Vehicle Video Detection and Tracking Quality Analysis[1,2]

V. D. Kustikova and V. P. Gergel

Lobachevsky State University of Nizhni Novgorod, Institute of Information Technologies, Mathematics and Mechanics, Nizhni Novgorod, Russia

e-mail: itlab.ml@cs.vmk.unn.ru

Abstract—This paper considers the problem of vehicle video detection and tracking. A solution based on the partitioning a video into blocks of equal length and detecting objects in the first and last frames of the block is proposed. Matching of vehicle locations in the first and last frames helps detect pairs of locations of the same object. Reconstruction of vehicle locations in the intermediate frames allows restoring separate parts of motion tracks. Combination of consecutive segments by matching makes it possible to reconstruct a complete track. Analysis of detection quality shows a true positive rate of more than 75% including partially visible vehicles, while the average number of false positives per frame is less than 0.3. The results of tracking of separate vehicles show that objects are tracked to the final frame. For the majority of them the average overlapping percent is not less efficient than the currently used Lucas-Kanade and Tracking-Learning-Detection methods. The average tracking accuracy of all vehicles makes about 70%.

Keywords: computer vision; object detection; tracking; feature extraction; detector; descriptor.

DOI: 10.1134/S1054661816010156

INTRODUCTION

This paper considers the practically important problem of vehicle video-based detection and tracking. The problem occurs in the course of analysis of qualitative and quantitative composition of transport flow. In comparison with [1] this paper proposes new modifications of the solution method for the specified problem allowing to increase the accuracy of detection and tracking.

This paper is organized as follows. First an overview of existing methods is given. Then the problem of vehicle video detection and tracking is formulated. A scheme of the proposed solution method is provided according to [1]. Principal modifications of the method are given. Details of implementation and experimental results are discussed.

RELATED WORKS

A review and classification of the existing methods of video-based object detection problem are provided in [1]. The problem includes object detection in frames and their subsequent tracking. The object tracking methods fall into several categories [2]:

Feature points tracking [3–9]. Objects are represented in consecutive frames by sets of corresponding feature points. Deterministic methods [3] reduce the problem to the minimization of point descriptor compliance function, probabilistic—use an approach based on the concept of state space. Typical examples are methods based on the Kalman [4–6] and particle filters [7–9].

Kernel tracking—tracking the shape of an object or its appearance described by a geometrical primitive (a template of a rectangular or oval shape, a projection of a three-dimensional model). As a rule, methods of this group are applied, if motion is determined by an ordinary shift, turn or affine transformation. In practice, tracking of components is performed using mean shift and its continuous modification (CAM Shift) [10].

Silhouette tracking—tracking a contour or a set of interconnected simple geometrical primitives limiting tracked regions. There are separate methods for matching and tracking segments containing an object [11], and methods of tracking of a contour. Tracking of fragments is carried out by calculation of an optical flow for inner points of a region [12, 13].

VIDEO-BASED VEHICLE DETECTION PROBLEM

The method of video detection deals with a sequence of video frames. Let us assume that the object location is defined by the bounding box placement [1]. Then the problem consists in mapping each frame into a set of objects locations and finding relevant vehicle location in pairs of consecutive frames to reconstruct tracks of vehicles. Thus, a *track* is an

[1] This paper uses the materials of the report submitted at the 9th Open German-Russian Workshop on Pattern Recognition and Image Understanding, held in Koblenz, December 1–5, 2014 (OGRW-9-2014).

[2] The article is published in the original.

Received August 6, 2015

Fig. 1. Test video frames.

ordered sequence of locations of the same object in a corresponding set of video frames. As a vehicle can be overlapped completely by other traffic participants, vehicle location is not necessarily seen in consecutive frames. A formal description of the mathematical problem definition is provided in [1].

