely occluded, or totally missing in the current frame, the object model will be updated improperly, and the impact will accumulate continually during the whole tracking. Few research studies used APCE, PSR, or PNR to measure the confidence of current tracking result. These rule-based indicators can be calculated from the response map easily and rapidly, but a lot of information in raw images is thrown away in the calculation. Therefore, they are limited in the evaluation of tracking performance. Different from them, in this paper, we introduce a data-based method to evaluate the performance of tracking results and use it as a guidance to update the object model online. For the reader's reference, Table 1 summarizes some main symbols and their corresponding descriptions used in the following.

Table 1: Symbols summary.

$\mathbf{x}$	Target region
$\mathbf{y}$	Response of correlation filer
$\mathbf{z}$	Search region
$\mathbf{w}$	Correlation filter
$\mathbf{g}$	Response map of correlation filter
φ	Feature extraction network
θ	Parameters of φ
$L(\theta)$	Object function of θ
M, N, D	Size of extracted feature
ε	Accumulated ridge loss
β_t	Updating rate at time t
λ	Regularization coefficient
$F(\cdot)$	Discrete Fourier transform
$\mathbf{B}$	Bounding box
$\mathbf{c}(\cdot,\cdot)$	Modulation vector
$\mathbf{r}(\cdot,\cdot)$	Feature representation of test image
$\varphi I(\cdot)$	IOU predictor module
$\theta\,\mathrm{IOU}(\cdot)$	Predicted IOU of bounding box
T_1, T_2	IOU thresholds
$\mathrm{Lr}_1, \mathrm{Lr}_2, \mathrm{Lr}_3$	Predefined updating rates

3. Object Tracking with Online Updating Guided by IOU

3.1. Base Object Tracker. The features used in traditional discriminant correlation filtering-based object tracking methods are either hand-crafted features like HOG, LBP, and CN or convolutional features trained independently in other visual tasks like image classification and object detection. The separation between feature learning and correlation tracking makes the achieved tracking performance not be optimal. Aiming at this problem, Wang et al. [20] proposed DCFNet which is an end-to-end lightweight network architecture to learn the convolutional features and perform the correlation tracking process simultaneously. Because of its high efficiency and performance, we use it as the base object tracker in this work.

In DCFNet, $\varphi(\mathbf{x}_t, \theta) \in \mathrm{R}^{M\times N\times D}$ denotes the convolutional feature of object region $\mathbf{x}_t$, where φ is the convolutional network used for feature extraction with parameter θ. $\mathbf{y} \in R^{M\times N}$ is the correlation filer's ideal response which is generated by a Gaussian function and peaked at the object region's center. Given the feature extraction network φ, the desired filter $\mathbf{w}_p$ at time p can be obtained by minimizing the accumulated ridge loss ε as follows:

$$\varepsilon = \sum_{t=1}^{p} \beta_t \left(\left\| \sum_{l=1}^{D} \mathbf{w}_p^l * \varphi^l(\mathbf{x}_t, \theta) - y \right\|^2 + \lambda \sum_{l=1}^{D} \left\| \mathbf{w}_p^l \right\|^2 \right), \quad (1)$$

where the parameter $\beta_t \geq 0$ is the updating rate which expresses the impact of object region $\mathbf{x}_t$, D is the channel number of extracted feature, and λ is the regularization coefficient. The closed-form solution of the optimization problem in equation (1) can be formulated in an incremental mode as follows:

$$\widehat{w}_p^l = \frac{\sum_{t=1}^{p} \beta_t \widehat{y}^* \odot \widehat{\varphi}^l(\mathbf{x}_t, \theta)}{\sum_{t=1}^{p} \beta_t \left(\sum_{k=1}^{D} \widehat{\varphi}^k(\mathbf{x}_t, \theta) \odot \left(\widehat{\varphi}^k(\mathbf{x}_t, \theta) \right)^* + \lambda \right)}. \quad (2)$$

Here, hat $\widehat{y}$ denotes the discrete Fourier transform $F(\mathbf{y})$, $*$ represents the complex conjugate of a complex number, and $\odot$ denotes Hadamard product. In the test process, the feature of search region $\mathbf{z}$ is extracted by feature extraction network φ and denoted as $\varphi(\mathbf{z}, \theta)$. Finally, the target's final location is estimated by searching for the maximum value of correlation response map $\mathbf{g}$ as follows:

$$\mathbf{g} = F^{-1} \left(\sum_{l=1}^{D} \widehat{w}^{l*} \odot \widehat{\varphi}^l(\mathbf{z}, \theta) \right). \quad (3)$$

