UDC 004.932:621.397](045)
DOI: 10.20535/SRIT.2308-8893.2024.4.03

CROWD NAVIGATION MONITORING DURING EMERGENCIES

O.L. TYMOSHCHUK, M.O. TISHKOV, V.G. BONDARENKO

Abstract. The paper considers the task of crowd navigation monitoring, which might be performed using various sensors and technologies, with surveillance cameras being the most commonly employed. These cameras provide a video stream that typically lacks supplementary information. Extracting additional data from these streams could significantly enhance pedestrian behavior modeling and the automation of the monitoring process. A critical parameter in the analysis of pedestrian movement is their speed. The analytical method and the algorithm of pedestrians' speed estimation based on the surveillance camera video are proposed. The first step of the proposed algorithm is object detection and tracking between frames. The second step is the speed estimation method, which is based on calculating the real-world distances and knowing camera parameters and distances in pixels on the resulting image. Implementation of the algorithm was tested on real videos and showed an error of about 0.04 m/s.

Keywords: computer vision, object tracking, object speed, video surveillance.

INTRODUCTION

Internal navigation monitoring in areas of large crowds during emergencies requires advanced methods of computer vision, object tracking, and estimating flow speed.

Over the past decade, there have been numerous researches aimed at creating intelligent video surveillance systems. Some typical applications are listed as follows:

- public areas such as colleges, campuses, and governmental buildings;
- traffic monitoring;
- crowd management and analysis;
- home security and intrusion detection;
- home care and safety;
- public transport areas such as airports, seaports, and bus/train terminals;
- pedestrian detection and autonomous cars;
- remote military surveillance, border monitoring, perimeter surveillance for power plants, companies, etc [1].

One of the tasks of intelligent surveillance systems is collecting and processing data about objects' behavior. One of the core characteristics of the objects is their moving speed. This paper focuses on estimating pedestrians' speed, however, the same approach could be used for the estimation of any moving objects. Considering pedestrian threads, this data can help develop more effective and safe infrastructures during urban planning and transport engineering. In computer modeling and human behavior simulation, moving speed is a critical parameter for creating realistic agents, that reproduce human behavior in different use cases, like evacuation or interaction with other pedestrians. Thereby, data about the

moving speed of pedestrians could facilitate improvements in various technologies that improve different aspects of our lives.

Problem statement

This paper aims to develop the analytical method and the algorithm for estimating pedestrian speed using a video from a regular surveillance camera.

In order to achieve the goal, the next steps need to be done:

- Analysis of existing methods and approaches of detection, tracking, and speed estimation of moving objects, particularly pedestrians.
- Development of the method for estimating real-world distance, having an image from the camera.
- Implementation of the algorithm based on the developed method that can perform detection, tracking, storing of the pedestrians' tracked data, and estimating their speed.
- Testing of the implementation on the model problems. Algorithm quality evaluation.

Related papers

The computer vision has been investigated by numerous scientists worldwide [1–3]. There are research and implementations for object detection and classification in images and videos [1]. The implementation proposed by authors in this article is based on utilizing various sensors in video surveillance systems. Infrared or thermal cameras aid in detecting people or other objects with contrasting temperatures with the surrounding environment, providing the technical capability for object detection [1].

Radars and lidars also provide technical ability for object detection and accurate speed measurement. However, such sensors are expensive and rarely used. For instance, the number of surveillance cameras in London is about 127.000 [4], but the number of cameras with radar for speed estimation is only about 1000 speed cameras with radar [5]. Therefore, during further research of the methods and their implementations, we will rely on the existing radarless surveillance systems.

An alternative approach utilizing object tracking [2] proposes the extraction and tracking of objects in a video with a stationary background based on motion analysis within frames and representation of images and objects as sets of structural elements. Evaluation of this approach, according to the provided experimental data, reveals that the method does not consider a potential merging and loss of one of the objects, leading to data distortion.

There are numerous studies on object detection and tracking based on background separation with some specific improvements [6–9]. However, this approach has essential drawbacks as explained above.

The methods based on spatial filtering look promising for contour detection [3]. The implemented solution enables image adjustment for object contour detection. One of the advantages of this approach is the ability to simplify data processing having comparably low technical complexity.

