Proceedings of the 34th European Modeling & Simulation Symposium (EMSS), 030
19th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0029 © 2022 The Authors.
doi: 10.46354/i3m.2022.emss.030

Algorithm for computer vision based indoor positioning system for autonomous railway safety system simulation

Jurijs Timofejevs[1,*], Mikhail Gorobetz[1,*] and Andrejs Potapovs[1,*]

[1]Riga Technical University, Azenes street 12, Riga, LV-1048, Latvia

[*] Email address: jurijs.timofejevs@rtu.lv, mihails.gorobecs@rtu.lv, Andrejs.Potapovs@rtu.lv

Abstract

In this paper, an algorithm for detecting position for model trains is presented. It uses a single affordable video camera, a software which utilizes computer vision algorithms to track objects and a simple server to share that data with model embedded devices. This approach has proven that it provides accurate enough (several centimeter precision) positional data to control the models during testing. The system is easy to setup and allows to migrate the software code to use GPS without significant changes.

Keywords: autonomous vehicles, speed estimation, computer vision, safety system, satellite positioning simulation, autonomous vehicle modelling

1. Introduction

The numbers of types of transport units, passenger counts, and amount of transported cargo worldwide is growing every year, that is why the control and management of traffic flow is also becoming more and more important [1], [2]. It is not at all surprising to see a global positioning device installed in majority of commercial vehicles, which are used to address the control problem. Needless to describe the importance of such devices, but research and development of systems whose functionality uses global positioning data poses a challenge (among many others) – there is no GPS signal indoors for testing and its precision is not sufficient for small model appliances. In this paper an indoor positioning system is described, that can help tackle the mentioned problem and add another solution to the pool of already developed ways of setting up an indoor positioning system for testing small scale transport models that require GPS like coordinates.

There is undoubtedly a need for such systems, since the railroad infrastructure will increase significantly which in its turn will require more and more research and development of systems that use GPS and thus more and more demand to the variety of tools available to perform such tests. According to European Commission (EC) Sustainable and Smart Mobility Strategy [3] which indicates that, by 2030 high-speed rail traffic will double across Europe, but by 2050 rail freight traffic will double and a fully operational, multimodal Trans-European Transport Network (TEN-T) for sustainable and smart transport with high-speed connectivity will be developed. But in Europe's Rail Joint Undertaking Master plan [4] EC market, that the railway sector to undergo a significant transformation - increasing its capacity for passenger and goods transport, enabling an increase in the use of rail transport, and reducing further the greenhouse gas emissions of the railway sector itself.

Such global strategies, pose new challenges for raising the level of automation and safety of rail transport management systems to increase the level of automation and security. At present, both the existing railway transport management systems are actively developing (ERTMS [5] in Europe, PTC [6] in USA and for example CTCS [7] in China, although research is being carried out into the development of new automatic control methods and technical solutions using, advanced technologies such as various types of automatic unmanned transport [8], [9], [10] and [11], computer vision [12], artificial intelligence [13], [14], IoT technologies [15] etc., which, in general, makes it possible both to reduce the impact of human factor on the level of safety of railway transport control systems and to increase its performance, accuracy, efficiency and other important parameters.

When solving a problem for some transport system (and railway systems), testing is inevitable and while numerical modelling methods provide an excellent test bed, testing by using models is still unavoidable to bring the developed system closer to the "real world" without spending too many resources for full-scale experiments. Modelling allows to expose the fragile parts of the developed system when it is moved from perfect mathematical simulation environment. Given the reliance of many transport systems on some positional service that can track the position of each agent and feed its coordinates to some service that generates a control signal

In this paper authors provide setup and algorithm of a simple and affordable system, capable of tracking models and detecting their coordinates. The system can track object locations with centimeter accuracy.

2. State of the art

There are quite a few indoor positioning systems developed each of which has its pros and cons.

Bluetooth based systems such as [16], [20] and [21] can be a valid choice, as they are well available, but they lack precision needed, which is around 50cm. Wi-Fi based systems are even less accurate.