The feature extraction network φ can be trained offline by the stochastic gradient descent method to minimize the

object function $L(\theta)$ on a dataset which consists of a large number of image pairs:

$$L(\theta) = \|\mathbf{g} - \tilde{g}\|^2 + \lambda\|\theta\|^2, \tag{4}$$

where $\tilde{g}$ is the desired response map. Compared with traditional discriminant correlation filter-based tracking methods in which the features and the filters are learned independently, DCFNet can be learned in an end-to-end fashion and get higher accuracy. Furthermore, because the lightweight network architecture is adopted, it can also get a balance between speed and accuracy in tracking and operate in real time. This is the reason that we choose it as the base object tracker.

3.2. IOU Prediction. IOU is defined as the ratio of the intersection area between the candidate object region and the ground truth region to the union area of them. It evaluates the accuracy of the candidate region relative to the ground truth region and is useful in many visual tasks. The prediction of IOU is first implemented by IOU-Net [31] in object detection, in which each IOU-Net is trained for a certain object class independently but not suitable for other sorts of objects. However, the class-specific IOU predictors are of little use for generic visual tracking because the object's class is generally unknown and arbitrary in object tracking. To predict the candidate region's IOU of all sorts of objects in visual tracking, Danelljan et al. [32] proposed a new IOU predictor which could predict an arbitrary object's IOU given only a single reference image by a modulation-based network architecture as shown in Figure 1.

As shown in Figure 1, the IOU predictor network has two branches, and both of them take the specific convolution layers of ResNet-18 as backbone. The reference branch accepts convolution feature $\varphi(\mathbf{x}_0, \theta)$ and the object's bounding box annotation $\mathbf{B}_0$ in the reference image as inputs and outputs a modulation vector $\mathbf{c}(\varphi(\mathbf{x}_0, \theta), \mathbf{B}_0)$. The test branch takes convolution feature $\varphi(\mathbf{x}_t, \theta)$ and the estimated bounding box $\mathbf{B}_t$ in current test image as inputs and outputs a feature representation $\mathbf{r}(\varphi(\mathbf{x}_t, \theta), \mathbf{B}_t)$. Then, the feature representation $\mathbf{r}$ of the estimated object region is modulated by the coefficient vector $\mathbf{c}$ via a channel-wise multiplication. Finally, the modulated representation is fed to the IOU predictor module I which consists of three fully connected layers. The predicted IOU of the estimated bounding box $\mathbf{B}_t$ in current test image is given by

$$\mathrm{IOU}(\mathbf{B}_t) = I(\mathbf{c}(\varphi(\mathbf{x}_0, \theta), \mathbf{B}_0) \cdot \mathbf{r}(\varphi(\mathbf{x}_t, \theta), \mathbf{B}_t)). \tag{5}$$

3.3. Online Updating of the Base Object Tracker with the Guidance of IOU. As many discriminant correlation filtering-based object trackers, DCFNet has an incremental model updating mechanism as shown in equation (2). In the updating process, the parameter β_t which denotes the impact of current tracking result $\mathbf{x}_t$ remains unchanged from the start of tracking as follows.

The assumption behind equation (6) is that the estimated object regions in different frames are of equal importance. Obviously, it does not hold in many cases. For example, if the object is occluded at time t, the estimated object region $\mathbf{x}_t$ may be unreliable and its importance β_t should be reduced to avoid the model drift. On the contrary, if the object's appearance changes little in recent frames, the estimated object region $\mathbf{x}_t$ is more reliable and its importance β_t should increase to update the model more adequately. Therefore, it is necessary to estimate the reliability of the estimated object region $\mathbf{x}_t$ and take it as a guidance to adjust the importance β_t adaptively as follows:

$$\beta_t = \begin{cases} 0, & t = 0, \\ 0.01, & t > 0. \end{cases} \tag{6}$$

In fact, the evaluation of tracking performance has received certain attention and been used for model updating in visual tracking. In most existing methods, the reliability of tracking result is expressed as statistical indexes such as APCE, PSR, PNR, and so on. These statistical indexes are defined manually and calculated based on an intermediate response map. In the evaluation, the original information contained in tracking results such as color, texture, and intensity are ignored. Different from them, we introduce the IOU to measure the reliability of tracking result and use it as a guidance to update the object tracker online. As shown in Figure 2, the original DCFNet is supplemented with an IOU predictor to constitute a new tracker and in which the architectures of two networks are remained unchanged.