Shvandt and Moroz consider a specific case of recognition and tracking of laboratory animals, mice, and fish specifically. The authors provide numerous methods of pattern recognition, including background subtraction and filtering

methods. Also, trainable models are considered, which are used for face recognition, for instance. Such models are more generic, meaning they can work in a wider range of situations, although, they are not ideal as well and can make mistakes or lose tracking objects. It is worth mentioning, that cases reviewed by the authors, — laboratory animal recognition and tracking, are specific and facilitate working conditions for algorithms like background separation since in laboratory conditions background might be homogenous and also there is a possibility to obtain its image without tracking objects [10].

Khan proposes a method of pedestrian detection based on foreground segmentation and calculating the speed on pixels [11]. The author focuses on pedestrian detection on the crosswalks and aims to detect slow speed movements. This approach has similar limitations to the previous author's. Also, the calculation speed in pixels on the image plane is the first step of estimating the real-world speed, which is explained further in this article.

Zhao and Li present a pedestrian tracking method combining a Histogram of Oriented Gradients detection and particle filter and a method for the detection of abnormal crowd activity [12].

In this article, on the stage of people recognition and tracking a well-known model YOLOv7 is used [13]. This model is rapidly developed. It is built on a complex deep-learning model based on a convolutional neural network, which consists of several convolutional and pulling blocks, that subtract characteristics of the image on a different scale level. The determined characteristics are further used to predict the position, size, and class of the object in the image.

Despite the breadth of previous research, pedestrian flow speed measurement has received comparatively little attention, despite its significance. Understanding the flow speed of pedestrians can help to detect unusual events like congestion or emergencies. Additionally, it is vital information for developing pedestrian traffic models, which are instrumental in architectural design and evacuation planning.

Teknomo et al. introduced a data collection framework for analyzing pedestrian flow in [14]. Their system autonomously identifies moving objects, tracks them, and records their positions along with timestamps. By leveraging this collected data on individual movements, the system could discern various characteristics of pedestrian traffic flow, including individual speeds, flow rates, average speeds, and directions. However, it's worth noting that their approach is constrained to video sequences captured from a top-view camera and does not ensure accurate results for videos recorded with lower angles.

Tordeux et al. use an artificial neural network for predicting pedestrian speed in different situations like corridors or bottlenecks [15]. The result of the implementation is prediction, but not the estimation of the physical movements which is not the same and could be used in different use cases and for different goals.

There are studies aimed at estimating pedestrian flow speed using portable devices like phones. Guo et al. propose a method of pedestrian speed estimation based on the human pose detected by sensors of smartphones [16]. Huang et al. use Wi-Fi sniffers to catch Wi-Fi probe requests from mobile devices to estimate pedestrian flow speed and number of people in the crowd [17].

The approach by Lee et al. [18] is closely related to the problem discussed in this research. They proposed a speed estimation algorithm based on calculating conversion factors between the angles on the camera image and real-world motion

vectors. Also, they track the heads of the people for the algorithm. However all people have different heights and the vertical distance from the camera to the tracking object is different for each pedestrian, so they propose a way to predict this distance based on statistical data and crowd density. It adds additional complexity and might be a root for inaccuracy in calculations. Additionally, when calculating conversion factors the authors assume that the rectangle on the camera image projects into a rectangle on the pedestrian movement plane although this is wrong since during central projection, a rectangle is projected into a trapezoid, which changes the provided calculations.

In the considered resources different approaches for moving object detection are proposed. One of them satisfies our needs so it will be used without any changes. However, the main focus of our research is the speed estimation. Method and implementation that cover all our needs and requirements have not been found. The method suggested by Teknomo et al. [14] doesn't work with cameras in their real-life positions, Lee et al. [18] propose a similar approach to ours, but simplifies projection which increases the error of calculations, Tordeux et al. [15] try to predict the speed instead of estimation, and Guo et al. [16] propose an interesting, but completely different approach using the radio signals detected from the pedestrians' phones.

The suggested analytical method allows pedestrians' speed estimation from the video of the camera located above the pedestrians, knowing the camera parameters.

PEDESTRIAN SPEED ESTIMATION METHOD

In this section, the proposed method for pedestrian speed estimation will be explained. The method is based on the distance estimation between two points on a real-world plane based on the points on the image plane. Knowing the distance and time between estimations the moving speed could be calculated.

Experiment model

Fig. 1 shows the pedestrian's movement from point A to point B. To calculate the pedestrian speed between these two positions we need to know the distance and time elapsed. Time could be defined from the video. The real-world distance we aim to estimate.

