2023 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
Tomar, Portugal – April 26-27, 2023

2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) | 979-8-3503-0121-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICARSC58346.2023.10129557

Real-time GPS indoor for USV tracking using Lookup Table

Fábio Vieira[1], Pedro Teodoro[2] and Pedro Mendes Jorge[3]

***Abstract*—This study introduces an approach that utilizes Lookup Tables (LUT) to enable real-time tracking of an Unmanned Surface Vehicle (USV) in an indoor setting, using a fish-eye camera. The proposed method streamlines image processing and achieves O(1) complexity, significantly reducing application run time. The paper also outlines the process of calibrating the fish-eye camera to correct image distortion, computing the homography matrix for re-projection, and obtaining a virtual top view of the camera's field of view. The paper provides a detailed explanation of the replacement of the undistortion and re-projection steps with the new LUT method. Experimental results demonstrate a significant enhancement in the process's run time, making it feasible for real-time tracking, regardless of the image size.**

***Index Terms*—Lookup Table, GPS indoor, SIFT, real-time tracking, USV pose estimation**

I. Introduction

The work developed within the scope of this paper is part of the project Sea2Future [1] where the development of an autonomous surface vehicle, USV-enautica1 [2], is proposed (see Fig. 1).

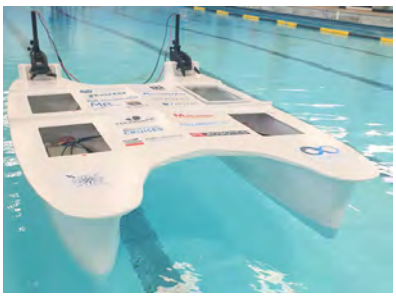


Fig. 1: Unmanned Surface Vehicle: USV-enautica1

To evaluate the autonomous driving of the USV, it is necessary to test it in controlled environments and know its position and orientation in real-time. In this project, an indoor pool is used. Indoor localization is a research topic with many applications, including tracking objects. A well-known method is the Global Navigation Satellite System (GNSS), but it is not suitable for indoor environments. Several alternatives have been proposed, such as infrared, computational vision, among others [3].

Studies show infrared accuracy is impacted by multi-path error and the covered distance [4–7]. Ultrasound performance is affected by noise interference and environmental conditions [8–10]. Due to the widespread of WLAN infrastructure, several applications use this technology. This technology has a low accuracy which must be improved by adding more routers/access points, increasing its complexity and costs [11–13]. Laser systems are another alternative, but these systems can be costly and have more complex implementations [14, 15]. Imaged-based systems work by using markers [16], features detection [17], segmentation [18] or through optical flow analysis [19]. A vision-based technique was preferred to install the system decoupled from the vessel and at some distance from the pool. A simple solution was pretended for fast integration in the USV. Hence, a vision-technique for accurate real-time indoor tracking with $O(1)$ complexity is proposed.

[1]First author, Instituto Superior de Engenharia de Lisboa
[2]Escola Superior Náutica Infante D. Henrique
[3]Instituto Superior de Engenharia de Lisboa, NOVA LINCS - ISEL

979-8-3503-0121-2/23/$31.00 ©2023 IEEE

II. Materials and Methods

A common solution regarding tracking in an indoor environment using vision is through the use of a fixed camera. The methodology consists in calibrating the camera followed by re-projecting the region of interest [20]. The proposed method for real-time USV tracking uses image processing to create a lookup table, to replace the undistortion and re-projection steps in each frame with direct indexation to obtain the position (x, y) and orientation (θ) of the vessel (see Fig. 2).

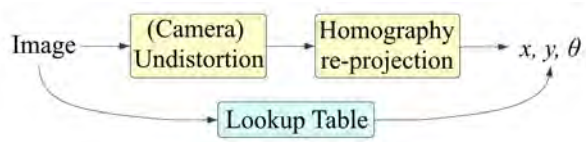


Fig. 2: Replacing undistortion and re-projection steps with a Lookup Table

A. Setup

The work was conducted in an indoor pool using the USV with LED markers installed. Two LED are positioned in the front of the vessel with 1 meter between them while the back only has one at 1.5 meters from the front markers. These positions allow detecting the front and back by comparing the distance between each LED.

A fish-eye camera is also used in order to capture the pool and vessel (see Fig. 3(b)). The camera is connected to a servo motor to rotate the camera to the desired position. When the camera is not used, it is rotated to a protected position.