Because of the prediction error, it is hard and unnecessary to adjust the parameter β_t very precisely based on the predicted IOU. In our method, β_t in equation (2) is redefined by a linear piecewise function as follows:

$$\beta_t = \begin{cases} \mathrm{Lr}_1, & \text{if } \mathrm{IOU}(\mathbf{B}_t) \le T_1, \\ \mathrm{Lr}_2, & \text{if } T_1 = \mathrm{IOU}(\mathbf{B}_t) \le T_2, \\ \mathrm{Lr}_3, & \text{else}, \end{cases} \tag{7}$$

where T_1 and T_2 are the IOU thresholds and Lr_1, Lr_2, and Lr_3 are the model updating rates.

Compared with the manually defined statistical indexes such as APCE, PSR, and PNR, the predicted IOU between the estimated object region $\mathbf{x}_t$ and the ground truth location is learned from large numbers of samples. It can adapt to different complex traffic scenarios and measure the reliability of current tracking result more exactly and therefore be helpful for the updating of the object appearance model.

4. Experiments

4.1. Experiment Settings. To verify the effectiveness of the proposed object tracker, we first conduct extensive experiments on 2 challenging public datasets including OTB-2013 [33] with 50 sequences and its updated version OTB-2015 [34] with 100 sequences. Without loss of generality, the used datasets contain both traffic and nontraffic scenes. The hardware for the environments includes an Intel E5-2687 3.0GHz CPU, 128GB RAM, and a Nvidia 1080Ti GPU. We implement our object tracker on Pytorch and compare it

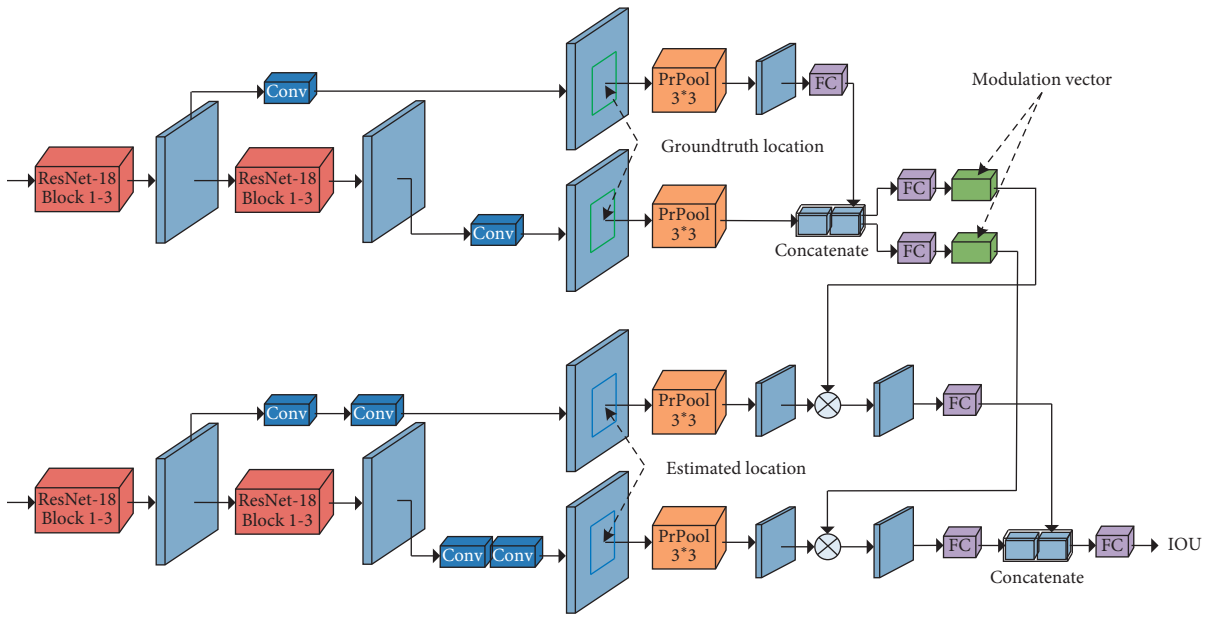

Figure 1: Network architecture of the IOU predictor.