Fig. 1. Pedestrian movement

Using Cartesian coordinates and knowing the resolution of the image we can calculate the distance in pixels. In the Fig. 2 rectangle $A_0B_0C_0D_0$ is the full view of the camera, $\overrightarrow{BD}$ — the vector of the movement and the diagonal of the rectangle $ABCD$.

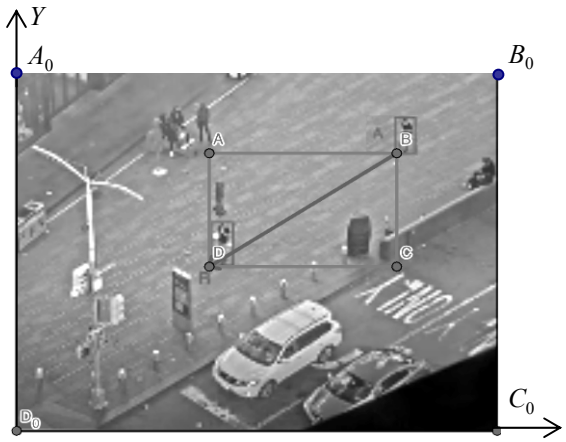

Fig. 2. Pedestrian movement formalization

Fig. 3 shows how the rectangle from Fig. 2 is projected to the real-world plane from the camera point located above the image plane.

In the result we need to estimate size of the vector $\overrightarrow{B_pD_p}$ on the real-world plane which is a projection of the vector of the pedestrian movement $\overrightarrow{BD}$ from the image plane.

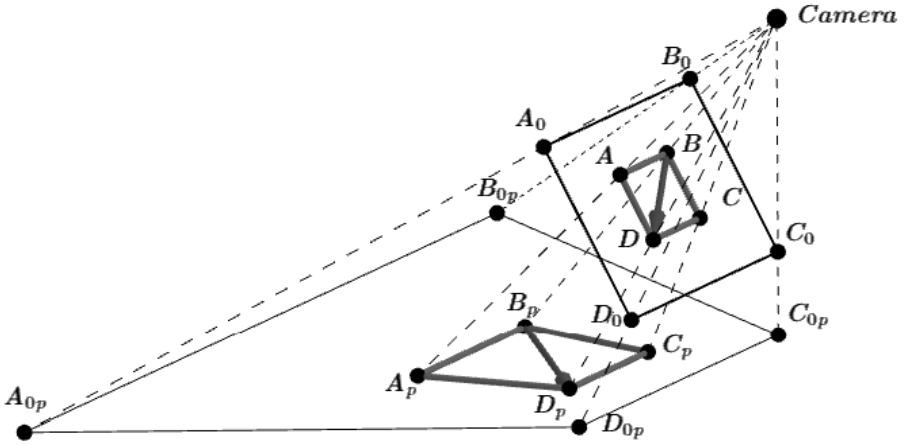

Fig. 3. Projection from image plane to a real-world plane

Mathematical formalization

The first thing that is worth mentioning is that the rectangle on the image plane, located at an angle relative to a real-world plane, is being projected from a point above the image plane to a trapeze on a real-world plane — Fig. 4, as shown in Fig. 1.

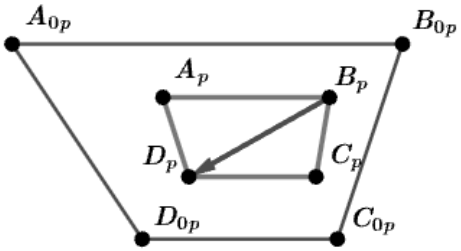

Fig. 4. Camera image and movement vector projection to a real-world plane

We aim to calculate the size of the vector *s* — the diagonal of the trapeze.

For better understanding, we will use simplified figures — Fig. 5.

According to a known formula, denoting the bigger base *AD* as a, we have for both cases of Fig. 5:

$$d = \sqrt{h^2 + (a + h\,ctg\,(\alpha))^2}\ . \qquad (1)$$

In order to find the required diagonal we need to find the height of the trapeze — h, its larger base — a, and the angle near the smaller base from which the diagonal extends — α.