A good way of detecting the position would be to use some sort of an onboard device, which tracks markers, say on a ceiling, and returns own position, like [17]. But in this case image processing load must be carried out by a relatively slow embedded device, in experiments performed by the authors, the track width of the model train is 5cm which limits the size of the board and sensors and the amount of computational power available on board of the model. In [18] 8 infrared cameras are used, which can provide great precision, but that comes at a considerable cost.

Acoustic system as described in [19] does provide good precision of under 5cm and is not very costly to set up. Its drawback is lack of mature software and accessible components.

In an attempt to fill the gap, this paper describes a method to set up a small-scale positioning system, which allows to work around the shortcomings of other solutions thus being a more suitable solution for setting up a test bed with indoor positioning system for transport solutions.

3. Mathematical model

The object is detected by using a colorful marker attached at the top of the model vehicle. This is a simple pixel difference task, which any computer vision software can perform. The result of it is a mask, where white pixels represent the marker, and the rest of the image is black. After the marker has been detected and its screen coordinates received, those must be recalculated to get the real-world coordinates as well as adjust for detected object height. However, for some cases raw pixel coordinates are useful as well since pixel grid is a coordinate system by itself.

As thoroughly described in [22] the calculations need a measured point in the center of the image as a base. The calculations itself start from achieving the distance to the calibrated center point located on the image plane:

$$CO = \frac{0.5 * w_{px}}{\tan\left(\frac{\angle KCL}{2}\right)}, \tag{1}$$

Where:

w_{px} – is image width (pixels)

$\angle KCL$ – camera's the field of view (horizontal)

The distance to the center point from the camera mounting post:

$$HX = \frac{h}{\tan(\alpha)} \tag{2}$$

where: H is the ground position of the camera mount, h is the height of the camera above ground, α is the angle between the camera axis and ground plane.

The distance from the camera to the calibrated point (center point) on the ground:

$$CX = \sqrt{h^2 + HX} \tag{3}$$

The vertical component Δ of the angle between the line from viewpoint to CX and the line, connecting the viewpoint and the point of interest on the ground, where d_v is the vertical component of the vehicle:

$$\Delta = \operatorname{atan}\left(\frac{d_v}{CO}\right) \tag{4}$$

β is the angle of the line that connects point of view to the point of interest and the ground:

$$\beta = \begin{cases} 180° - \alpha - \Delta, & dv > 0 \\ 180° + (180° - \alpha) + \Delta, & dv \le 0 \end{cases} \tag{5}$$

After that we can calculate the X and Y components the distance from the center point to the vehicle (6) and (7):

$$D_v = \mathrm{D} * \frac{\sin(\angle\Delta)}{\sin(\angle\beta)} \tag{6}$$

$$D_h = D * \sin(\angle OCZ_h) \tag{7}$$

Once the components of current location of the vehicle are known at a current frame, then the process needs to be repeated for the second frame and the distance travelled by the vehicle would be:

$$S = \sqrt{\left(D_h^i - D_h^{i-1}\right)^2 + \left(D_v^i - D_v^{i-1}\right)^2} \tag{8}$$

4. Developed Simulation and Control Algorithms

The camera was positioned on a tripod and targeted at the area of interest. Each of the tracked model vehicles should have a color marker and an on-board computing device, capable of controlling its motors, an MCU board with network module is sufficient.

Alternatively, the camera can be positioned directly above the testing area. This way it is easy to use the pixel coordinates directly without any calculation since the distortions are negligible for the selected scale.

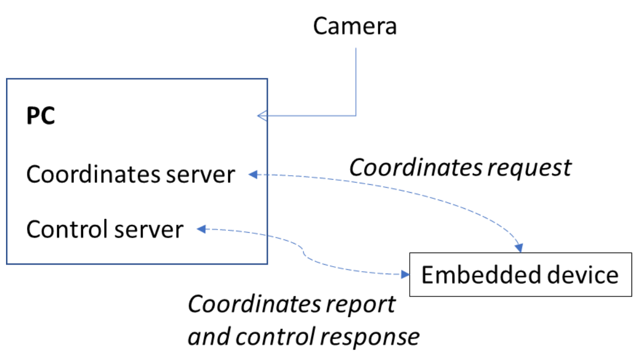

Figure 1. Scheme of the test system

When launched, the server starts the video recognition process, receiving video stream from a connected camera and tracks color markers on the model vehicles, recording their positions. Train controllers periodically create a http request and get their coordinates. These coordinates can be used in any way necessary: either processing them on board or creating a request to some other controlling software. In the example described in the experiment section, the embedded software creates another http request to another server that acts as a station controller.