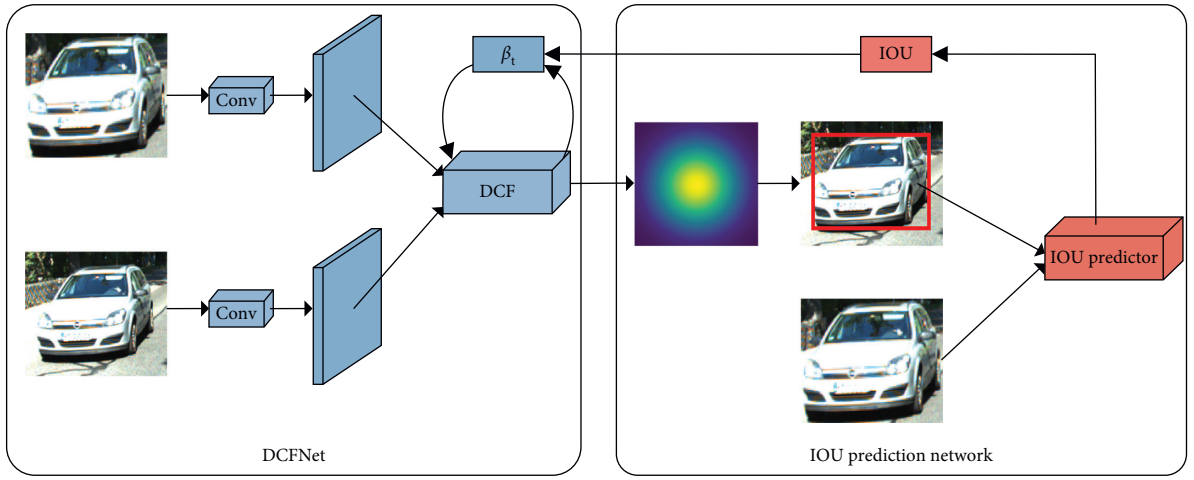

Figure 2: Online updating of DCFNet under the guidance of IOU.

with other 8 modern object trackers such as SRDCF [35], Staple [36], SiamFC [21], CFNet [37], the original DCFNet [20], and its 3 modified versions which update their object models using APCE, PSR, and PNR, respectively.

For a fair comparison with the original DCFNet, the model updating rate Lr_2 in equation (7) is set to 0.01 as same as that in equation (2); meanwhile, Lr_1 and Lr_3 are set to 0.005 and 0.015. The relevant parameters used for model updating in DCFNet-APCE, DCFNet-PSR, and DCFNet-PNR are chosen according to references [28–30], respectively. It is worth mentioning that the proposed tracker is evaluated under 6 different conditions to verify its robustness to hyperparameter selection.

4.2. Experimental Results. The tracking performance of each object tracker is estimated by one-pass evaluation (OPE). Figure 3 shows the success plots of OPE for the propose tracker under condition 1 and other trackers on OTB-2013 and OTB-2015, and the numbers in the legends indicate the average area under curve (AUC) scores of all trackers. A more complete quantitative comparison between our tracker under all conditions and other trackers is shown in Table 2.

In addition to the above experiments on OTB-2013 and OTB-2015, a group of experiments on KITTI [38] which is a vision benchmark of autonomous driving are also conducted subsequently to prove the feasibility of the proposed method

