

DOI 10.31489/2020No2/123-127

UDC 004.896

## THE OBJECT TRACKING ALGORITHM USING DIMENSIONAL BASED DETECTION FOR PUBLIC STREET ENVIRONMENT

Matveev I. G.<sup>1,3</sup>, Karpov K. A.<sup>2,3</sup>, Yurchenko A. V.<sup>1</sup>, Siemens E.<sup>3</sup>

<sup>1</sup>National Research Tomsk Polytechnic University, Tomsk, Russia, matveev.ivan.gr@gmail.com

<sup>2</sup>Siberian State University of Telecommunications and Information Sciences, Novosibirsk, Russia

<sup>3</sup>Anhalt University of Applied Sciences, Koethen, Germany

*The paper proposes an approach to object tracking for public street environments using dimensional based object detection algorithm. Besides the tracking functionality, the proposed algorithm improves the detection accuracy of the dimensional based object detection algorithm. The proposed tracking approach uses detection information obtained from multiple cameras which are structured as a mesh network. Conducted experiments performed in a real-world environment have shown 10 to 40 percent higher detection accuracy that has proved the proposed concept. The tracking algorithm requires negligible computational resources that make the algorithm especially applicable for low-performance Internet of things infrastructure.*

**Keywords:** object tracking, smart city, object detection, street lighting.

### Introduction

Different object tracking algorithms are actively used in applications for automatic traffic monitoring, surveillance, and security. Typical purposes of object tracking algorithms are identification of an object in a video scene (e.g. in consecutive video frames) and prediction of possible object locations in the future. The problem of object tracking is extensively researched in the area of computer vision [1–4], however, these algorithms still have a number of limitations: high computational requirements to a processing unit [1], [5] strong dependency of tracking accuracy on image quality (environment illumination, contrast, resolution, etc.) and varying object representation between frames [6]. There are some implementations of tracking algorithms [7,8] which use background subtraction as a detection technique that can be applied on low-performance computational platforms, however, such an approach does not provide object classification functionality as needed e.g. by Smart Lighting applications [9]. For sure, as a low computational power solution, global positioning system (GPS) tracking algorithms can be used instead of computer vision methods [10], however, it makes devices for such tracking more complicated and expensive from hardware point of view.

The proposed algorithm combines the tracking approaches described in the references above to create a new tracking system that includes a computer vision approach and GPS coordinates extraction that can be executed on low-performance hardware in real-time. The tracking method is developed in a frame of the smart city applications, in particular, diverse smart lighting systems aimed at energy saving and lighting pollution reduction via effective control of the street lamps. The decision to switch the lamp on is made depending on object presence in the street. Each lamp of such a system has its (fixed) GPS coordinates, a low-performance computation unit (e.g. Raspberry Pi based), and a low-resolution camera. The camera data is processed by the real-time detection algorithm (DBOD) [11]. This detection algorithm provides to the system a number of object parameters: object class affiliation probability, coordinates of the object in real world coordinate system, and approximate real-world dimensions (height and width) of the object. All the street lamps inform neighbors about the moving objects [12] through a common wireless mesh network. The tracking algorithm in this system improves the detection accuracy of a particular lamp by taking into account the object detection information received from the lamps which have previously detected the object.

**1. Tracking algorithm description**

The proposed tracking algorithm accumulates the object detection information from multiple nodes. The previously obtained information affects a decision about an object class affiliation on an active node.

The following requirements must be satisfied before applying the proposed tracking method:

- the nodes can communicate through a common network;
- a basic object detector outputs the detection result in a form of an object class probability (confidence score);
- GPS coordinates of each particular node are known;
- the route distance between the neighbors is known to the nodes.