The station controller accumulates all the coordinates from all the trains and constantly updates their state. On each request the server processes the acquired data and decides what to do with the model vehicle which issued the request: whether it should brake, change its course, or slow down for instance.

The object tracking uses computer vision at its core. Each input frame color of the video is converted to HSV format.

Then, a mask for a marker color is created. A contour is drawn, and a center point of the tracked contour is used as an on-screen coordinate. Software is configured to draw a minimum area contour, this way the framework rotates the bounding rectangle, and the detected center point of the marker is as close as possible to the actual center point of the marker. Alternatively, a circular marker can be used.

The coordinate update and request processes are not synchronized in any way, so the request for coordinates returns whatever there is at that moment. The response time observed is short enough, around 50ms which is enough to not get too much lag unless the models are fast. For example: in the tested case it took 7 seconds for a model to complete the entire circle at top speed and the whole system was able to keep up.

Pixel coordinates in some cases can be used directly or they can be sent into the calculation function to get the real-world X and Y distance components counted from the point which was selected as a center. Knowing the two coordinates, the height of the detected marker above the base plane, and the angle β it is possible to correct the calculated coordinate and get its projection on the base plane.

Let's consider the drawing at **Figure 2**. The distance calculator function returns the coordinates of the point Y1 that the camera tracks. Unless the tracked point is at the ground level, there will be a position error. The real position will be at point Xc. Two projections of the shown drawing: Figure 3 and **Figure 4** need to be considered, one for each component of the corrected coordinate.

The two values Y_vX_c and BX_c are the searched coordinates of the projection of the tracked point onto the ground plane, where the calibrated center point is.

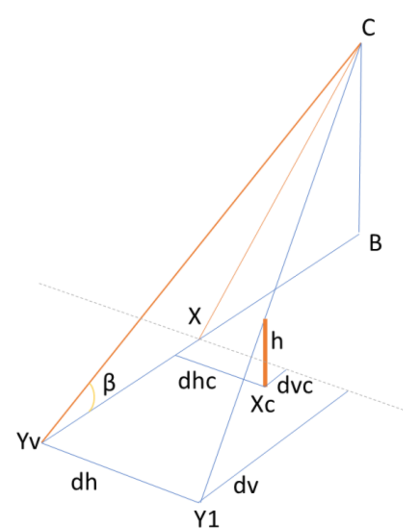

Figure 2. 3D illustration for coordinate correction.

$$YM = \frac{Y_vN}{\cos\beta} \tag{1}$$

$$MN = \sqrt{YM^2 - Y_vN^2} \tag{2}$$

$$YE = \frac{d_v * h}{MN} \tag{3}$$

$$Y_v X_c = d_v - YE \tag{4}$$

$$BX_c = d_h - \frac{d_h * h}{CB} \tag{5}$$

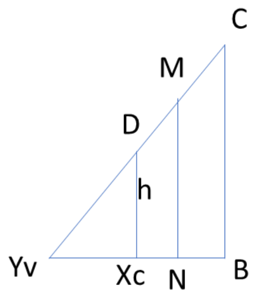

Figure 3. Projection for correcting the *dv* component.

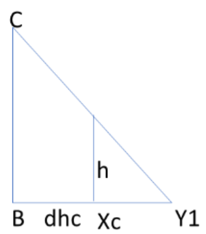

Figure 4. Projection for correcting the *dv* component.

5. Simulation and experiments

To set up the system the materials were used as mentioned in Table 1.

Table 1. Hardware.

