

Visual Objects Tracking on Road Sequences Using Information about Scene Perspective Transform

Nikolay Nemcev^a, Nickolay Kozyrev^a

^aITMO University, Saint Petersburg, 197101, Russian Federation

Abstract

The paper studies existing approaches and methods used in the task of objects tracking on video, which is one of the most important tasks facing both visual data analysis systems as a whole and road traffic control systems mounted on moving participants of the scene directly (including self-driving vehicles). The proposed approach is used for road scene perspective transform estimation, the search area location, and works in conjunction with a convolutional neural network for objects tracking. The proposed approach helps significantly increase tracking efficiency (on average 10 %, up to 20 % for certain object classes) on a subset of the road scenes videos shot from a moving vehicle and can be used in practice in environment perception modules mounted directly to vehicles.

Keywords

Visual Data Processing, Visual Object Tracking, Convolutional Neural Networks, Perspective Transform, Vanishing Point, RANSAC

1. Introduction

In the automotive industry, computer vision algorithms are used to solve various problems. For example, object and lane detection, velocity and free space estimation, creating functions for understanding the environment, motion planning for autonomous moving devices.

The task of tracking an object between two frames of a video sequence can be represented as a search for the position of the object $R(F_i)$ at some frame F_i , by the known state of the object on the previous frame of the sequence $R(F_{i-1})$, which is given by a rectangular bounding box.

Object tracking technology is widely used in systems for the road environment understanding in the modules of perception and motion planning for unmanned vehicles. Extensive use of technology leads to additional requirements on them. These requirements are related to real-time data processing in the changing weather and illumination conditions, and with the specific nature of the movement of tracked objects. The specific nature of the movement of tracked objects is characterized by high movement speed, frequent overlap, and significant frame-to-frame object's size change caused both by the own movement of the scene objects and the movement of the camera.

In general, real-time object tracking algorithms can be divided according to the method of obtaining and describing the model of the tracked object on two types of algorithms: classical


Proceedings of the 12th Majorov International Conference on Software Engineering and Computer Systems, December 10–11, 2020, Online & Saint Petersburg, Russia

✉ nicknemcev@gmail.com (N. Nemcev); kozyrevkoly@mail.ru (N. Kozyrev)

ORCID 0000-0003-4801-3284 (N. Nemcev); 0000-0003-1952-0041 (N. Kozyrev)



© 2020 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

ones, and based on the principles of machine learning.

Classical algorithms include the basic pattern of the search algorithm [1]. Those algorithms estimate the position of an object at the next frame by searching the area most similar to the used object template (object image at the previous frame) according to the minimum matching error (SAD) criterion or a maximum of the correlation coefficient. Algorithms which are based on principles of contours tracking uses as template information not about the entire pixel field of an object, but its shape and boundaries [2]. It is necessary to take into account approaches based on the methods of the object's key points extracting and their subsequent comparison with key points of the next frame search area [3]. The key points estimation algorithm can be performed by the usage of different approaches described in [4], [5], [6], [7]. The task of tracking objects on a video can be solved by the related task of the object's motion estimation [8].

The advantages of classical algorithms include availability to work without preliminary stage training for tracking module, low computational complexity and high speed of the baseline approaches. The disadvantages of classical algorithms include sensitivity to changes in the illumination of the scene, the problems with object tracking at scenes with a non-static background. It should be noted that the above problems are inherent in the baseline algorithms of this class, and there are algorithms based on the classical principles of computer vision, devoid of these shortcomings. However, such approaches are usually computationally complex and unable to work in real-time [9] and even more for organizing inter-machine exchange through a network [10], [11], [12].

Algorithms based on the principles of machine learning use various neural network architectures [13], [14]. Also, algorithms can use other methods of machine learning, for example, RandomForest [14], [15] These principles of machine learning allow extracting a set of tracked object's features, used later for searching object position at the next frame of the sequence. Some of these approaches search for the position of the object on the next frame by searching for candidate regions in a certain area [16]. Other approaches solve the problem of tracking the object as a one-shot detection task [17]. The need for preliminary preparation of feature extraction modules is a hallmark of algorithms which use machine learning methods [17].