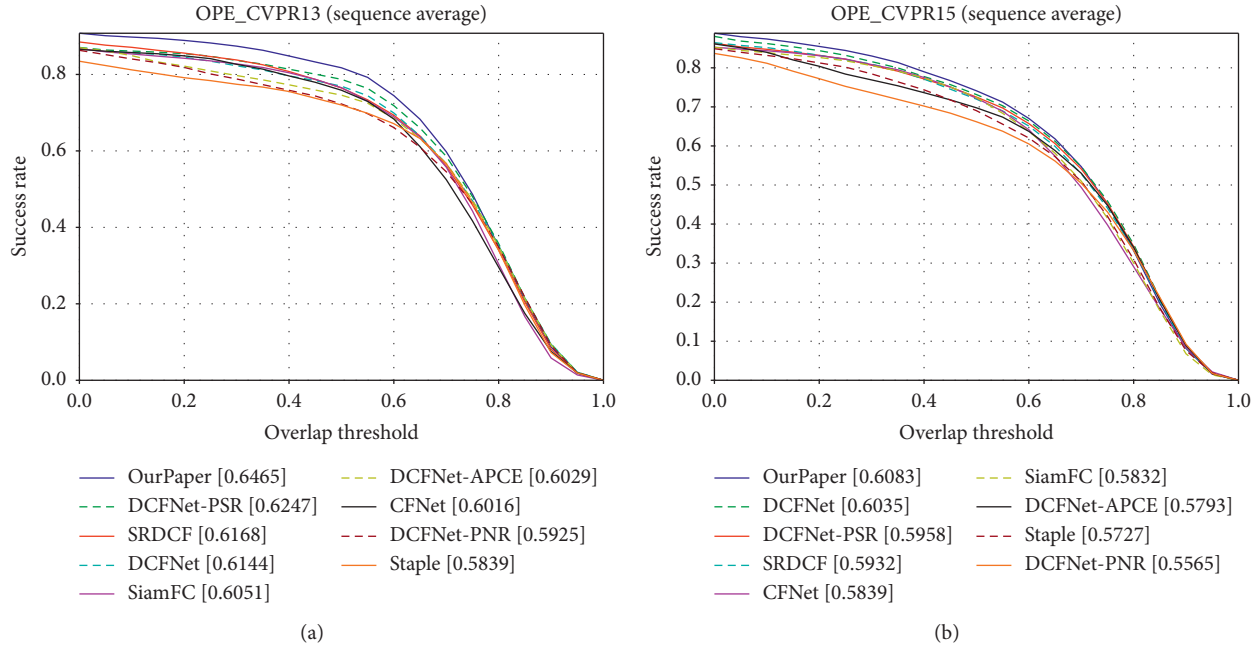

Figure 3: The success plots of OPE on OTB-2013 and OTB-2015 (under condition 1).

Table 2: The average AUC scores of all trackers in the experiments.

	Conditions	OTB-2013	OTB-2015
Our method	Condition1: $T_1 = 0.15, T_2 = 0.80$	0.6465	0.6083
	Condition2: $T_1 = 0.15, T_2 = 0.85$	0.6429	0.6056
	Condition3: $T_1 = 0.15, T_2 = 0.90$	0.6420	0.6069
	Condition4: $T_1 = 0.20, T_2 = 0.80$	0.6465	0.6096
	Condition5: $T_1 = 0.20, T_2 = 0.85$	0.6428	0.6082
	Condition6: $T_1 = 0.20, T_2 = 0.90$	0.6423	0.6097
DCFNet	—	0.6144	0.6035
DCFNet-APCE	The same as reference [28]	0.6029	0.5793
DCFNet-PSR	The same as reference [29]	0.6247	0.5958
DCFNet-PNR	The same as reference [30]	0.5925	0.5565
CFNet	—	0.6016	0.5839
Staple	—	0.5839	0.5727
SRDCF	—	0.6168	0.5932
SiamFC	—	0.6051	0.5832

in traffic scenes. Partial experimental results are shown in Figure 4, and for viewing convenience, the KITTI images are cropped to reduce the field of vision.

4.3. Experimental Analysis. It can be found from Table 2 that the proposed method achieves the highest tracking accuracy under all conditions and therefore has a certain degree of robustness to hyperparameter selection. Taking the results under condition1 as example, the tracking accuracy of our method increases by 5% on OTB-2013 and 1% on OTB-2015 compared with that of the original DCFNet. The improvement verifies that the proposed dynamic update mechanism which is guided by IOU is more adaptable to the variation of object appearance than the fixed update mechanism used in the original DCFNet. Furthermore, our method also exceeds the DCFNets which are modified by APCE, PSR, and PNR, respectively. The reason is that these rule-based evaluation indicators are easy to be affected by the irregularity and noise of the response map. By contrast, as a data-based evaluation indicator which is learned from a mass of videos, the IOU used in our method can evaluate the tracking results more realistically. However, the price is that the tracking velocity has decreased from 80FPS to 30FPS because of the IOU calculation.

In addition, the experimental results shown in Figure 4 demonstrate that our tracking method with online updating can track the traffic participants well and guarantee the operational efficiency and safety of car sharing services.