Item	Quantity	Notes
HD web camera	1	60FPS capable. HD resolution. 90° FOV.
PC	1	System of AMD Ryzen 7 level used, but AMD Ryzen 5 or Intel i5 would be sufficient.
ESP32	For each model	

The distance detection method was tested by setting up the camera and measuring coordinates to known points.

Figure 6. Model trains during the test. Notebook, acting as a server is visible next to the monitor, not the color markers on train roofs. The size of the table is 1m by 1,60m

Overall, 103 measurements were done which yielded the average difference between the calculated and actual point was under 1cm as seen on **Figure 7** and **Figure 8**. The size of the trapezoid enclosing the test area is 560mm (width top), 360mm (width bottom) and 360mm height. **Figure 9** shows the error value separately.

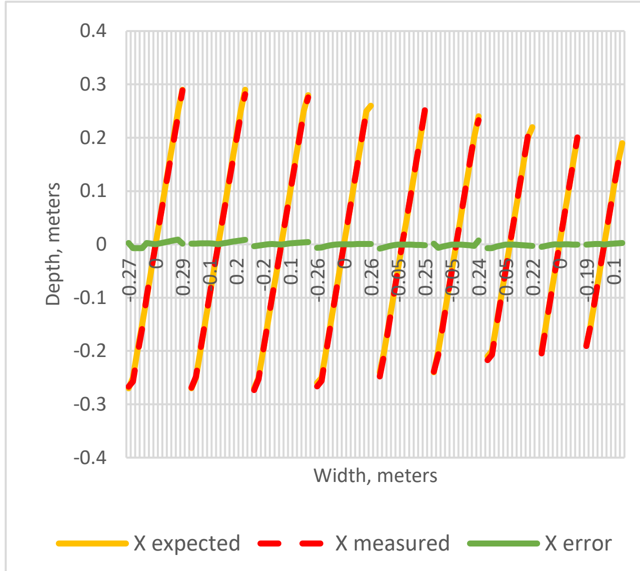

Figure 7. X axis measurements actual vs expected.

The model trains are controlled by an ESP32 board. Each board polls the specified IP address once in 250ms, same for the switches. The positioning server tracks the train color markers re-calculates pixel coordinates to real coordinates and returns them when requested. The train, after receiving the coordinates requests the speed command from the control server. The control server processes all the coordinates received from the trains and generates a speed value to be sent back to each of them and the trains change their speed accordingly or stop.

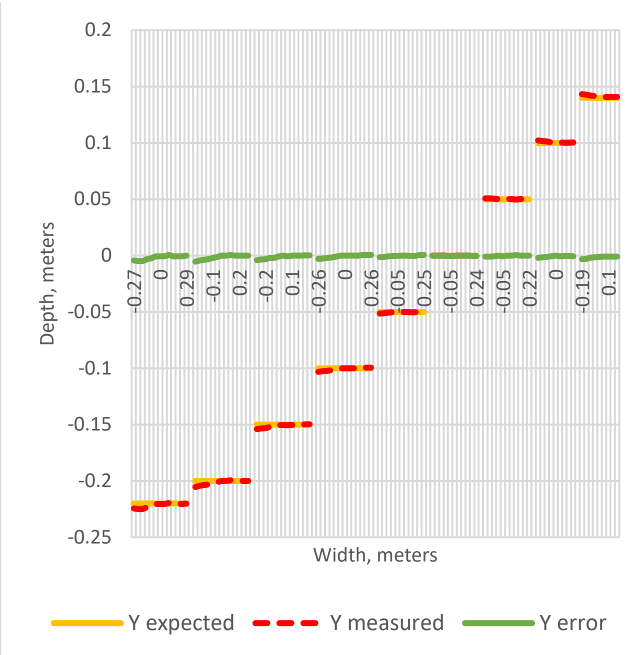

Figure 8. Y axis measurements actual vs expected.