B. Camera calibration

The project needs a camera with a field of view able to detect the vessel in the entire pool. To accomplish this, a

fixed fish-eye camera was used, connected to a Raspberry Pi microcontroller (see Fig. 3(a)). This type of lens introduces distortion to each captured image (see Fig. 3(b)). For accurate pose estimation, undistorted images are required to obtain valid measurements.

(a) Raspberry Pi and fish-eye camera

(b) Pool image acquired by the setup

Fig. 3: (a) Raspberry Pi and the fish-eye camera used. (b) Pool obtained with fish-eye camera

The process of finding the camera parameters is known as camera calibration. Once those parameters are found, they can be used to remove any distortion a camera might add to an image [21]. Camera parameters are composed of extrinsic and intrinsic parameters (Fig. 4).

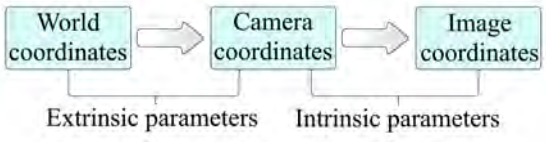


Fig. 4: Camera extrinsic and intrinsic parameters

The parameters are used to undistort the images (see Fig. 5).

Fig. 5: Fig. 3(b) undistorted

C. Homography transformation

To obtain the undistorted top view of the pool for correct vessel trajectory, the image plane must undergo homographic transformation with the plane at the same height as the vessel markers. To estimate the homography, it is necessary 4 pairs of corresponding points [22]. The four pool corners satisfy these requirements. The points are then used to estimate the homographic matrix and are transformed into the four image corners. This way, the vessel pose estimation is straightforward.

To correct small errors in the camera position caused by its rotation, the base image pool corners are registered and the displacement vector between the base and the current image is estimated by computing *SIFT* key-points [23] in both images and using a matching algorithm. The camera position suffers small oscillations so feature search is limited to a small window. False matches are filtered with a *Ratio Test* [23]. This test consists of comparing the closest match with the second closest match. If the difference between these two matches is not big enough, then they are all discarded, otherwise, noise might be included.

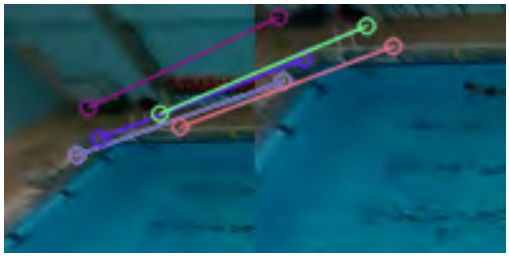


Fig. 6: Matches after *Ratio Test*

The average vector is calculated and added to the position of the base image corner to obtain its position in the current image.

The four pool corners are the four source points for the homography transformation while the destination points are the four corners of the image. The output image can be seen in Fig. 7.

Fig. 7: Re-projected image

D. Lookup Table

The camera calibration and homography matrix estimation occur once when the system starts but the undistortion and re-projection occur for every image in real-time applications. This can be time-consuming, so a Look-Up Table (LUT) is created to reduce the runtime by replacing the image processing steps with an indexing one (Fig. 2).

First, a normalized image is created (see Fig. 8), a colored image where each pixel is assigned a value between 0 and 1 based on its position in the image. Only two color channels are required, so the third channel is ignored by assigning it a value of 1. For the other two channels, each value is based on

the pixel's row and column indices. This normalized image has a resolution of 640 pixels by 480 pixels.

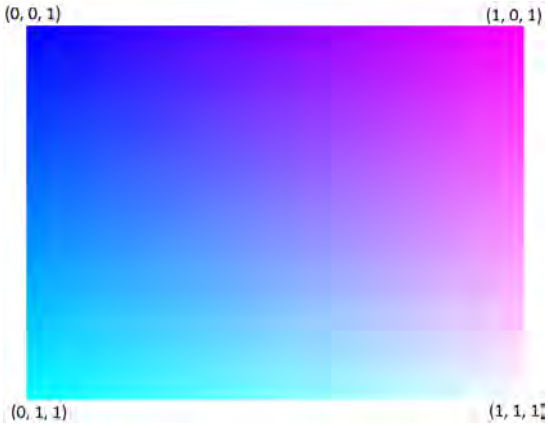


Fig. 8: Normalized image

The undistortion and homography transformations are then applied to the normalized image, in that order (see Fig. 9).