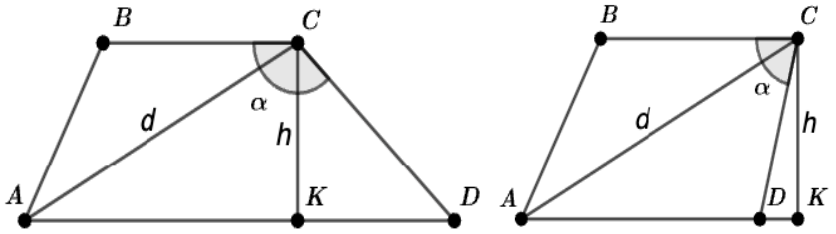

Fig. 5: Trapeze diagonal

The height of the trapeze

Depending on the location of points on the image plane, the calculation of height will vary. It depends on the vertical position of each point relative to the center of the image plane. There are 3 possible cases (Fig. 6).

Here:

C — camera position;

H — height of the camera position;

P_i, P_j — points of the vector on the image plane;

M — middle of the image plane;

Θ — angle of deviation of the camera from the vertical;

Θ_i — angle of deviation of the point P_i from image plane vertical;

Θ_j — angle of deviation of the point P_j from image plane vertical.

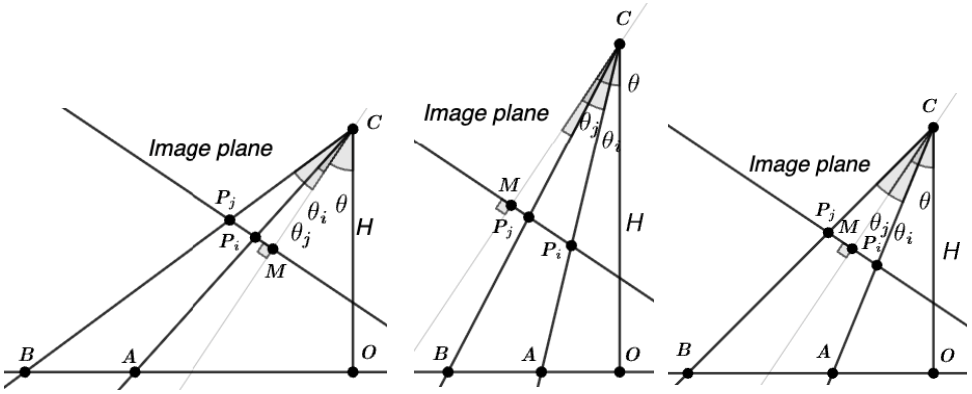

Fig. 6: Trapeze height projection

Segment AB would be the height of the projected trapeze on the real-world plane. Having all described input data the height of the trapeze for cases *a*, *b*, and *c* could be calculated by the next formulas:

$$h = H(\mathrm{tg}(\Theta + \Theta_j) - \mathrm{tg}(\Theta + \Theta_i)) \quad (2)$$

$$h = H(\mathrm{tg}(\Theta - \Theta_j) - \mathrm{tg}(\Theta - \Theta_i)) \quad (3)$$

$$h = H(\mathrm{tg}(\Theta + \Theta_j) - \mathrm{tg}(\Theta - \Theta_i)) \quad (4)$$

$$\Theta_i = \mathrm{arctg}\left(\frac{|P_{iy} - M_y|}{f}\right) = \mathrm{arctg}\left(\frac{\Delta P_i}{f}\right), \quad (5)$$

where ΔP_i — deviation of point P_i from the center of the image plane along the *y*-axis, f — focal length of the sensor.

Knowing the matrix dimensions, we can calculate the metric size of the pixel:

$$k = \frac{Res_{length}}{Matrix_{length}}. \tag{6}$$

We can measure the deviation $\Delta P_{i,j}$ in pixels. By multiplying the vector's size in pixels by the size of the pixel k we can calculate the metric size of the vector $\Delta P_{i,j}$.

The larger base of the trapeze

A larger base of a trapeze will always be the projection of the upper side of the rectangle on the image plane. In Fig. 4 A_1B_1 is the top side of the rectangle on the image plane; $a = AB$ — the larger side of the trapeze; $CO = H$ — the height of the camera position, $B_1O_1 \perp CO$ and $A_1O_1 \perp CO$.