ML-based algorithms (ML - machine learning) are more resistant to changes in the parameters of the scene and more robustly extract features of partially overlapping scene objects, which makes them more applicable in object tracking modules mounted on moving vehicles.

It should also be noted that when solving the task of tracking objects on video often used modifications of the Kalman filter [18]. This modification used both for filtering the trajectory of objects and for predicting the position of the object on the next frame based on the history of its motion [16] are often used in the task of tracking objects on video.

The proposed approach combines a method for assessing the parameters of the perspective transformation of the scene, used to refine the search region at the next frame of the sequence, and a modified convolutional Siamese network for the object position estimation within the given search region [17]. Usage of the proposed method for refining the parameters of the search region is caused by the need to compensate the displacement and resizing of objects moving longitudinally to the camera (in this case, the movement of these objects is generated by both their movement and the displacement of the camera mounted on the vehicle).

2. The general scheme of the proposed approach

Conventionally, the task of tracking an object between two frames can be represented as a search for the state of the object $R(F_i)$ on some frame F_i , based on the known state of the object on the previous frame of the sequence $R(F_{i-1})$, specified by a rectangular bounding box. The proposed approach can be divided into two separate modules - a module for defining parameters of an object search region on the next frame, used to calculate the assumed position and scale of the object, and a modified convolutional neural network that searches for the position of the object in a given region of interest [16]. The general diagram of the approach is given in figure 1.

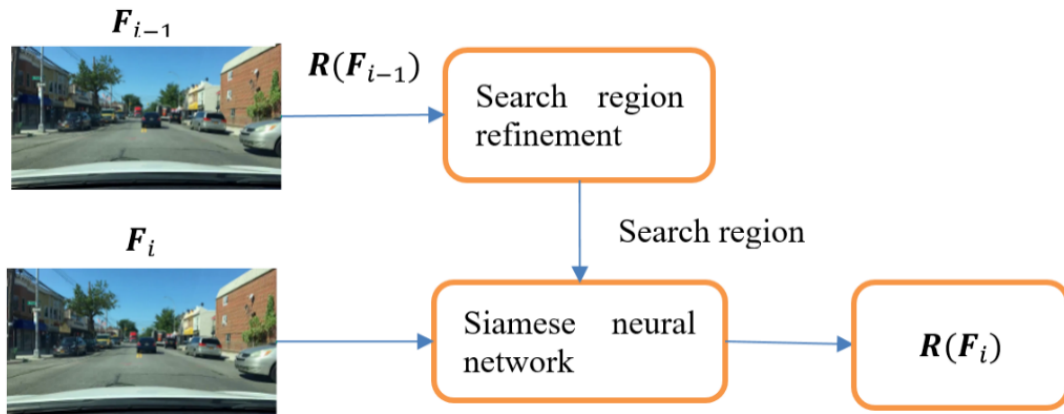


Figure 1: The general scheme of the proposed approach

3. Method for refining search region parameters

For search region parameters estimation (center of an area $i = (x, y)$ and scale S_i) on the frame F_i used by the tracker for the object position estimation is used the procedure based on a method of random samples [18] (RANSAC, Random Sample Consensus) and estimation of parameters of the scene perspective transform by vanishing point search. At first step of the perspective transformation estimation, object boundaries are searched using the Canny edge detector [1], and a set of linear object boundary segments whose length exceeds 3 pixels is searched by the Huff transform. Each segment $E = (pos, dir, len)$ is described by the combination of the position of the center pos , the slope of the segment dir and its length len , while segments whose angle of inclination to the vertical axis of the frame didn't belong to the range from 10 to 70 degrees were removed.

The structure of the RANSAC algorithm can be described by two stages. At first stage a set of hypotheses is selected, in this case, the hypothesis is a model of the vanishing point $M(E_1, E_2)$, selected as the intersection of two random segments E_1 and E_2 , obtained at the previous stage. Finally, the votes for each model are counted and the model with the most votes is the output of the algorithm (target vanishing point).

