2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Autonomous RGBD-based Industrial Staircase Localization from Tracked Robots

Jérémy FOURRE[1] Vincent VAUCHEY[1] Yohan DUPUIS[1] Xavier SAVATIER[1]

***Abstract*— This paper presents an industrial staircase localization algorithm based on RGBD data from a tracked robot. This situation is really challenging as the camera is placed close to the ground. Moreover, RGBD can be really noisy on sparse staircases. Contrary to existing works, our evaluation relies on ground truth data provided by a motion capture system. Our experiments suggest that our algorithm can robustly locate industrial staircase. We also propose a new framework to evaluate stair localization performance from RGBD data. The overall performance allows to safety control a robot to rally the staircase.**

I. Introduction

Autonomous systems are required to be able to move in complex environments. They are either dedicated to indoor or outdoor applications. Moving in complex environments requires perceiving, analyzing and acting according to the surrounding environment and the task to be achieve. Field robotics works are now able to tackle highly irregular environments. Field robotics applied to multiple levels building involve to move through stairs. Still, staircases remain really challenging. In fact, they can be narrow, steep and result in low grips both for wheeled and tracked robots.

3D detection and location of staircase are actually the first step necessary to properly reach and start climbing staircase. Robot location with respect to the staircase must be accurately determined in order to align the robot with the first stair. It allows the robot to travel in staircases as narrow as the robot width.

Several modalities are available to detect and locate stairs from low robots. They can be LiDAR-based or vision-based.

First, several LiDAR-based technologies have been investigated. Fair and Muller [2] used range measurement sensors placed below a track robot to detect stairs. The robot must be descending the staircase and be able to stand on the stair to apply a detection based on a height gap. As a result, it cannot be applied in the reaching phase of the staircase. Mihankhah *et al.* [8] have used a single-layer LiDAR to detect and climb stairs. The LiDAR scans vertically and detects layers of the staircase. The detection was applied on a regular staircase with plain stair riser and tread. Still, it may not be applicable for industrial staircases with no riser and drilled tread.

Secondly, vision based and its variants have been explored. Mourikis *et al.* [9] used stair detection to servo a robot while climbing the staircase. Edge detection is applied in the camera image to find the stair direction with respect to the robot and control the robot accordingly. A similar approach has been implemented by Carbonara and Guaragnella [1] to detect stairs from a camera placed on an impaired person. The camera image encompasses a large part of the staircase as the camera is placed at a person height. The edge detection result is combined with a FFT to estimate the stair case location. Consequently, the staircase can be properly located by merging the detection of several stairs. Shahrabadi *et al.* [12] also employed an edge detection framework and a Hough transform to find the main stair direction. Hesch *et al.* [6] tackled descending stairs detection by mixing edge detection and optical flow. The work has been applied on a broad and plane staircase. Results have been achieved for a tracked robot descending from a floor into the staircase. While preceding works relied on 2D information, vision-based 3D information can be computed from several vision sensors. It allows detecting the staircase states, namely location and direction, in 3D. Gutmann *et al.* [4] placed a stereo camera in the head of a humanoid robot. The robot was assumed to be roughly aligned in front of the stair case. Planes resulting from the riser and tread were then detected. Schwarze and Zhong [11] combined the edge detection from the 2D images and the disparity map to detect the tread plane. Both methods use high views of the staircase in addition to features like plain riser and tread. RGBD camera can directly output 3D information at high rates as the processing is offloaded on the device itself. Wang *et al.* [14] employed a RGBD camera worn on a person belt to detect stairs and crosswalks. The RGB data is use to detect parallel lines in the scene. Then, the ambiguity between crosswalks and stairs is leveraged with the depth data. Perez-Yus *et al.* [10] placed a RGBD camera on a person chest to detect stairs. Normal estimation is performed on the down-sampled depth data. Then region growing is applied to find the planes resulting from plain treads and risers. Stairs are then extracted given geometric constraints.