VIDEO-BASED VEHICLE DETECTION METHOD

The idea of the proposed method is to divide a video into blocks of images of equal length and then to execute processing of each block. Let us assume that the set of all vehicle locations in the first frame of a block is constructed during the previous iteration. It includes a subset of locations seen in the previous frames, and a subset of objects locations found by the detection algorithm for the first time (could contain false positives). Then it is necessary to detect vehicles in the last frame of the block, match the sets of locations in the first and last frames of the block and reconstruct the vehicle locations in intermediate frames. As a result of reconstruction existing tracks are continued or new ones are created. A more detailed description of the method is provided in [1]. Here we will dwell on the modifications that were made.

The processes of matching sets of vehicle locations constructed in the first and last frames of a block and further reconstruction of locations in the intermediate frames extensively use the operation of matching pairs of images. The operation is intended for building sets of feature points and their SURF descriptors [14] in every image, and for complete matching of the descriptors with the following cutoff of outliers by RANSAC [15]. If in [1] it is assumed that locations are of the same object with a maximal number of inliers, here we suggest using the maximal relative number of inliers. It represents the ratio of the absolute number of inliers to the total number of feature points in the first image of the pair considered. This modification will provide method stability for vehicles of different classes. For example, images of trucks may contain over 200 inliers in the case of full visibility, while images of cars—not more than 100. At the same time, relative numbers are approximately equal, which is also confirmed experimentally.

IMPLEMENTATION

Implementation is based on OpenCV computer vision library [16]. Latent SVM [17] is used as detection algorithm. Unlike [1], the vehicle classifier (CAR class) has been trained using images from PASCAL Visual Object Challenge 2007 dataset [18] and one of the videos (hereinafter—*track_10_5000-7000*) at the ratio of 50 to 50%: 1650 objects, 4250 images not containing objects. The model consists of two components, each of which defines a foreshortening (a view point). The source code and the model are available for downloading [28].

VEHICLE DETECTION QUALITY

For analysis of vehicle video detection quality a few videos (frame rate—25 FPS, resolution—720 × 405 px) were collected:

- *track_10_5000-7000* (2000 frames = 80 s, ~3000 bounding boxes, 58 tracks)—video with only CAR class vehicles, which move in 4 lanes in one direction;
- *track_10_7000-8000* (1000 frames = 40 s, ~1000 bounding boxes, 30 tracks). Contains objects of the CAR and BUS classes;
- *track_10_9000-11000* (2000 frames = 80 s, ~2300 bounding boxes, 48 tracks). Contains objects of the CAR and BUS classes. The principle difference is a large number of trucks.

The marking included all vehicle locations with partially visible objects (up to 2% of visibility).

In addition, synthetic video sequences 2000 frames long containing stationary vehicles were generated. The video sequences were received as a result of repeated copying of one frame (Fig. 1):

- *track_10_5000-7000_1044×2000.* Contains 3 objects: 2 partially visible and 1 is seen less than by 50% (a car);
- *track_10_5000-7000_1192×2000.* Contains 4 objects: 3 fully visible, 1 object is visible more than by 50% (a car entering review region the camera);
- *track_10_5000-7000_656×2000.* Contains 3 objects: 2 completely visible, 1 object is visible by more than 50% (a truck entering the region of interest, the cabin of the driver is fully visible);
- *track_10_9000-11000_206×2000.* Contains 2 fully visible objects (a truck and a car).

This work uses the following measurements for assessment of detection quality: average precision (*AP*) [18];

Table 1. The vehicle detection results (column 1 of each measurement corresponds to the results of [1], column 2 corresponds to the achieved results)

Video	*AP*		*TPR*, %		*FDR*, %		*FPF*	
track_10_5000-7000	0.68	0.80	74.8	84.2	19.9	4.4	0.27	0.06
track_10_5000-7000_1044×2000	0.64	0.64	66.7	66.7	33.3	33.3	1	1
track_10_5000-7000_1192×2000	1	1	100	100	0	0	0	0
track_10_5000-7000_656×2000	1	1	100	100	0	0	0	0
track_10_7000-8000	0.68	0.80	71.3	82.8	32.4	16	0.38	0.17
track_10_9000-11000_206×2000	1	1	100	100	0	0	0	0
track_10_9000-11000	0.62	0.68	69.8	75.9	39.9	29.1	0.44	0.30

the true positive rate (*TPR*); the false detection rate (*FDR*); the average false positives per frame (*FPF*) [19–21]. An object is considered to have been detected correctly, if percentage of overlapping of detected and marked bounding boxes exceeds a threshold (for *TPR*, *FDR*, and *FPF* it was selected as 50%).