To count the votes of some hypothesis $M(E_1, E_2)$, we iterate around all available segments E_i and calculate the weight of each voice using the following expression:

$$vote(E_i, M(E_1, E_2)) = \begin{cases} \frac{1-e^{-\alpha \cos^2 \theta}}{1-e^{-\alpha}} \cdot \beta \cdot len(E_i), & \text{if } \theta \leq 5^\circ, \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where θ is the smaller angle between the voting segment and the line connecting the hypothetical vanishing point to the center of the given segment, α is the parameter describing the dependence of the voice weight on the angles similarity level, β is the coefficient describing the influence of the segment length on the weight of the its vote.

The model with the most votes is the approximate position of the vanishing point, describing the parameters of the perspective transformation of the scene. After finding this model, the point refinement procedure is performed according to the approach described in [19].

Knowing the parameters (coordinates of angles) of the bounding box of the object $R(F_{i-1})$ on frame F_{i-1} and the coordinates of the vanishing point $VP = (x, y)$, we can construct a set of estimated parameters of the search region (position and scale) based on the hypothesis about the longitudinal motion of objects [1]. Hypothetical search regions are selected by shifting (taking into account perspective transformation parameters) the bounding box of the object $R(F_{i-1})$ on the frame F_i along the line connecting the center of the given object and the vanishing point, so the coefficient of object scale (ratio of the area of the supposed bounding box to the size of object's box at the frame $R(F_{i-1})$) varies in the range from 0.75 to 1.25 with step 0.1. The illustration is shown in Figure 2.

At next step, the most appropriate $R'(F_i)$ from the set of assumed bounding frames, is selected based on the criterion of the maximum correlation level with the frame area F_{i-1} corresponding to the image of an object $R(F_{i-1})$.

4. Neural network model

After determining the hypothetical search region $R'(F_i)$ the position of the object on the frame F_i is searched using the Siamese neural network for tracking objects. The architecture is almost identical to the network described in [17]. The main difference of this network architecture is that in addition to the object template and search area specified by the previous center of the frame $R(F_{i-1})$, a hypothetical search region is also supplied to the network input. The hypothetical search region is described by the bounding box center $R'(F_i)$ and size change factor S_i . This network solves the problem of object tracking as detecting with template, operates in parallel with both fields of search and describes the received results using the rectangular bounding boxes of $B_{prop}(F_i, R(F_{i-1}), 1)$ and $B_{prop}(F_i, R'(F_i), S_i)$ and the probabilities of detection of $P(F_i, R(F_{i-1}), 1)$ and $P(F_i, R'(F_i), S_i)$ corresponding to the search