(a) After applying undistortion (b) After applying homography transformation

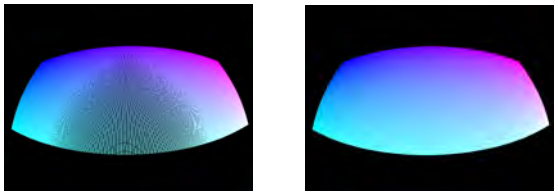
Fig. 9: (a) Result of applying undistortion. (b) Result of applying homography transformation.

The matrix obtained provides information in order to build the LUT. Its values represent the coordinates of the normalized image and so it's possible to build the LUT based on the corresponding locations of the initial and final pixel positions. The LUT is therefore:

$$LUT(D(i,j)(0), D(i,j)(1)) = N(i,j) \quad (1)$$

Where D is the matrix obtained after applying the undistortion and reprojection steps, N is the normalized image and i, j represent each index/position. The N values are multiplied by the pool dimensions to convert to real world positions.

As it can be seen in Fig. 10(a), the LUT is missing pixels but is fixed by applying a *Morphological Closing* operation (see Fig. 10(b)).



(a) Incomplete Lookup Table (b) Lookup Table

Fig. 10: (a) Incomplete LUT. (b) Completed LUT

After detecting the vessel LED in the original image, the LUT is used to obtain the corresponding points in the undistorted, re-projected image.

E. USV tracking

To detect a vessel in a controlled environment, a simple and computationally efficient method is to use active markers (LED) in the vessel. The camera's sensitivity to light is adjusted to enhance the LED in the image. Since the LED are red, the vessel can be detected by highlighting this feature, by using the Hue plane of the HSV format.

The centroid of each active region is computed and the LUT applied. The position and orientation can then be estimated using the obtained points. To do this, the distances between the markers are compared. The two closest markers represent the front of the vessel while the remaining one represents the back. The average between the two front markers is used to obtain a single point to represent the front of the vessel. Having the front and the back points of the vessel, the position (average between both points), and the orientation of the vessel can be calculated as well. Fig. 11 shows the steps to estimate the vessel's pose.

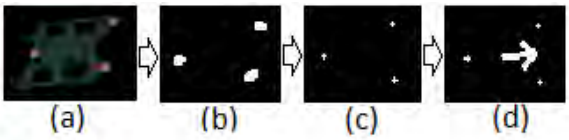


Fig. 11: (a) Vessel with LED. (b) LED detected - active regions. (c) Points obtained after applying LUT. (d) Computed vessel position and orientation

III. Experimental results

One of the main goals of this paper is to compute the correct vessel position and orientation for real-time applications, for example, automatic navigation. To validate the proposed method, the two approaches, i) image processing algorithms (regular) and ii) LUT access, are tested and the computation time is compared. Each method is executed 10 times and the time it takes is registered. Fig. 12 shows the results, executing in a Raspberry Pi Zero 2 for different image resolutions.

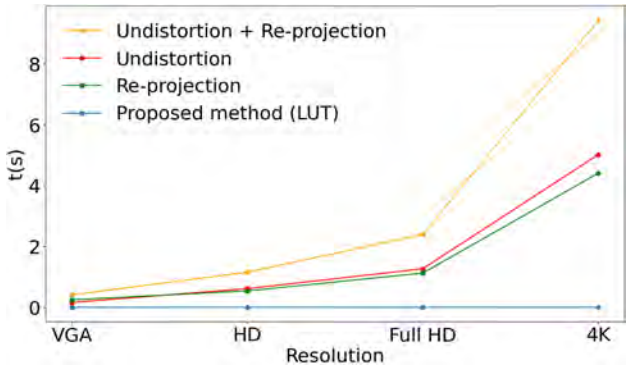


Fig. 12: Comparison between regular method steps and proposed LUT method

The regular approach's time increases with the image resolution and shows an $O(n^2)$ complexity. The LUT method's time is almost constant independently of the image size ($O(1)$ complexity). Table I shows the detailed times for both approaches.