When an object has been detected in the street for the first time, the system registers properties of this object (table 1): a unique ID, GPS coordinates, probabilities of class affiliation, estimated real-world dimensions and its velocity. The velocity parameter is calculated based on GPS coordinates and corresponding timestamps (according to the equation 1). These properties are continuously updated after each detection cycle and broadcasted to the mesh network. These parameters are received by the neighboring nodes and stored as a history set for further calculations.

$$v_{12} \approx \frac{R \sqrt{(\theta_2 - \theta_1)^2 + \cos^2 \left( \frac{\theta_2 - \theta_1}{2} \right) (\varphi_2 - \varphi_1)^2}}{t_2 - t_1}, \tag{1}$$

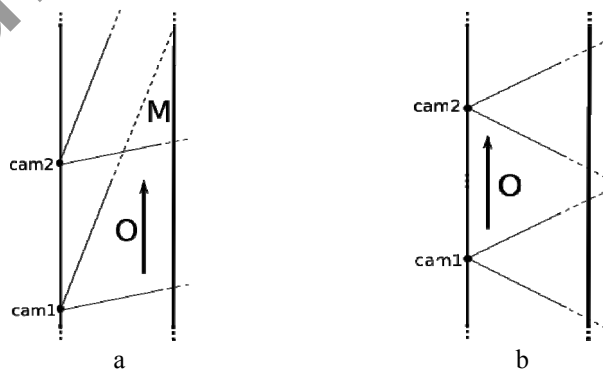
where  $R$  – the approximate Earth radius;  $\theta$ ,  $\varphi$  – latitude and longitude in radians;  $t$  –timestamp.

**Table 1** - Data structure of the detected object

Detected object	Example
ID (10-byte string)	5b759d52b4
GPS array (latitude, longitude)	[(51.1657442, 12.3555677), ..., (53.1453441, 11.1345685)]*
Object width (m)	[0.65, ..., 0.33]*
Object height (m)	[1.77, ..., 1.85]*
Velocity (km/h)	[3.2, ..., 2.5]*
Object class probability	[p_pedestrian=0.7, p_cyclist=0.1, p_vehicle=0.1, noise=0.1]
Timestamp (Unix time)	[1607554893, ..., 1607554899]*

\*the new value is appended to the array after each detection iteration

The tracking algorithm proposes solutions for the two most common scenarios of camera installation in the street: with and without overlap of the camera's angles of view (AOV). Decision about applying the concrete scenario depends on the camera's installation scenario which are shown on Fig. 1.



**Fig.1.** Handover scheme: a) explicit handover; b) implicit handover

In the first scenario (Fig. 1a) the object is observed by both cameras (cam1 and cam2) simultaneously, when the object enters the mutual area M between the cameras. If cam2 finds an object which has the same

coordinates as *cam1*, the object inherits the label assigned by *cam1*. The transition of the object between *cam1* and *cam2* is referred to as explicit handover. The properties of the object are recalculated during the handover that leads to cumulative precision growth of the system. The object detection probability is recalculated in the following way [13]:

$$\hat{p} = \frac{\prod_{i=1}^N \frac{p_i}{1-p_i}}{1 + \prod_{i=1}^N \frac{p_i}{1-p_i}} \quad (2)$$

where  $\hat{p}$  – an updated probability of detection based on probabilities from  $N$  sources;  $p_i$  – a probability of  $i$ -th source;  $N$  – a number of sources.

In the second scenario, there is no mutual area between the lamps (Fig1b), therefore the system will try to predict the coordinates and timeframe of an object's appearance in *cam2* AOV based on information obtained from the camera *cam1*. This scenario is referred to as implicit handover. The implicit handover case is error-prone since the object enters the area of uncertainty and can be interpreted as another or new object at the moment of the next detection. To authenticate the object several verifications should be made.

Firstly, the system checks whether the recent value of the dimensional parameter (width and height) belongs to the inter quartile range (IQR) [14] of the object dimensions history set. To calculate a timeframe of expected object appearance in the area of the next camera the IQR of object velocities is used. Having a distance between areas of observations it is possible to calculate the time interval of the object's expected appearance in the area of the neighboring camera.

## 5. Experimental results

To confirm the effectiveness of the developed tracking algorithm a number of experiments in a real-world street scenario have been conducted. These experiments include movements of *pedestrian*, *cyclist*, and *vehicle* during nighttime. The points of observation have been equipped with Logitech C920 HD cameras which captured images with a resolution of 320x240 pixels. The horizontal AOV of the camera was equal to 40°. The angles of the camera incline varied in the range of 10 – 30°. The height of the camera installation lays in the range of 3.5 – 4.5 m. The tested environment included 6 areas which are shown in Fig. 2.