The experimental results (Table 1) show that application of the described method modifications allows improving detection quality for all test videos, but for the synthetic ones. This is explained by that in case vehicles are not moving the final result is determined only by the choice of detection algorithm. We note that for *track_10_5000-7000_1044×2000* an object seen less than by 50% is considered to have been found incorrectly due to inaccuracy of bounding box detecting (the intersection region of constructed and marked boxes ranging from 40 to 50%). For other videos the true positive rate increased by 6–11.5%, false positives rate decreased by 10.8–16.4%, and the average false positives per each 10 frames decreased by 1.4 and 2.1 objects respectively.

Let us analyze the consistence of the true positive rate and the false detection rate. We will remove vehicles from the marking that are visible more than the certain threshold and compute appropriate measurements. Threshold changes from 0 (corresponds to the full marking) to 100% (corresponds to the subset of fully visible vehicles). Experimental results show that the true positive rate becomes greater than 90% if the marking contains objects visible more than 20%. When the marking contains only fully visible vehicles this measurement is about 96–98% for all test videos (Fig. 2).

Obviously that at the same time the false detection rate increases (Fig. 3) because of some partially visible vehicles detected correctly by the algorithm will be charged to the set of false positives.

If to analyze a set of vehicles that were not detected by the algorithm we will find about 3/4 objects visible less than 50% (entering/leaving into the frame), and 1/4 objects visible more than 50% (substantially trucks which have invisible cabs).

Let us consider detection quality for the complicated video (Fig. 4): resolution—640 × 480, 3456 frames = 2 min 18 s, 25 FPS, 20292 bounding boxes, 117 tracks.

The proposed method demonstrates the next results: *TPR* = 73.7%, *FDR* = 13.2%, *FPF* = 0.63. True positive rate is worse by 2.2% in comparison with *track_10_9000-11000*. False detection rate is better in average because this video contains more vehicles detected correctly than test. Average false positives per frame twice more, it is explained by the fact of existence of traffic signs and trolleybus lines located above the road (new overlapping situations which are nonstandard for the vehicle model). If to consider only

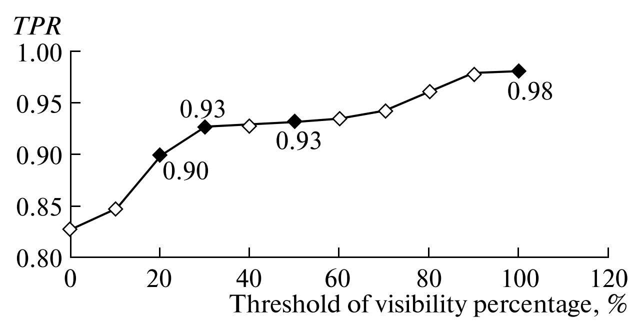

Fig. 2. Variation of true positive rate (*TPR*) for *track_10_7000-8000* while changing the visibility percentage of markup vehicles.

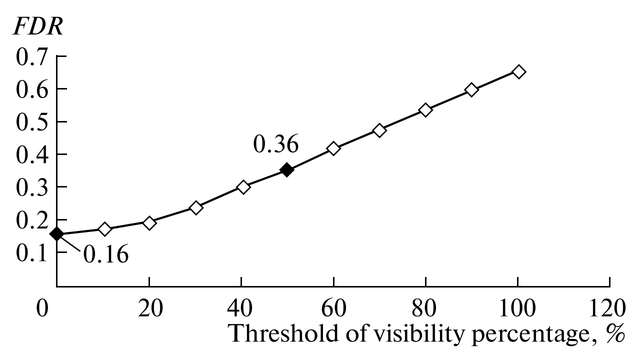

Fig. 3. Variation of false detection rate (*FDR*) for *track_10_7000-8000* while changing the visibility percentage of markup vehicles.