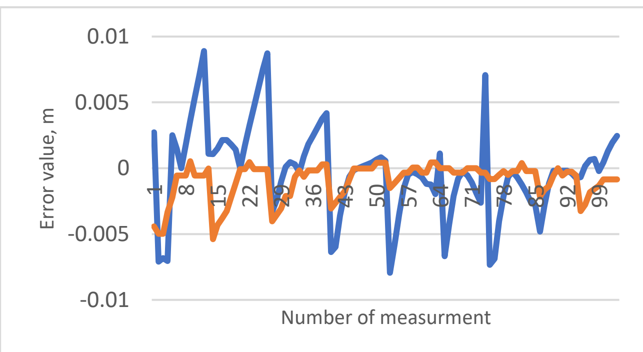

Figure 9. Coordinate error value X(blue) and Y(Orange).

6. Results and Discussion

The developed system has proven to be a simple, cheap, and suitable way of mocking the positional data. Overall cost of the system, excluding the model trains, is approximately $200 and importantly, does not use components that are difficult to obtain, but rather widely available ones. Same is applicable to software. There is a wide variety of free HTTP server solutions available, that allow to set up one on virtually any computing device, be it a laptop or a Raspberry board.

It took the authors two days to set up a positioning server from scratch and several hours to set up control server. After that it became possible to work with the most complex part – the control logic, which is beyond the scope of this paper.

The railroad model used, could be operated by using the positional data provided by the system and allowed to test variants of setting up the control. The locations of model trains were detected with accuracy sufficient to pinpoint their position, process the state of the railway and generate a valid control command.

The system is easy to setup, uses basic components, which are accessible thus allowing to quickly set up a test bed to model the required case. In some cases, raw pixel coordinates can be used, but, if necessary, they can be re-calculated to a different coordinate system which will provide more flexibility for processing.

7. Conclusions

If a bigger area, that exceeds the field of view of the camera needs to be covered, then using cameras can become challenging, however there seems to be a way to use 2 or more cameras to cover larger area, this is something to consider for the future research. Also, the proposed method is impractical for cases when 3D positioning is needed.

Some errors are introduced when calculations are performed to convert pixel coordinates into real coordinates, but as mentioned the overall error stays below 1cm. There is room for improvement in this area which could be addressed by researching methods about how to better set up the camera to minimize measurement errors and to search for a better mathematical model to account for distortions, as well as to check the number of errors introduced by the computer vision part and the calculations part. Overall, a single camera is enough to track position of model vehicles on a 2D plane and is useful to perform various tests indoors with scale models.

Funding

Publication is created with support of European Regional Development Fund project "Development of new sensors and research of their applications for traffic and pedestrian flow data to improve smart city lighting and other urban planning solutions", agreement no. 1.1.1.1/20/A/028."

This work has been supported by the European Social Fund within the project «Support for the implementation of doctoral studies at Riga Technical University».

References

1) European Union, the United Nations and the International Transport Forum at the OECD "Glossary for transport statistics" - 5TH EDITION ISSN 2315-0815, 2019. y.

2) U.S. Department of Transportation, Bureau of Transportation Statistics "Transportation Statistics Annual Report 2018", Washington, DC: 2018. y.

3) European Commision "Sustainable and Smart Mobility Strategy – putting European transport on track for the future", Brussels, 9.12.2020 - 789 p.

4) European Commision "Europe's Rail Joint Undertaking Master Plan" – Brussel, Ref. Ares(2021)7930343 - 22/12/202.

5) https://www.era.europa.eu/activities/european-rail-traffic-management-system-ertms_en

6) Congressional Research Service "Positive Train Control (PTC): Overview and Policy Issues" - 2018. y.

7) B. Ning, T. Tang, K. Qiu, C. Gao & Q. Wang "CTCS—Chinese Train Control System" // Advanced Train Control Systems, WIT Transactions on State of the Art in Science and Engineering, Vol 46, 2010. y.

8) Economic and Social Commission for Asia and the Pacific "Inspection and monitoring of railway infrastructure using aerial drones" // Working Group on the Trans-Asian Railway Network 6th session Bangkok, 10 and 11 December 2019. y.

9) A. V. Bolshakova, A. M. Boronachin, D. Y. Larionov, L. N. Podgornaya and R. V. Shalymov, "Adaptive Safe and Energy-Efficient Driving Control of Unmanned Rail Vehicles," 2020 International Conference Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS), 2020, pp. 157-161, doi: 10.1109/ITQMIS51053.2020.9322952.