	VGA	HD	FHD	4K
Undistortion	0.1620 ±0.0007	0.6153 ±0.0046	1.2685 ±0.0038	5.0127 ±0.0097
Re-projection	0.2496 ±0.0004	0.5404 ±0.0014	1.1266 ±0.0038	4.4036 ±0.0373
Undistortion +Re-projection	0.4116 ±0.0011	1.1557 ±0.0060	2.3951 ±0.0076	9.4163 ±0.0470
Proposed method (LUT)	0.0012 ±6.4e-05	0.0012 ±0.0001	0.0012 ±0.0002	0.0012 ±0.0002

TABLE I: Time in seconds for regular method steps and proposed LUT method

These results prove the proposed LUT method is the best approach independently of the image size used, achieving an $O(1)$ complexity.

The camera is not placed on top of the pool. This means there are fewer pixels to represent the farthest side of the pool. Applying the undistortion and homography transformation steps to the normalized image creates an image that represents the undistorted and re-projected pool. After these steps, the mm/pixel ratio can be obtained by computing the distance between each position value. The pool is 25 per 10 meters. A scale in decimeters was chosen so it is necessary to divide 100 by the calculated distance to convert to millimeters/pixel, for both axes (see Fig. 13 and 14).

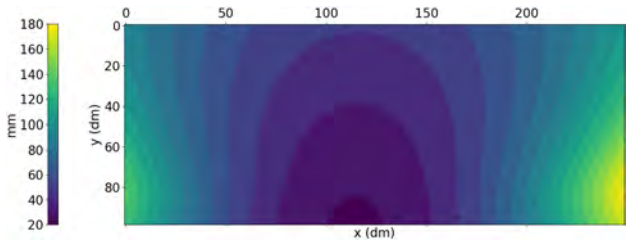


Fig. 13: Horizontal resolution (mm/px)

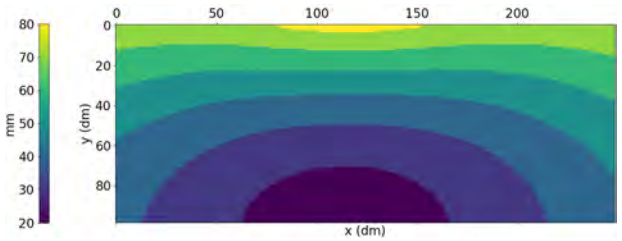


Fig. 14: Vertical resolution (mm/px)

In Fig. 13, the bottom center is the area with less mm/pixel. This is the area closer to the camera. The ratio is bigger in the farthest areas from the camera (in the x axis). The same analysis can be done for Fig. 14 for the y axis. The intersection between these two graphs (see Fig.15) allows us to better understand the best pool areas for the vessel to navigate.

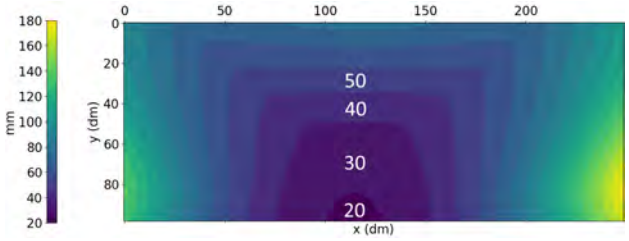


Fig. 15: Intersection between Fig. 13 and Fig. 14

In the area of a smaller mm/pixel ratio, an error of 1 pixel means an error of $\leq$ 20mm in the computed vessel position. The error increases if the vessel moves away from the camera. The center area of the pool, where the error is $\leq$ 30 and 40mm, is the best area for navigation. The remaining areas, either have a bigger ratio or are too close to the pool limits and must be avoided to prevent collisions.

To study the localization in different pool areas, the vessel was remotely controlled to navigate different predefined trajectories (see Fig. 16).

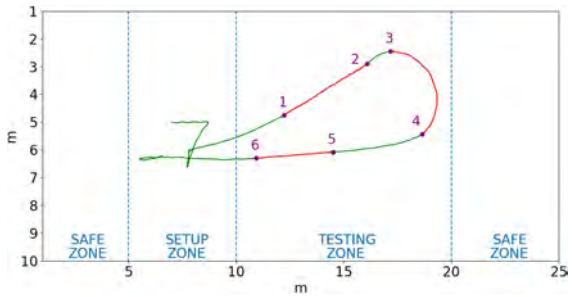


Fig. 16: Vessel navigating predefined trajectories

Fig. 16 shows a vessel route in the pool with the objective of navigating first a straight line (trajectory 1-2), a curve (trajectory 3-4) and then a second straight line (trajectory 5-6). Each trajectory is compared with its ideal trajectory. The

ideal trajectory for each of the two straight lines is computed by using the first and last points of each trajectory. For the curve trajectory, an arc passing between the first and last point is computed. For each point of each navigated trajectory, the closest point of the ideal trajectory is computed and the position and orientation errors are calculated.