Fig. 4. Complicated traffic situation (another camera's point of view, shadows, high density of traffic flow, traffic signs, trolleybus lines).

fully visible vehicles in the marking the true positive rate achieves 93%.

SINGLE OBJECT TRACKING QUALITY

For analysis of tracking quality tracks of separate objects (Fig. 5) of the test videos were selected:

• *track_10_5000-7000_103-121* contains fully visible moving car;

• *track_10_5000-7000_757-780* contains fully visible moving car of another appearance;

• *track_10_5000-7000_852-889* contains fully visible moving truck;

• *track_10_5000-7000_1712-1727* contains a moving car overlapped less than 50%;

• *track_10_5000-7000_1569×100* contains a stationary car, no other objects in the frame;

• *track_10_5000-7000_1041×100* contains a stationary car is overlapped by another object and is visible approximately by 40%;

• *track_10_9000-11000_1563-1591* contains a car entering the frame, visibility increased from ~50 to 85%, then decreased to ~30% when another object (bus) appears in the foreground.

The tracking was carried out using the following methods:

The proposed method of vehicle video-based detection.

• The Lucas-Kanade algorithm based on optical flow calculation [12, 13]. Pyramidal implementation from OpenCV [22] library was used.

• Median flow algorithm (Predator or Tracking-Learning-Detection, TLD) [23, 24]. An open-source implementation of the algorithm authors [25] was used.

We note that each implementation uses detection results of Latent SVM for CAR objects: the proposed method—in every fifth frame, the other two algorithms—only in the initial frame of the sequence.

To compare the tracking quality of a single object two metrics [26] were used:

• *k*—the ratio of the number of frames, in which the object was tracked until it was lost to the total number of frames containing a track;

• *AvIP*—the average percentage of bounding boxes overlapping in marked and built tracks.

The final values of measurements for the selected set of tracks and methods are shown below (Table 2). It can be easily seen that in all the test sequences the developed method tracks the object without losing it in the intermediate frames (column 2). And in most cases the proposed method is not less effective than the TLD (columns 3 and 7, the shadowed cells). The difference (1–10%) in sequences *track_10_5000-7000_757-780* and *track_10_5000-7000_852-889* (lines 2 and 3) is caused by the fact that the sizes of a bounding box change, when the detection algorithm of the proposed method works at another time. It is noteworthy that small losses of simplicity in the scene lead to a decrease in quality of tracking for the TLD (columns 6 and 7, test sequences *track_10_5000-7000_1041×100* and *track_10_5000-7000_1563-1591*). Partial overlapping of a car by a moving bus (*track_10_5000-7000_1563-1591*) causes losing of the object by the algorithm in the last frames (k = 0.86), and in the initial frames the average percentage of overlapping with marking differs almost twofold in comparison with the proposed method. In practice, such overlapping occurs quite often, which makes application of this tracking algorithm more complicated. It should be noted that the Lucas-Kanade algorithm does not change the size of a bounding box. In every next frame the location is reconstructed based on the mutual location of feature points and the bounding box in the previous frame. As the scale of an object is slightly changed in motion, the

Fig. 5. The initial frames of test sequences for assessment of single vehicle tracking quality.