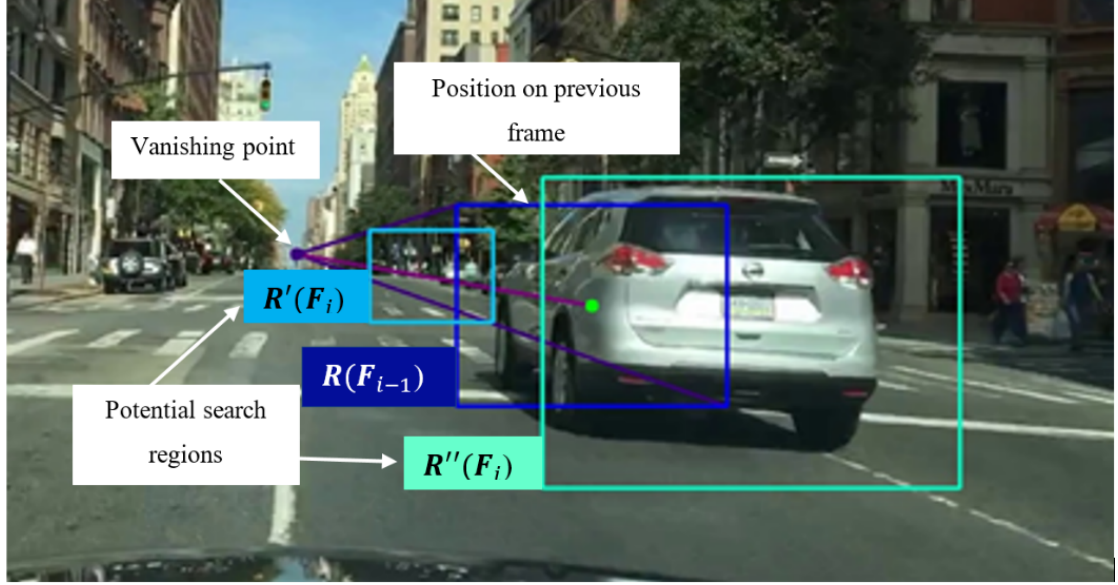


Figure 2: The procedure for determining the parameters of the search region

area described by the previous object position and hypothetical search area respectively. The resulting bounding box $R(F_i)$ is calculated according to the following expression:

$$R(F_i) = \begin{cases} B_{prop}(F_i, R'(F_i), S_i), & \text{if } P(F_i, R'(F_i), S_i) - 0.1|1 - S_i| > P(F_i, R'(F_{i-1}), 1), \\ B_{prop}(F_i, R'(F_{i-1}), 1), & \text{otherwise} \end{cases}, \quad (2)$$

5. Assessment of the effectiveness of the proposed approach

At this stage, a comparative analysis of the proposed approach and the classical implementation of the Siam-RPN tracker [15], which became the basis of the proposed approach, was produced according to the mean overlap criterion (EAO, expected average overlap), calculated in compliance with the procedure described in [19]:

$$\phi = \frac{1}{N_{hi} - N_{lo}} \sum_{N_s=N_{lo}}^{N_{hi}} \phi_{N_s} \quad (3)$$

In equation (3) N_{lo} is the minimum length and N_{hi} is the maximum length of the sequence of frames on which the tracked object is present, and N_s calculated according to the following formula:

$$\phi_{N_s} = \frac{1}{N_s} \sum_{i=1}^{N_s} \phi_i \quad (4)$$

Here (4) N_s is the average overlap for length sequence N_s , and ϕ_i is the coefficient of overlap a predicted position of the object and its true position on the frame i (IOU, intersection over

Table 1

Assessment of the effectiveness of the proposed approach

	Expected Average Overlap, EAO					
	Siam-RPN [17]	Proposed	Diff, %	Siam-RPN [17] + UKF [18]	Proposed + UKF [18]	Diff, %
Car	0.36	0.41	13.89	0.37	0.42	13.51
Pedestrian	0.32	0.38	18.75	0.33	0.36	9.09
Rider	0.35	0.43	22.86	0.35	0.41	17.14
Bus	0.44	0.42	-2.22	0.45	0.45	0.00
Truck	0.44	0.46	4.5	0.43	0.47	9.3
Motorcycle	0.24	0.29	20.83	0.28	0.26	-7.14
Bicycle	0.21	0.24	14.29	0.2	0.25	25
Average	0.34	0.38	11.81	0.34	0.37	8.7

union [1]). A subset of the data set BDD100K [20], consisting of 61 road scenes sequences shot from a moving car in various weather conditions and at different day times was used as a test data set. Tracking was performed for each video object from its first appearance to the end of the video. In this case, the initial state (position of the bounding box) of the object was taken directly from the markup of used dataset. The results of the effectiveness analysis of the proposed approach for different object's classes are given in Table 1.

The obtained results show that using the proposed approach makes it possible to significantly increase the efficiency of tracking objects compared to the classical implementation of Siam-RPN [17]. It should be noted that the proposed approach operates with the same parameters (weights) of the neural network as the classical implementation. At the same time, using the Kalman filter [18] to predict the position of the object on the next frame (and select the corresponding search region) does not give a noticeable increase in tracking quality (Siam-RPN [17] + UKF [18] and Proposed + UKF [18]), this is primarily due to the small length of used video sequences, during which the filter often does not have enough time to formalize the model of the object movement.