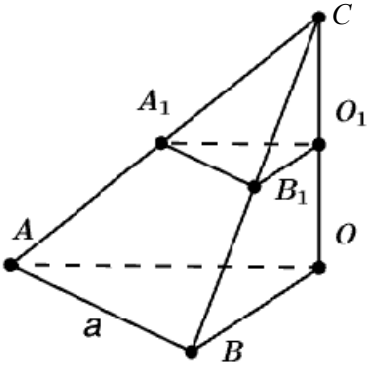

Fig. 7. Larger base projection

Since the triangles ΔAOB and $\Delta A_1O_1B_1$ on the Fig. 7 are similar:

$$a = \frac{CO\,A_1B_1}{CO_1} = \frac{H\,|\,P_{i_x} - P_{j_x}\,|k}{h} \tag{7}$$

Depending on the position of the point on the image plane regarding the center of the plane there are 2 possible cases: when the point is above or below the center of the image — Fig. 8.

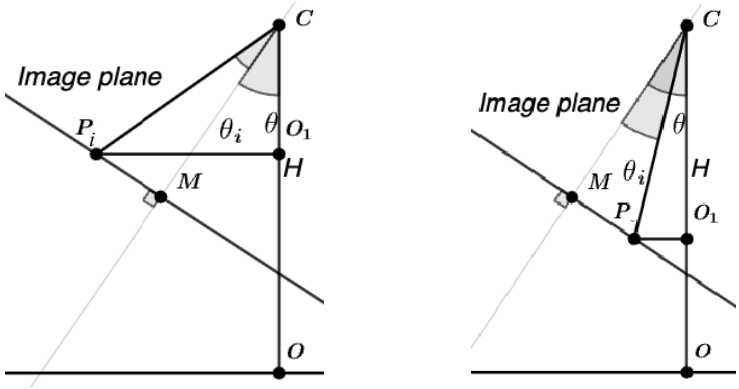

Fig. 8. Point position on the image plane

According to Fig. 8, h could be found using the following equation:

$$h = CP_i \cos(\Theta \pm \Theta_i) = \frac{f \cos\left(\Theta \pm \operatorname{arctg}\left(\frac{\Delta P_i}{f}\right)\right)}{\cos\left(\operatorname{arctg}\left(\frac{\Delta P_i}{f}\right)\right)} \tag{8}$$

Equations (7) and (8) give us the total equation for calculating the larger base of the trapeze:

$$a = \frac{H\,|P_{i_x} - P_{j_x}|\,k\cos\left(\operatorname{arctg}\left(\frac{\Delta P_i}{f}\right)\right)}{f\cos\left(\Theta \pm \operatorname{arctg}\left(\frac{\Delta P_i}{f}\right)\right)}, \tag{9}$$

α — the angle near the small base of the trapeze

Fig. 9 shows the same projection as Fig. 3 but with required additions.

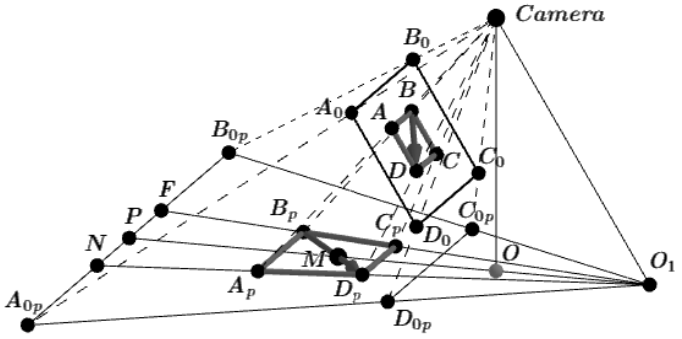

Fig. 9. Extended projection

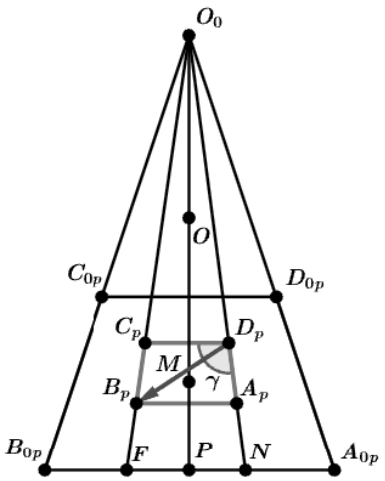

Fig. 10. Projection plane

In this section, we will consider triangle $\Delta A_{0p}B_{0p}O_1$ from Fig. 9 in more detail — Fig. 10.