Table 2. Single vehicle tracking quality

Test track	Proposed method		Lucas-Kanade method		Tracking-learning-detection	
	k	*AvIP*	*k*	*AvIP*	*k*	*AvIP*
track_10_5000-7000_103-121	1.000	0.773	1.000	0.640	1.000	0.731
track_10_5000-7000_757-780	1.000	0.690	1.000	0.662	1.000	0.794
track_10_5000-7000_852-889	1.000	0.688	1.000	0.651	1.000	0.698
track_10_5000-7000_1712-1727	1.000	0.832	1.000	0.709	1.000	0.732
track_10_5000-7000_1569×100	1.000	0.776	1.000	0.776	1.000	0.776
track_10_5000-7000_1041×100	1.000	0.493	1.000	0.493	1.000	0.479
track_10_9000-11000_1563-1591	1.000	0.614	1.000	0.396	0.862	0.328

algorithm loses on this measurement of quality in all tests containing moving vehicles (lines 1, 2, 3, 4, and 7). And in cases with stationary objects (lines 5 and 6) the possible maximum is reached.

ALL OBJECTS TRACKING QUALITY

For assessment of tracking quality for all vehicles in test videos one of the most representative metrics was chosen—*average tracking accuracy* [27]. This indicator reflects the percentage of marked and built tracks overlapping. The results (Table 3) show that the average tracking accuracy does not get lower than 0.7 in all test videos with the exception of the last on. It means that the percentage of overlapping of bounding boxes of the marked and constructed tracks is not less than 70%. It should be noted that it is a high result as the location of a vehicle is determined by a bounding box, which inevitably contains parts of the background or other overlapping objects in its corner segments. The value of the measurement is 2% lower for *track_10_9000-11000*—67.6%. This is due to inaccurate building of bounding boxes. The video contains a large number of trucks, and the detection algorithm, as a rule, finds only the driver's cabin, sometimes with a small part of the body.

CONCLUSION

This paper proposes a modification of the method described in [1]. It was proved that the modification allows improving vehicle detection and tracking quality. The true positive rate grows by 6–11.5% depending on the video compared to [1] and made over 75% provided that the marking contains objects with visibility of up to 2%.

The results of tracking of separate vehicles show that the proposed method is not less effective than the Lucas-Kanade and TLD methods, and for some test objects the overlapping of vehicle locations in marked and built tracks is 10% higher, than for existing methods. The tracking average accuracy of all objects that reflects the percentage of tracks overlapping in a video as a whole makes around 70%.

Table 3. Average tracking accuracy

Video	Average tracking accuracy (%)
track_10_5000-7000	72.1
track_10_5000-7000_1044×2000	72.9
track_10_5000-7000_1192×2000	77.8
track_10_5000-7000_656×2000	74.7
track_10_7000-8000	70.4
track_10_9000-11000_206×2000	74.6
track_10_9000-11000	67.6

ACKNOWLEDGMENTS

The research is supported by the grant of the Ministry of education and science of the Russian Federation (the agreement of August 27, 2013 no. 02.B.49.21.0003), in the laboratory "Information Technologies" of the Institute of Information Technologies, Mathematics and Mechanics of the Lobachevsky State University of Nizhni Novgorod.

REFERENCES

1. V. D. Kustikova, I. B. Meyerov, and N. Yu. Zolotykh, "Vehicle video detection method," Pattern Recogn. Image Anal.: Adv. Math. Theory Appl. **24** (4), 588–592 (2014).
2. A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," ACM Comput. Surveys **38** (4), Article 13 (2006).
3. C. Veenman, M. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points," IEEE Trans. Pattern Anal. Machine Intellig. **23** (1), 54–72 (2001).
4. A. Salarpour, A. Salarpour, M. Fathi, and D. MirHossein, "Vehicle tracking using Kalman filter and features," Int. J. Signal Image Proc. **2** (2) (2011).
5. S. Dan, Zh. Baojun, and T. Linbo, "A tracking algorithm based on SIFT and Kalman filter," in *Proc. 2nd Int. Conf. on Computer Application and System Modeling* (Taiyuan, 2012), pp. 1563–1566.