6. Conclusion

The approach described in this article for tracking objects on video is based on the method of refining the parameters of the search region and using a modified neural network for tracking objects. The proposed approach of refining parameters of the search region is based on the method of estimating the perspective transformation of the scene by searching for the vanishing point and used to compensate the movement and scaling of objects caused by their longitudinal movement and allows to significantly increase the efficiency of the neural network for tracking objects (average 10%, up to 20% for some object classes) on a subset of video sequences of road scenes taken from a moving camera. Modified network performs object search simultaneously at two search areas using the same object template. It should be noted that the search region refinement module usage slightly increases the computational complexity of the tracking process and its duration. However, the information about perspective transformation may be used by

other unmanned vehicle modules, such as the road marking detection and tracking module. The modified neural network also imposes higher requirements on the computational capabilities of the used graphics accelerator (primarily its memory). However, the relative simplicity of the original Siam-RPN architecture [17] allows the proposed approach for tracking objects to work in real-time on devices mounted directly on moving unmanned vehicles.

References

- [1] E. S. L. Gonzalez R. C., Woods R. E., Digital image processing using MATLAB, Pearson Education India, 2004.
- [2] B. A. Isard M., Contour tracking by stochastic propagation of conditional density, in: European conference on computer vision, Springer, Berlin, Heidelberg, 1996, pp. 343–356.
- [3] D. L. Yang C., Duraiswami R., Efficient mean-shift tracking via a new similarity measure, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE, 2005, pp. 176–183.
- [4] M. J. Lienhart R., An extended set of haar-like features for rapid object detection, in: Proceedings. international conference on image processing, IEEE, 2002.
- [5] S. C. Zhou H., Yuan Y., Object tracking using sift features and mean shift, in: Computer vision and image understanding, 2009, pp. 345–352.
- [6] R. E. et al, Orb: An efficient alternative to sift or surf, in: 2011 International conference on computer vision, IEEE, 2011, pp. 2564–2571.
- [7] B. T., Pedestrian detection and tracking using temporal differencing and hog features, in: Computers Electrical Engineering, 2014, pp. 1072–1079.
- [8] B. J. Y. et al., Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm, in: Intel Corporation, 2001, pp. 1–10.
- [9] F. M. et al., Handcrafted and deep trackers: Recent visual object tracking approaches and trends, in: ACM Computing Surveys (CSUR), 2019, pp. 1–44.
- [10] B. S. V. Bogatyrev A. V., Bogatyrev V. A., Multipath redundant transmission with packet segmentation, in: 2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), 2019.
- [11] P. V. Arustamov S.A., Bogatyrev V.A., Back up data transmission in real-time duplicated computer systems, in: Advances in Intelligent Systems and Computing, 2016, pp. 103–109.
- [12] B. V.A., Exchange of duplicated computing complexes in fault-tolerant systems, in: Automatic Control and Computer Sciences, 2011, p. 268–276.
- [13] S. S. Held D., Thrun S., Learning to track at 100 fps with deep regression networks, in: European Conference on Computer Vision, Springer, Cham, 2016, pp. 749–765.
- [14] N. G. et al., Spatially supervised recurrent convolutional neural networks for visual object tracking, in: 2017 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2017, pp. 1–4.
- [15] L. A. et al., Classification and regression by randomforest, in: R news, 2002, pp. 18–22.
- [16] M. J. Kalal Z., Mikolajczyk K., Tracking-learning-detection, in: IEEE transactions on pattern analysis and machine intelligence, 2011, pp. 1409–1422.
- [17] L. B. et al, High-performance visual tracking with siamese region proposal network, in:

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8971–8980.

- [18] V. D. M. R. Wan E. A., The unscented kalman filter for nonlinear estimation, in: Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, 2000, pp. 153–158.
- [19] I. S. Chaudhury K., DiVerdi S., Auto-rectification of user photos, in: 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 3479–3483.
- [20] Y. F. et al, Bdd100k: A diverse driving video database with scalable annotation tooling, 2018.