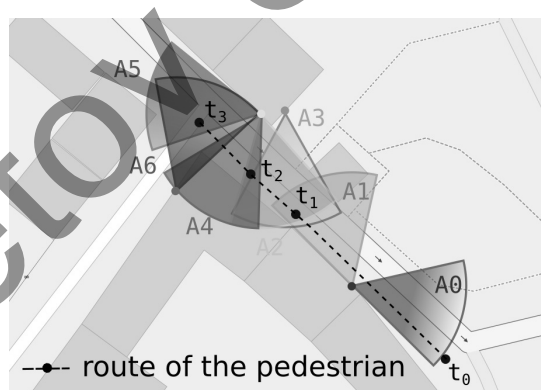


Fig.2. Map of the experiment area

These areas comprise scenarios of explicit and implicit handover shown in Fig.1. So the transition between areas A0 and A1 is an implicit handover case. Transitions between other areas that have mutual regions (A1 – A6) operate explicit handover. Each individual camera detection results are shown in Fig. 3. These images represent a route of the same pedestrian shown in Fig. 2. Image 0 corresponds to the time  $t_0$ , images 1-3 have been taken at times  $t_1$ , image 4 –  $t_2$ , and images 5, 6 – at time  $t_3$ .

The detected object is highlighted by a green rectangle and has a class affiliation with corresponding probabilities. The green-colored values are the un-updated probabilities. The probabilities for all objects classes are plotted in Fig.4a. According to this plot, the detection probability does not exceed 0.85. The lowest detection probability is 0.34 that is below the chosen decision threshold [11] and leads to misclassification.

The red-colored values in Fig. 3 are reinforced by previous observations. The updated probabilities are recalculated using equation 2 and plotted in the in Fig. 4b. As can be seen, the cumulative probability update increases the reliability of the system by 10 to 40 percent since the system becomes more rigid to sudden

errors. For example, misclassifications that are made by node 1 for the pedestrian class and by node 4.5 for the cyclist class have been corrected using observations history. The curves for the pedestrian and cyclist classes approaching detection probability to value of 1.



Fig.3. The pedestrian movement in the experiment area observed from multiple cameras

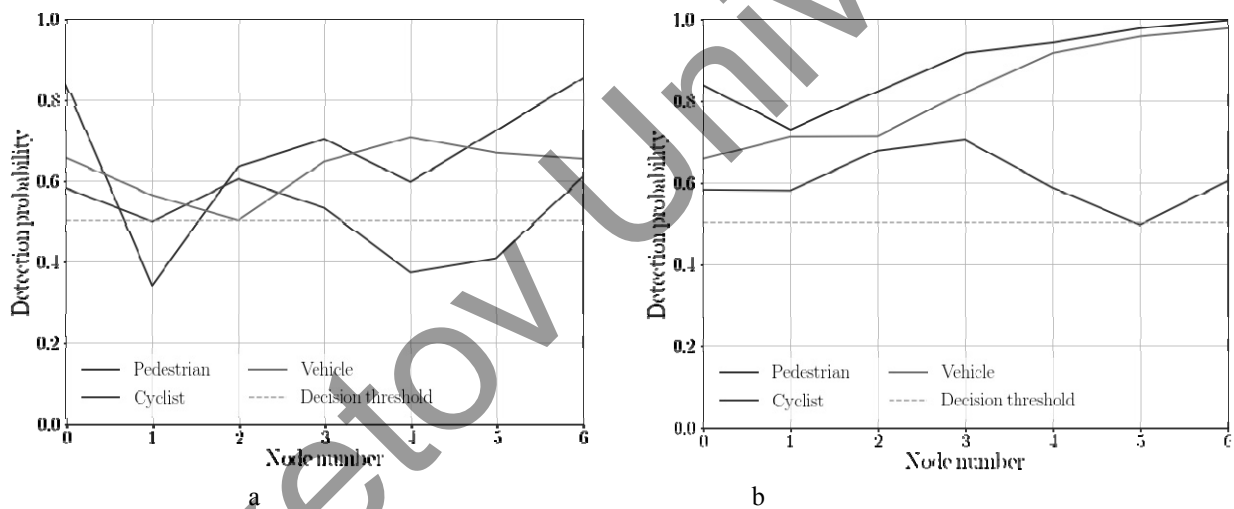


Fig.4. Object class prediction probabilities: a) without cumulative update; b) with cumulative update

## Conclusion

The described tracking algorithm covering explicit and implicit handover scenarios has been implemented and tested on a public street of a small town in Germany. The proposed tracking algorithm has increased the overall reliability of the system by 10 to 40 percent. The tracking method workability has been proved by the results of experiments on the setup in a real-world environment. The experiments on low-performance computational equipment have shown negligible extra load of central processing unit (CPU) load of the tracking algorithm. These CPU requirements make possible applying the proposed tracking approach in real-time on low-performance hardware in contrast to existing analogues. These results can be achieved via the nodes communication in a mesh network. In spite of tracking method specifics, the idea of such tracking can be generalized and applied in other scenarios of object tracking.