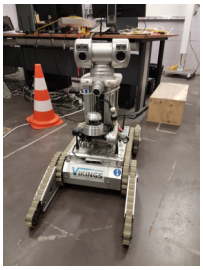


Fig. 1: Tracked robot equipped with a RGBD camera

[1]The authors are with Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, Rouen, France.firstname.lastname@esigelec.fr

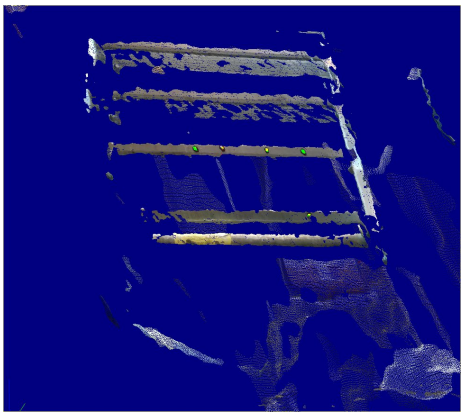


Fig. 2: 3D View of RGBD data for an industrial staircase observed from the tracked robot

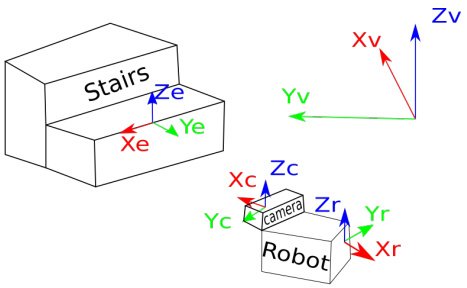


Fig. 3: Reference frames used in this paper.

In this paper, we propose a stair detection and location algorithm for tracked robots. The algorithm is designed to work for both regular and industrial staircases. Moreover, we investigated the scenario where a single sensor must be able to handle for climbing and descending detection as well as overall scene perception. As a result, the sensor must be placed low on the robot to be able detect descending stairs especially when the tracked robot is not equipped with rear flipper. Negative pitch cannot be reached. The low sensor location produces a field of view focused on the first two stairs on the staircase. The stair tread cannot be seen. The proposed method is able to deal with this complex situation. To the best of our knowledge, all existing works on stair location assessments are qualitative. Our evaluation has been carried out with a VICON system in order to provide quantitative results. Our results provide insights on the performance of our approach on real staircases.

The paper is organized as follows. Section II presents a flexible and realtime framework that allows locating stairs from a tracked robot. Section III qualifies the performance of our method on regular and industrial staircase. Section IV concludes the paper.

II. Methodology

First, let us define the different frames used in the paper (c.f. Figure 3). Our paper provides the ground truth as given by a VICON system. Its reference frame is named $\mathcal{R}_v$ and will be associated to the world frame. The staircase state in $\mathcal{R}_v$ is represented by $\mathcal{R}_e$. It is a six-state vector including position and orientation. The camera pose $\mathcal{R}_c$ attached to the robot $\mathcal{R}_r$ are also expressed in $\mathcal{R}_v$.

While RGBD cameras do not suffer from the density problem encountered with LiDARs due to low vertical sampling density, it can suffer from a 3D measurement noise and lack of depth information when the vision-based triangulation fails (c.f. Figure 2). This is particularly true for industrial staircase with rather narrow stair riser. Our proposed staircase location method aims at tackling this particular issue

A. Staircase localization

The algorithm is based on the depth information of the RGBD camera. As shown in Figure 3, the x-axis, y-axis and z-axis of the camera point respectively front, left, up. Stairs can be assimilated as a jump in depth especially from our low point of view. In fact, only stair nosing can be seen. Our algorithm will be looking for this information.

Perez-Yus *et al.* [10] applied spacial filters on planes namely vertical spacing between horizontal planes and horizontal spacing between vertical planes. As shown in Figure 2, the quality of the depth data can be poor. Moreover, only the stair nosing is available for industrial staircases.

We aim at robustly locating the first stair nosing as given by $\mathcal{R}_e$ in Figure 3 and the vector associated to $\vec{X_e}$.