6. N. Li, "Corner feature based object tracking using Adaptive Kalman Filter," in *Proc. 9th Int. Conf. on Signal Processing (ICSP 2008)* (Leipzig, 2008), pp. 1432–1435.
7. M. Isard and A. Blake, "Condensation—conditional density propagation for visual tracking," Int. J. Comput. Vision **29** (1), 5–28 (1998).
8. F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund, "Particle filters for positioning, navigation and tracking," IEEE Trans. Signal Processing **2** (2), 425–437 (2002).
9. Particle Filter Object Tracking. http://blogs.oregonstate.edu/hess/code/particles
10. D. Exner, E. Bruns, D. Kurz, and A. Grundhofer, "Fast and robust CAMShift tracking," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops* (San Francisco, 2010), pp. 9–16.
11. K. She, G. Bebis, H. Gu, and R. Miller, "Vehicle tracking using on-line fusion of color and shape features," in *Proc. 7th Int. IEEE Conf. on Intelligent Transportation Systems* (Washington, 2004), pp. 731–736.
12. J. Shi and C. Tomasi, "Good features to track," in *Proc. Conf. on Computer Vision and Pattern Recognition* (Seattle, 1994), pp. 593–600.
13. Pyramidal implementation of the Lucas Kanade feature tracker. http://robots.stanford.edu/cs223b04/algo_tracking.pdf
14. H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speed up robust features," Comput. Vision Image Understanding **110** (3), 346–359 (2008).
15. J. Ponce and D. A. Forsyth, *Computer Vision. A Modern Approach* (2004).
16. OpenCV Library. http:// opencv.org
17. P. N. Druzhkov, V. L. Erukhimov, N. Yu. Zolotykh, E. A. Kozinov, V. D. Kustikova, I. B. Meerov, and A. N. Polovinkin, "New object detection features in the OpenCV library," Pattern Recogni. Image Anal.: Adv. Math. Theory Appl. **21** (3), 384–386 (2011).
18. PASCAL Visual Object Challenge 2007. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007
19. Z. W. Kim and J. Malik, "Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking," in *Proc. Int. Conf. on Computer Vision* (Nice, 2003), Vol. 1, pp. 524–531.
20. S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," IEEE Trans. Intellig. Transport. Syst. **11** (2), 267–276 (2010).
21. X. Song and R. Netavia, "A model-based vehicle segmentation method for tracking," in *Proc. Int. Conf. on Computer Vision* (Beijing, 2005), Vol. 2, pp. 1124–1131.
22. OpenCV library functions for calculation of an optical flow using the Lucas-Kanade algorithm. http://docs.opencv.org/modules/video/doc/motion_analysis_and_object_tracking.html#id1
23. Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," EEE Trans. Pattern Anal. Mach. Intellig. **34** (7), 1409–1422 (2012).
24. A description of the Predator algorithm. http://robotdevelop.org/archives/4463
25. Implementation of the Predator algorithm (Tracking-Learning-Detection). https://github.com/zk00006/OpenTLD
26. R. T. Collins, X. Zhou, and S. K. The, "An open source tracking testbed and evaluation web site," in *Proc. IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS)* (Beijing, 2005). http://www.cs.cmu.edu/~rcollins/Papers/opensourceweb.pdf
27. R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol," IEEE Trans. Pattern Anal. Mach. Intellig. **31** (2), 319–336 (2009).
28. Developed vehicle video-based detection system. http://ml.vmk.unn.ru/index.php/en/resources-en

Kustikova Valentina Dmitrievna. Born 1987. Graduated in 2010 the Lobachevsky State University of Nizhni Novgorod. Year of dissertation completion (Candidate's, Doctoral): 2015, Candidate of Engineering Sciences. Assistant at the Lobachevsky State University if Nizhni Novgorod.

Research interests: computer vision, machine learning, parallel computing.

Number of publications (monographs and articles): 8.

Gergel Viktor Pavlovich. Born 1955. Graduated in 1977 the Lobachevsky State University of Nizhni Novgorod.

Year of dissertation completion (Candidate's, Doctoral): 1994, Doctor of Sciences.

Research interests: methods and decision support systems (global and multi-objective optimization problems); mathematical modeling; programming technologies and development of complicated applied software; mathematical models, methods and software of parallel computing for multiprocessor computer systems (clusters).

Number of publications (monographs and articles): above 120.