$\overrightarrow{QS}$ — projection of some vector on the image plane. We aim to find γ — the angle near a small base of the trapeze.

$$\alpha = \pi - \angle O_1NP = \operatorname{arctg}\left(\frac{O_1P}{NP}\right), \tag{10}$$

NP could be calculated using (9), O_1P consists of three parts

$$O_1P = PM + MO + OO_1, \tag{11}$$

PM could be calculated using (2)

$$MO = H\,\mathrm{tg}(\Theta), \tag{12}$$

$$OO_1 = H\,\mathrm{tg}\left(\frac{\pi}{2} - \Theta\right) = H\,\mathrm{ctg}(\Theta). \tag{13}$$

Considering equations above:

$$\alpha = \pi - \operatorname{arctg}\left(\frac{H\,(\mathrm{tg}(\Theta + \Theta_{\max}) + \mathrm{ctg}(\Theta\Theta)\cos(\Theta + \Theta_{\max})}{H|N_x - P_x|\,k\cos(\Theta)}\right), \tag{14}$$

where $\Theta_{\max}$ — maximum vertical deflection angle or half of the vertical viewing angle of the camera.

All required components for (1) could be found using the equations provided above. The required input parameters are:

H — the height of the camera position; f — the focal length of the camera; Θ — the tilt angle of the camera; β — the vertical viewing angle of the camera.

ALGORITHM IMPLEMENTATION

Detection and tracking

The first step of the implementation of pedestrians' speed estimation is pedestrians' identification and tracking. The YOLOv7 object detection model variant by Rizwan Munawar [19] is used in the proposed implementation.

Pedestrians are detected on each frame of the video and data is saved to the MySQL database for further processing. After the video file is processed, the speed estimation step can be executed. It updates the database with speed data, calculated using the provided algorithm, foreach $N's$ frame. As a result, we have data that could be displayed on the video or used for further research, for instance, for estimating the pedestrians' average speed.

The algorithm requires two input parameters, H and Θ, that can be measured manually and the measurement error can significantly impact the performance. To improve the accuracy, a calibration step is implemented that can be executed between detection and speed estimation steps. It requires an array of vectors on the image plane and the distance of their projections on the real-world plane. On the calibration step parameters are tuned to decrease the root-mean-square deviation of calculated results of provided vectors and their real distance.

Final algorithm

At the estimation step, we have all the required data:

- Coordinates of pedestrians at every frame;
- Camera parameters:
  – height of the camera position — h;
  – the tilt angle of the camera — Θ;
  – the focal length of the camera in millimeters;
  – camera resolution;
  – sensor size
  – vertical viewing angle of the camera — β.

The estimation algorithm calculates the speed between every N frames, where N could be set arbitrarily. Knowing the FPS of the video the time between frames could be calculated and the distance traveled by the pedestrian in N frames could be calculated using equation (1), substituting parameters from equations (2–4), (7), (14). Dividing the founded distance by time between N frames gives us speed. Operation is repeated for every N frames till the end of the video for each tracked pedestrian on the video and the results are stored in the database.

Results and model problems testing

Testing the algorithm performance on real videos is a problem since the actual speed of pedestrians is not known. Testing videos were recorded with known distances and times to evaluate the algorithm's performance. The results of the estimation and real manually measured values are presented in Table.

Test video results

Test #	Real average speed	Estimated average speed	Error
1	1.17 m/s	1.25 m/s	6.8%
2	1.1 m/s	1.09 m/s	1%
3	1.19 m/s	1.22 m/s	2.5%
4a	0.99 m/s	0.99 m/s	0%
4b	0.7 m/s	0.77 m/s	11%

In Fig. 11 the estimation process is shown. As we can see, due to the changing size of the detected object, the tracked trajectory for a certain perspective acquires a stepped form. It might increase the error, however, as we can see from the results, this error is compensated through the entire path.