Fig. 17 shows the position and orientation errors computed for trajectory 1-2.

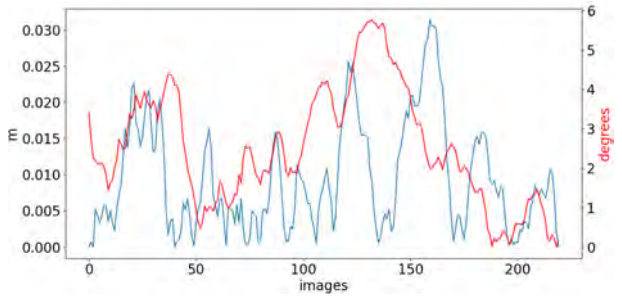


Fig. 17: Trajectory 1-2: pose errors

Fig. 18 shows the position and orientation errors computed for trajectory 3-4.

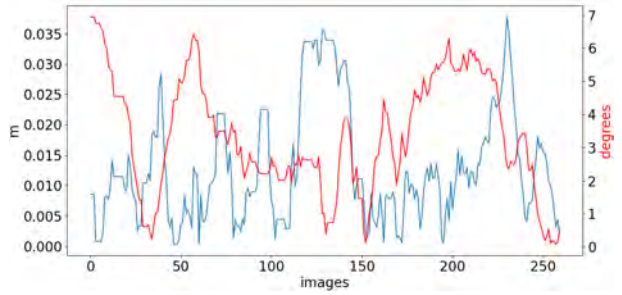


Fig. 18: Trajectory 3-4: pose errors

Fig. 19 shows the position and orientation errors computed for trajectory 5-6.

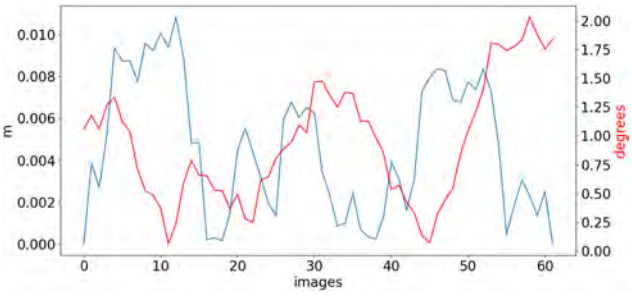


Fig. 19: Trajectory 5-6: pose errors

The position (table II) and orientation (table III) errors were used to calculate the Mean Absolute Error (MAE) to compare the different trajectories.

Trajectory	1-2	3-4	5-6
MAE	9.6 mm	12.92 mm	4.60mm

TABLE II: MAE for position errors

Trajectory	1-2	3-4	5-6
MAE	2.51º	3.35º	0.94

TABLE III: MAE for orientation errors

Analyzing the results, the first (1-2) and second (3-4) trajectories are the ones with the bigger errors in both position and orientation. The errors were calculated for a worst-case scenario where the acquired image has a VGA resolution.

Fig. 20 shows the overlay between Fig. 15 and the vessel trajectory. This allows the comparison between the trajectory and the pool areas where a bigger error might occur.

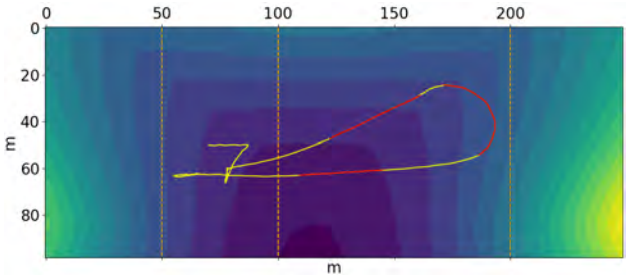


Fig. 20: Overlay between Fig. 15 and the vessel trajectory

These findings confirm that the best area to perform navigation is the one closer to camera where the trajectory 5-6 is performed. This trajectory is the one with less errors in both position and orientation and the least affected by outliers as well.

IV. Conclusion

This paper presents an algorithm to detect the position and orientation of a vessel in a indoor pool using a computer vision approach. A fish-eye lens camera to detect the entire pool area was used. A method, with a Lookup Table, is proposed to replace the image processing steps: undistortion and re-projection. The results prove that a Lookup Table is the logical alternative by improving the run time of the application with constant times independently of the acquired image size ($O(1)$ complexity). The results also present the mm-to-pixel ratio to show that is better to perform vessel navigation closer to the camera. The conducted tests confirm this by comparing the position and orientation errors of multiple trajectories.