First, we sort points in ascending order along each columns according to their x values. It aims at removing empty pixels, i.e. pixels with no depth information due to triangulation errors. Succeeding points are compared each other. As soon as a gap between τ_min and τ_max is encountered, the point value $X = \{x, y, z\}$ is stored in the 3D point candidate list L for the 3D line corresponding to the stair nosing N. The parsing process along the column is then stopped. Each column being processed independently, this step can easily run in parallel.

The list L is then used to robustly estimate the 3D line N associated with the stair nosing. Any points p_r belonging to the estimated stair segment is defined as follows :

$$p_r = r_r + k.\vec{e_r} \tag{1}$$

where :

r_r	location of the estimated segment origin
e_r	estimated segment direction
k	scalar value between 0 and 1 included

$\vec{e_r}$, the direction vector of the line, is actually equal to X_e or $-X_e$.

We used RANSAC [3] to find the best line from L. Two points P_1 and P_2 were randomly picked. One point was set to be P_0. $\vec{e}$ is equal to $P_2 - P_1$. The remaining points P_i in L are used to find inlier candidates using the following distance formula:

$$d_i = \frac{|P_i - P_2| \cdot |P_i - P_1|}{|P_2 - P_1|} \tag{2}$$

Algorithm 1 First Stair Detector

procedure FIRSTSTAIRSDETECTOR(I,XYZ)
  I ← image RGB
  XYZ(p) ← position (X,Y,Z) associated with pixel p of image I
  N,M ← size(I)
  for each column c in I **do**
    $p_0 = I(N, c)$
    $n = N$
    $i = n - 1$
    $p_1 = I(i, c)$
    while $i > 0$ **do**
      $n = i - 1$
      for $j = i - 1, j > 0, j = j - 1$ **do**
        if $X(I(j, c)) < X(p_1)$ **then**
          $p_1 = I(j, c)$
          $n = j$
        end if
      end for
      if $x_{min} < X(I(p_1) - X(p_0) < x_{max}$ **then**
        $P \leftarrow XYZ(p_0)$ ▷ add the point to the first step candidates
        $i = -1$
      else
        $i = n$
      end if
    end while
  end for
  RANSAC(P)
end procedure

A point is regarded as an inlier if d_i is less than τ_{inlier}. The line with the maximum number of inliers is chosen as the 3D line corresponding to the stair nosing. The location of the staircase actually corresponds to the middle of the two extreme inlier points, namely left-most and right-most. $\vec{X_e}$ is either equal to $\vec{e}$ or $-\vec{e}$ depending if $\vec{e}$ is positively oriented along the y-axis of the camera. The full algorithm is shown in Algorithm 1

B. Staircase location performance estimation

While existing papers on staircase detection provide a qualitative analysis of the performance, we have chosen to propose a quantitative analysis. We have used a VICON system to locate the staircase accurately. The VICON system gives a millimeter grade accuracy. Still, the staircase location is expressed in the VICON reference frame $\mathcal{R}_v$. The staircase measurement is done in the camera reference frame$\mathcal{R}_c$. It is therefore necessary to be able to express the points of the camera frame in the VICON frame as follows:

$$P_v = H_{v\to c} \cdot P_c \tag{3}$$

where:
P_v point in the VICON frame
P_c point in the camera frame
$H_{v\to c}$ is actually unknown. From Figure 3, it can be seen that the transform can be estimated using the robot transform in the VICON frame $H_{v\to r}$. As a matter of fact, the robot can be equipped with VICON marker whereas it is impossible to directly observe to camera location in the VICON frame. Equation 3 can be updated as:

$$P_v = H_{v\to r} \cdot H_{r\to c} \cdot P_c \tag{4}$$

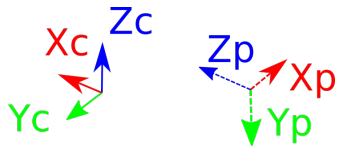


Fig. 4: Camera frame $\mathcal{R}_c(X_c, Y_c, Z_c)$ and pinhole frame $\mathcal{R}_p(X_p, Y_p, Z_p)$

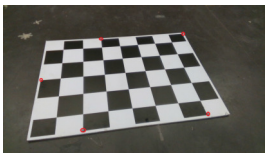


Fig. 5: Checkerboard used in the 2D-2D method
VICON markers are circled in red

where :
$H_{r\to c}$ Transformation from the camera frame to the robot frame
$H_{v\to r}$ Transformation from the robot frame to the VICON frame

$H_{v\to r}$ is directly given by the VICON system. We have investigated several methods to obtain $H_{r\to c}$ based on RGBD and VICON data.