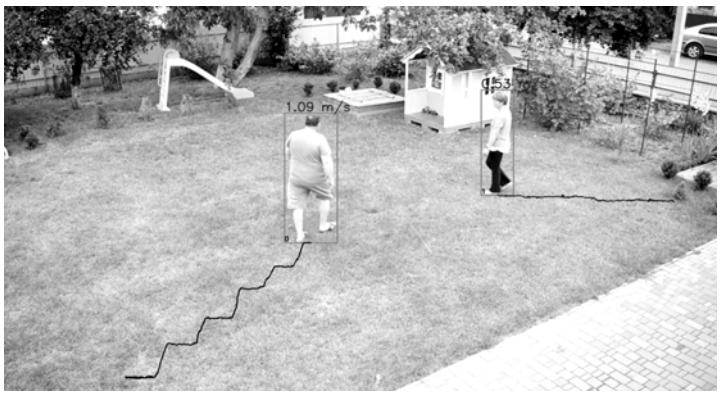

Fig. 11: Estimation process

As mentioned above, the proposed algorithm requires a set of parameters of the processed video to be set. Without them, it will work but the results might differ from real speeds significantly. An example of such a case is shown on a video from the Internet where the parameters of the camera are unknown, which is shown in Fig. 12.

Fig. 12. Speed detection with unknown parameters

Also, for videos with known parameters, we can expect realistic estimations only for pedestrians moving on one plane, which is used for measuring the height

of the camera position. The results for other detected pedestrians will be wrong. The detected pedestrian on the balcony at the left top corner of Fig. 12 can be used as an example of such a case. He is standing above the plain where other people walking, so even having the correct parameters of the camera won't allow us to estimate his speed.

CONCLUSION

In this paper, the method for pedestrian speed estimation based on the video from a surveillance camera, knowing camera parameters is proposed. The proposed method estimates distance on the real-world plane by visible movement vectors on the image plane. In the proposed implementation, the existing model for pedestrian detection and tracking is used. Unlike the previous methods, the proposed method doesn't have an issue with different heights of the tracked objects, since it uses points on the surface for tracking and estimation.

To evaluate the algorithm's accuracy, it was run on testing videos with known pedestrian speed. The resulting relative error is 0–11%, while in absolute terms, it ranges from 0 to 0.08 m/s.

Next steps. Knowing the parameters of the surveillance camera and having access to its video proposed algorithm could be used to retrieve data for pedestrians' behavior analysis. This information could be used during crowd modeling in decision support systems. Also, by improving performance or running on more powerful hardware proposed algorithm may work in a real time. In this case, it could be used for detecting unusual behavior in the crowd.

The current implementation can be found on GitHub [20].

REFERENCES

1. S.W. Ibrahim, "A comprehensive review on intelligent surveillance systems," *CST*, vol. 1, no. 1, pp 7–14, May 2016.
2. O.V. Aharkov, "Object detection and tracking by motion analysis," *Shtuchnyi intelekt*, no. 1-2, pp. 28–35, 2015.
3. O.V. Bahatskyi, "The instrumental complex for research algorithms for object boundaries," *Computers means, networks and systems*, no. 15, pp. 74–78, 2016.
4. P. Bischoff, *Surveillance camera statistics: which cities have the most CCTV cameras?* Accessed on: May 23, 2023. [Online]. Available: https://www.comparitech. com/vpn-privacy/the-worlds-most-surveilled-cities/
5. A. Moody, "Speed Camera Hotspots," *GoShorty*. Accessed on: Jan 17, 2022. [Online]. Available: https://goshorty.co.uk/blog/speed-camera-hotspots/
6. R.D. Sharma, S.K. Gupta, "A Survey on Moving Object Detection and Tracking Based On Background Subtraction," *Oxford Journal of Intelligent Decision and Data Science*, vol. 2018, pp. 55–62, 2018. doi: https://doi.org/10.5899/2018/ojids-00041
7. P. Spagnolo, T. Orazio, M. Leo, and A. Distante, "Moving object segmentation by background subtraction and temporal analysis," *Image and Vision Computing*, vol. 24, no. 5, pp. 411–423, 2006. doi: https://doi.org/10.1016/j.imavis.2006.01.001
8. A. Hardas, B.M. Dattatray, and M. Vibha, "Moving Object Detection using Background Subtraction Shadow Removal and Post Processing," *International Journal of Computer Applications, ICCT 2015*, pp. 1–5, 2015.
9. P. Kalaivani, S. Vimala, "Human action recognition using background subtraction method," *International Research Journal of Engineering and Technology*, vol. 2, pp. 1032–1035, 2015.
10. M.A. Shvandt, V.V. Moroz, "Overview of the detection and tracking methods of the lab animals," *System Research and Information Technologies*, no. 1, pp. 124–148, Apr 2022. doi: https://doi.org/10.20535/SRIT.2308-8893.2022.1.10