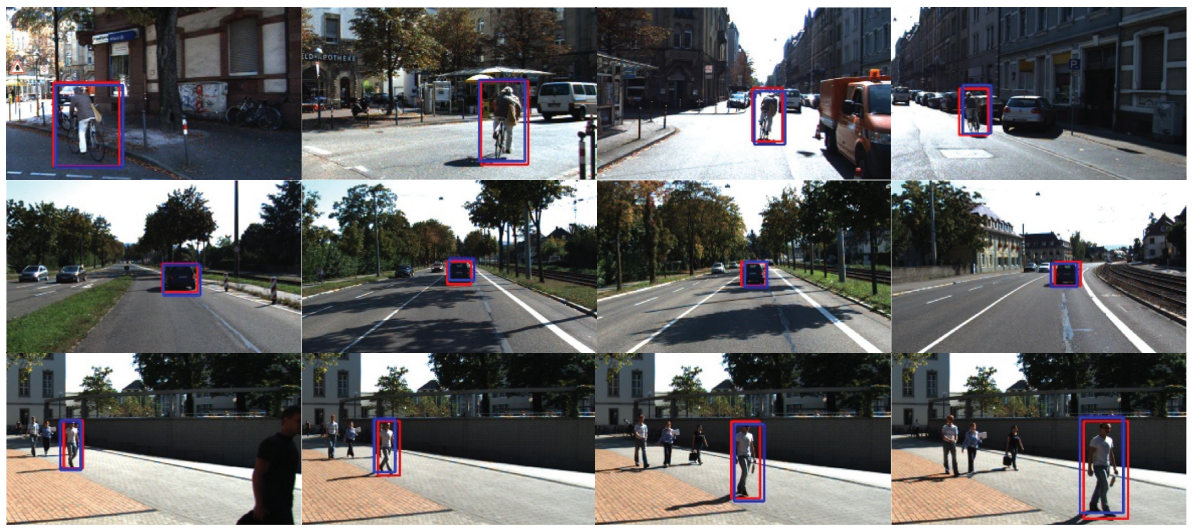

Figure 4: Partial experimental results on KITTI. The red and blue boxes are, respectively, the ground truth and the tracking results of our tracker.

5. Conclusion and Future Work

Visual object trackers can acquire the trajectories of the objects such as pedestrians and vehicles in traffic scene and make the car sharing services more secure and efficient. To promote the tracking performance in complex traffic scenes, it is necessary to update the object model adaptively, and the accurate evaluation of current tracking result is beneficial to the updating of the object appearance model. Instead of using the rule-based indicators such as APCE, PSR, and PNR, we introduced a data-based IOU predictor which is learned offline from a large number of image pairs to evaluate the tracking result. Based on predicted IOU, a dynamic updating mechanism of the object model is proposed. In the updating, if the predicted IOU is high, a larger weight may be assigned to the current tracking result and a smaller weight on the contrary. Finally, we integrate this dynamic updating mechanism into DCFNet tracker. Experiment results showed that compared with the original tracker, the proposed tracker's tracking accuracy increased by 5% on OTB-2013 and 1% on OTB-2015. More than that, out tracker also exceeds the modified DCFNet trackers which update their object models using APCE, PSR, and PNR, respectively. It is verified that as a data-based tracking performance evaluation index, IOU can act as a more reliable guidance than the rule-based evaluation indexes to update the object appearance model online and improve the accuracy of object tracking for car sharing services.

The limitation of our research is that because of the additional calculation produced by IOU prediction, the tracking velocity has decreased from 80FPS to 30FPS. Future research may include backbone network sharing, network structure searching, and model compressing of IOU prediction network to improve the accuracy and speed of the IOU predictor.

Data Availability

The OTB-2013 and OTB-2015 datasets used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the High-Level Talents of Jinling Institute of Technology (No. JIT-B-202013), the International Science and Technology Cooperation Project of Jiangsu Province (No. BZ2020069), the Research Fund for the Doctoral Program of Jinling Institute of Technology (No. JIT-B-201617), and the Major Program of University Natural Science Research of Jiangsu Province (No. 16KJA520003).

References

[1] K. Lu, J. Li, L. Zhou, X. Hu, X. An, and H. He, "Generalized haar filter-based object detection for car sharing services," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1448–1458, 2018.