10) I. Aydin, M. Sevi, K. Sahbaz and M. Karakose, "Detection of Rail Defects with Deep Learning Controlled Autonomous UAV," 2021 International Conference on Data Analytics for Business and Industry (ICDABI), 2021, pp. 500-504, doi: 10.1109/ICDABI53623.2021.9655796.

11) M. Sevi, İ. Aydm, E. Güçlü and E. Akin, "Developing an Image Processing Based Method for Rail Tracking with an Unmanned Aerial Vehicle in a Simulation Environment," 2021 Innovations in Intelligent Systems and Applications Conference (ASYU), 2021, pp. 1-5, doi: 10.1109/ASYU52992.2021.9599085.

12) A. O. Lebedev, V. V. Vasilev, B. N. Novgorodov and A. G. Paulish, "Computer Vision Controlling an Autonomous Unmanned Aerial Vehicle Flight over a Railway," 2020 1st International Conference Problems of Informatics, Electronics, and Radio Engineering (PIERE), 2020, pp. 210-213, doi: 10.1109/PIERE51041.2020.9314659.

13) R. Gorbachev, A. Novikov, A. Kalinkin, A. Cheranev and E. Zakharova, "Applying Virtual Modelling to Verify Control Systems Decision with Artificial Intelligence in Railway Transport," 2020 International Conference Engineering and Telecommunication (En&T), 2020, pp. 1-6, doi: 10.1109/EnT50437.2020.9431256.

14) R. Thendral and A. Ranjeeth, "Computer Vision System for Railway Track Crack Detection using Deep Learning Neural Network," 2021 3rd International Conference on Signal Processing and Communication (ICPSC), 2021, pp. 193-196, doi: 10.1109/ICSPC51351.2021.9451771.

15) G. Zhong, K. Xiong, Z. Zhong and B. Ai, "Internet of things for high-speed railways," in Intelligent and Converged Networks, vol. 2, no. 2, pp. 115-132, June 2021, doi: 10.23919/ICN.2021.0005.

16) E. Essa, B. A. Abdullah and A. Wahba, "Improve Performance of Indoor Positioning System using BLE," 2019 14th International Conference on Computer Engineering and Systems (ICCES), 2019, pp. 234-237, doi: 10.1109/ICCES48960.2019.9068142.

17) M. S. Ifthekhar, N. Saha and Y. M. Jang, "Neural network based indoor positioning technique in optical camera communication system," 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2014, pp. 431-435, doi: 10.1109/IPIN.2014.7275513.

18) J. Zou, C. Wang and Y. M. Wang, "The development of indoor positioning aerial robot based on motion capture system," 2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE), 2016, pp. 380-383, doi: 10.1109/ICAMSE.2016.7840347.

19) J. Yan, G. Bellusci, C. Tiberius and G. Janssen, "Analyzing non-linearity effect for indoor positioning using an acoustic ultra-wideband system," 2008 5th Workshop on Positioning, Navigation and Communication, 2008, pp. 95-101, doi: 10.1109/WPNC.2008.4510362.

20) N. Pakanon, M. Chamchoy and P. Supanakoon, "Study on Accuracy of Trilateration Method for Indoor Positioning with BLE Beacons," 2020 6th International Conference on Engineering, Applied Sciences and Technology (ICEAST), 2020, pp. 1-4, doi: 10.1109/ICEAST50382.2020.9165464.

21) S. Sophia, B. M. Shankar, K. Akshya, A. C. Arunachalam, V. T. Y. Avanthika and S. Deepak, "Bluetooth Low Energy based Indoor Positioning System using ESP32," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021, pp. 1698-1702, doi: 10.1109/ICIRCA51532.2021.9544975.

22) Mikhail Gorobetz, Jurijs Timofejevs, Ambuja Bangalore Srinivasa, "Vehicle Distance and Speed Estimation Algorithm for Computer Vision Sensor System" 2021 IEEE 62nd International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON), 2021, pp. 1-5, doi: 10.1109/RTUCON53541.2021.9711733.