11. D. Khan, “Estimating Speeds and Directions of Pedestrians in Real-Time Videos: A solution to Road-Safety Problem” in *AgeingAI 2013 The Challenge of Ageing Society: Technological Roles and Opportunities for Artificial Intelligence, Turin, Italy, 201*3.
12. F. Zhao, J. Li, “Pedestrian Motion Tracking and Crowd Abnormal Behavior Detection Based on Intelligent Video Surveillance,” *Journal of Networks*, no. 10, pp. 2598–2605, 2014. doi: https://doi.org/10.4304/jnw.9.10.2598-2605
13. B.Y. Wang, A. Bochkovskiy, and H.Y.M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” in *CVPR*, New Orleans, LA, USA, 2022, pp. 7464–7475. doi: https://doi.org/10.48550/arXiv.2207.02696
14. K. Teknomo, “Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model,” Ph.D. dissertation, Department of Human Social Information Sciences, Graduate School of Information Sciences, Tohoku University, Japan, 2002.
15. A. Tordeux, M. Chraibi, A. Seyfried, and A. Schadschneider, “Prediction of Pedestrian Speed with Artificial Neural Networks,” in *Proc. Int. Conf. Traffic Granular Flow*, pp. 327–335, 2017. doi: https://doi.org/10.1007/978-3-030-11440-4_36
16. G. Guo et al., “A Pose Awareness Solution for Estimating Pedestrian Walking Speed,” *Remote Sensing*, vol. 11, no. 1, pp. 55–73, 2019. doi: https://doi.org/10.3390/rs11010055
17. B. Huang, G. Mao, Y. Qin, and Y. Wei, “Pedestrian flow estimation through passive WiFi sensing,” *IEEE Trans. on Mobile Computing*, vol. 20, no. 4, pp. 1529–1542, 2021. doi: https://doi.org/10.1109/TMC. 2019.2959610
18. G.G. Lee, K.H. Ka, and W.Y. Kim, “Estimation of pedestrian flow speed in surveillance videos,” in *Conf. Korean Society of Broadcast Engineers Conference, Seoul, Republic of Korea, 2009*, pp. 330–333.
19. R. Munawar, *YOLOv7 object tracking*. 2023. [Online]. Available: https://github.com/RizwanMunawar/yolov7-object-tracking
20. M. Tishkov, *Pedestrian speed estimation*. 2024. [Online]. Available: https://github.com/MaksymTishkov/pedestrian_speed_estimation

Received 26.04.2024

INFORMATION ON THE ARTICLE

Oksana L. Tymoshchuk, ORCID: 0000-0003-1863-3095, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine, e-mail: oxana.tim@gmail.com

Maksym O. Tishkov, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine, e-mail: maksym.tishkov@gmail.com

Victor G. Bondarenko, ORCID: 0000-0003-1663-4799, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine, e-mail: bondarenvg@gmail.com

МОНІТОРИНГ НАВІГАЦІЇ НАТОВПУ ПІД ЧАС НАДЗВИЧАЙНИХ СИТУАЦІЙ / О.Л. Тимощук, М.О. Тішков, В.Г. Бондаренко

Анотація. Розглянуто задачу моніторингу навігації натовпу, що може здійснюватися за допомогою різних сенсорів і технологій, причому найчастіше використовуються камери спостереження. Ці камери забезпечують відеопотік, який зазвичай не містить додаткової інформації. Отримання додаткових даних з цих відеопотоків може значно покращити моделювання поведінки пішоходів та автоматизацію процесу моніторингу. Критичним параметром в аналізі руху пішоходів є їх швидкість. Запропоновано аналітичний метод та алгоритм оцінювання швидкості пішоходів на основі відео з камер спостереження. Першим кроком запропонованого алгоритму є розпізнавання об’єктів та їхній трекінг між феймами відео. Наступний крок — оцінювання швидкості руху об’єктів, що базується на розрахунку реальних відстаней з відомими параметрами камери та відстанями в пікселах на результуючому зображенні. Додатково пропонується алгоритм калібрування для вирівнювання параметрів камери з метою забезпечення найточніших результатів. Реалізацію алгоритму протестовано на реальних відео, похибка — близько 0,04 м/с.

Ключові слова: комп’ютерний зір, трекінг об’єктів, швидкість руху об’єктів, відеоспостереження.