V. Acknowledgments

The authors would like to thank Escola Superior Náutica Infante D. Henrique, its robotic group and, the project Sea2Future.

REFERENCES

[1] *Sea2Future*. URL: https : / / sea2future . pt/. (visited: 18/06/2022).

[2] Mário Assunção et al. "Design of an Underactuated USV Catamaran". In: *APCA International Conference on Automatic Control and Soft Computing*. Springer. 2022, pp. 656–666.

[3] Luca Mainetti, Luigi Patrono, and Ilaria Sergi. "A survey on indoor positioning systems". In: *2014 22nd international conference on software, telecommunications and computer networks (SoftCOM)*. IEEE. 2014, pp. 111–120.

[4] Taiga Arai et al. "Evaluation of indoor positioning system based on attachable infrared beacons in metal shelf environment". In: *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE. 2019, pp. 1–4.

[5] Erwin Aitenbichler and Max Muhlhauser. "An IR local positioning system for smart items and devices". In: *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.* IEEE. 2003, pp. 334–339.

[6] Thibaut Raharijaona et al. "Local positioning system using flickering infrared leds". In: *Sensors* 17.11 (2017), p. 2518.

[7] Jens-Steffen Gutmann et al. "Challenges of designing a low-cost indoor localization system using active beacons". In: *2013 IEEE conference on technologies for practical robot applications (TePRA)*. IEEE. 2013, pp. 1–6.

[8] Seong Jin Kim and Byung Kook Kim. "Dynamic ultrasonic hybrid localization system for indoor mobile robots". In: *IEEE Transactions on Industrial Electronics* 60.10 (2012), pp. 4562–4573.

[9] Faheem Ijaz et al. "Indoor positioning: A review of indoor ultrasonic positioning systems". In: *2013 15th International Conference on Advanced Communications Technology (ICACT)*. IEEE. 2013, pp. 1146–1150.

[10] Carlos Medina, José Carlos Segura, and Angel De la Torre. "Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy". In: *Sensors* 13.3 (2013), pp. 3501–3526.

[11] José Luis Carrera et al. "A real-time indoor tracking system in smartphones". In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 2016, pp. 292–301.

[12] Weixiao Meng et al. "Optimized access points deployment for WLAN indoor positioning system". In: *2012 IEEE wireless communications and networking conference (WCNC)*. IEEE. 2012, pp. 2457–2461.

[13] Fan Wang et al. "EESM-based fingerprint algorithm for Wi-fi indoor positioning system". In: *2013 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE. 2013, pp. 674–679.

[14] Ajo Fod, Andrew Howard, and MAJ Mataric. "A laser-based people tracker". In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*. Vol. 3. IEEE. 2002, pp. 3024–3029.

[15] Zheng Wang et al. "Experimental comparison of dynamic tracking performance of iGPS and laser tracker". In: *The international journal of advanced manufacturing technology* 56 (2011), pp. 205–213.

[16] Dawar Khan, Sehat Ullah, and Syed Nabi. "A generic approach toward indoor navigation and pathfinding with robust marker tracking". In: *Remote Sensing* 11.24 (2019), p. 3052.

[17] Jason Zhi Liang et al. "Image based localization in indoor environments". In: *2013 Fourth International Conference on Computing for Geospatial Research and Application*. IEEE. 2013, pp. 70–75.

[18] Jae-Hong Shim and Young-Im Cho. "A mobile robot localization using external surveillance cameras at indoor". In: *Procedia Computer Science* 56 (2015), pp. 502–507.

[19] Daniel Gehrig et al. "Asynchronous, photometric feature tracking using events and frames". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 750–765.

[20] João Dias and Pedro Mendes Jorge. "People tracking with multi-camera system". In: *Proceedings of the 9th International Conference on Distributed Smart Cameras*. 2015, pp. 181–186.

[21] Juho Kannala and Sami S Brandt. "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses". In: *IEEE transactions on pattern analysis and machine intelligence* 28.8 (2006), pp. 1335–1340.

[22] Elan Dubrofsky. "Homography estimation". In: *Diplomová práce. Vancouver: Univerzita Britské Kolumbie* 5 (2009).

[23] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.