There is a frame for further investigations and improvements. The behavior of the tracking system should be refined under more complicated object movement scenarios, e.g. high traffic intensity. The important question is finding suitable method for estimating an object geometrical similarity. In the case of implicit handover, the object location prediction mechanism can be improved for a better object identification in areas of uncertainty.

## REFERENCES

- 1 Yilmaz A., Javed O., and Shah M. Object tracking: A survey. *ACM Comput. Surv.*, 2006, Vol. 38, No. 4, pp. 1 – 45. doi: 10.1145/1177352.1177355.
- 2 Comaniciu D., Ramesh V., and Meer P. Kernel-based object tracking, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2003, Vol. 25, No. 5, pp. 564–577. doi: 10.1109/TPAMI.2003.1195991.
- 3 Wang Q., Zhang L., Bertinetto L., et al. Fast Online Object Tracking and Segmentation: A Unifying Approach. *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, USA, 2019, pp. 1328–1338.
- 4 Xu R., Nikouei S., Chen Y., et al. Real-Time Human Objects Tracking for Smart Surveillance at the Edge. *Proceedings of 2018 IEEE International Conference on Communications (ICC)*, Kansas City, USA, 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422970.
- 5 Voigtlaender P., Krause M., Osep A., et al. MOTs: Multi-Object Tracking and Segmentation. *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, USA, 2019, pp. 7942–7951.
- 6 Philip R., Ram S., Gao X., et al. A comparison of tracking algorithm performance for objects in wide area imagery. *2014 Southwest Symposium on Image Analysis and Interpretation*, San Diego, USA, 2014, pp. 109–112, doi: 10.1109/SSIAI.2014.6806041.
- 7 Ooi H. L., Bilodeau G. A., and Saunier N., Tracking in Urban Traffic Scenes from Background Subtraction and Object Detection. *Image Analysis and Recognition*, Cham, 2019, Vol. 11662, pp. 195–206, doi: 10.1007/978-3-030-27202-9.
- 8 Dziri A., Duranton M., and Chapuis R. Real-time multiple objects tracking on Raspberry-Pi-based smart embedded camera. *J. Electron. Imaging*, Vol. 25, No. 4, pp. 1 – 16, 2016, doi: 10.1117/1.JEI.25.4.041005.
- 9 Zinov S., and Siemens E., The Smart Lighting Concept. *Proceeding of the first Workshop on Problems of Autonomous Power Systems in the Siberian Region*, Koethen, Germany, 2013, pp. 1 – 10.
- 10 Hewawasam H. S., Ibrahim M. Y., Kahandawa G., et al. Comparative Study on Object Tracking Algorithms for mobile robot Navigation in GPS-denied Environment. *Proceedings of 2019 IEEE International Conference on Industrial Technology (ICIT)*, Melbourne, Australia, 2019, pp. 19–26, doi: 10.1109/ICIT.2019.8754960.
- 11 Matveev I., Karpov K., Chmielewski I., et al. Fast Object Detection Using Dimensional Based Features for Public Street Environments. *Smart Cities*, Vol. 3, No. 1, 2020, doi: 10.3390/smartcities3010006.
- 12 Siemens E., *Method for lighting e.g. road, involves switching on the lamp on detection of movement of person, and sending message to neighboring lamps through communication unit*, German Patent No. DE102010049121A1, 2012, 15 p.
- 13 Satopää V. A., Baron J., Foster D. P., et al. Combining multiple probability predictions using a simple logit model. *Int. J. Forecast.*, Vol. 30, No. 2, pp. 344 – 356, 2014, doi: 10.1016/j.ijforecast.2013.09.009.
- 14 Zwillinger D., and Kokoska S. *CRC standard probability and statistics tables and formulae*. Crc Press, 1999, 568 p.