[2] S. A. Shaheen, M. A. Mallery, and K. J. Kingsley, "Personal vehicle sharing services in north America," *Research in Transportation Business & Management*, vol. 3, pp. 71–81, 2012.

[3] N. Y. Wang, J. P. Shi, D. Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3101–3109, Santiago, Chile, December 2015.

[4] L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, 2004.

[5] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Hilton Head, SC, USA, June 2000.

[6] N. S. Peng, J. Yang, and E. Q. Liu, "Model update mechanism for mean-shift tracking," *Journal of Systems Engineering and Electronics*, vol. 16, no. 1, pp. 52–57, 2005.

[7] G. D. Hager and P. N. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination," in

Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 403–410, Hilton Head, SC, USA, June 1996.

[8] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008.

[9] S. Hare, S. Golodetz, A. Saffari et al., "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.

[10] C. Leistner, A. Saffari, P. M. Roth, and H. Bischof, "On robustness of on-line boosting—a competitive study," in *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops*, pp. 1362–1369, Kyoto, Japan, September 2009.

[11] H. Masnadi-Shirazi, V. Mahadevan, and N. Vasconcelos, "On the design of robust classifiers for computer vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 779–786, San Francisco, CA, USA, June 2010.

[12] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 234–247, Marseille, France, October 2008.

[13] A. Saffari, C. Leistner, M. Godec, and H. Bischof, "Robust multi-view boosting with priors," in *Proceedings of the European Conference on Computer Vision*, pp. 776–789, Crete, Greece, September 2010.

[14] B. Babenko, M. H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2010.

[15] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof, "On-line semi-supervised multiple-instance boosting," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1879–1886, San Francisco, CA, USA, June 2010.

[16] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, San Francisco, CA, USA, June 2010.

[17] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3074–3082, Santiago, Chile, December 2015.

[18] A. Lukezic, T. Vojir, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6309–6318, Honolulu, HI, USA, July 2017.

[19] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1090–1097, Columbus, OH, USA, June 2014.

[20] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu, "DCFNet: discriminant correlation filters network for visual tracking," 2017, https://arxiv.org/abs/1704.04057.

[21] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision*, pp. 850–865, Amsterdam, The Netherlands, October 2016.

[22] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980, Salt Lake City, UT, USA, June 2018.

[23] Q. Wang, Z. Teng, J. Xing, J. Gao, A. Vedaldi, and P. H. S. Torr, "Learning attentions: residual attentional siamese network for high performance online visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4854–4863, Salt Lake City, UT, USA, June 2018.

[24] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 101–117, Munich, Germany, September 2018.

[25] B. Huang, T. Xu, Z. Shen, S. Jiang, B. Zhao, and Z. Bian, "SiamATL: online update of siamese tracking network via attentional transfer learning," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.

[26] L. C. Zhang, A. Gonzalez-Garcia, J. Weijer, M. Danelljan, and F. S. Khan, "Learning the model update for siamese trackers," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4010–4019, Seoul, Republic of Korea, October 2019.

[27] B. Li, W. Xie, W. Zeng, and W. Liu, "Learning to update for object tracking with recurrent meta-learner," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3624–3635, 2019.

[28] M. M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4021–4029, Honolulu, HI, USA, July 2017.

[29] Z. Sun, Q. D. Li, L. Wang, and J. F. Wu, "Deep learning based visual object tracker with template update," *University Politehnica of Bucharest Scientific Bulletin-Series A-Applied Mathema*, vol. 82, no. 2, pp. 65–76, 2020.

[30] Z. Zhu, W. Zou, G. Huang, D. Du, and C. Huang, "High performance visual object tracking with unified convolutional networks," 2019, https://arxiv.org/abs/1908.09445.

[31] B. R. Jiang, R. X. Luo, J. Y. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European Conference on Computer Vision*, pp. 784–799, Munich, Germany, September 2018.

[32] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: accurate tracking by overlap maximization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4660–4669, Long Beach, CA, USA, June 2019.

[33] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: a benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418, Portland, OR, USA, June 2013.

[34] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[35] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4310–4318, Santiago, Chile, December 2015.

[36] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2805–2813, Honolulu, HI, USA, July 2017.

[37] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: complementary learners for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1401–1409, Las Vegas, NV, USA, June 2016.

[38] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, Providence, RI, USA, June 2012.