1) $2D-2D$ method: RGBD sensors provides RGB and depth information. Checkerboard are widely used to calibrate cameras. As a result, our first proposal uses a checkerboard embedding VICON markers (c.f. Figure 5) and RGB data. The VICON marker locations are consequently known in the checkerboard frame $\mathcal{R}_m$. The robot to camera transform estimation can be updated as follows:

$$H_{r\to c} = H_{r\to v} \cdot H_{v\to m} \cdot H_{m\to c} \tag{5}$$

$H_{v\to m}$ Transformation from the checkerboard frame to the VICON frame
$H_{m\to c}$ Transformation from the camera frame to the checkerboard frame

$H_{m\to c}$ is estimated by *PnP* [5], which provides the checkerboard pose in the camera frame.

Markers on the checkerboard are given as a list of 3D points X_m. The corresponding points are also given in the VICON frame as the list X_v. They are both rigidly related as follows:

$$X_v = H_{v\to m} \cdot X_m \tag{6}$$

Umeyama [13] algorithm has been used to estimate $H_{v\to m}$. The resulting $H_{r\to c}$ obtained by applying Equation 5 must be slightly updated as our camera reference frame does not fit the pinhole model with the following transform:

$$H^{2D}_{r\to c} = H_{r\to c} \cdot R \tag{7}$$

with :

$$R = R_x(90) \cdot R_y(0) \cdot R_z(90) \tag{8}$$

2) $3D-3D$ method: The second approach is based on the 3D information provided by the RGBD sensor. The previous approach requires the *PnP* step to obtain the 3D location of the VICON markers in the camera space. This second solution relies only on 3D information. As a matter of fact, the depth information provided and consequently is the 3D location the marker. Still, this information can be noisy as illustrated earlier in the paper. Given a set of markers placed in the FOV (Field of View) of the RGBD camera X_c and their corresponding points given by the VICON, we can estimate $H_{v\to c}$ with the Umeyama algorithm. Then, the transform from the robot to the camera $H^{3D}_{r\to c}$ can be estimated as follows:

$$H^{3D}_{r\to c} = H_{r\to v} \cdot H_{v\to c} \tag{9}$$

Our approach relies on clicking the markers in the RGBD image to obtain their 3D locations with respect to the camera. $H^{2D}_{r\to c}$ and $H^{3D}_{r\to c}$ are theoretically equal.

3) Robust $H_{r\to c}$ estimation: The robot-camera assembly ensures that $H_{r\to c}$ remains constant over time. Nevertheless, the camera data can be noisy and so is the PnP results or 3D points computation. As a consequence, several datasets must be used to robustify the transform estimation for both methods. The transform must be correctly averaged. We separated the rotation and translation components of the transform.

The final translation T is the average of the translations from each sample T_i :

$$\bar{T} = \frac{1}{n}.\sum_{i=1}^{n} T_i \tag{10}$$

Rotational components R_i have been converted into quaternions q_i in order to properly estimate the average quaternion q as:

$$\bar{q} = \underset{q\in\mathbb{S}^3}{argmin} \sum_{i=1}^{n} w_i \|A(q) - A(q_i)\|^2_F \tag{11}$$

We solved this equation based on the eigenvalue method given by Markley and al. [7]. $\bar{q}$ is then converted back to a rotation matrix. Both resulting $H^{2D}_{r\to c}$ and $H^{3D}_{r\to c}$ are based on averaged transforms.

4) Staircase location evaluation: Ground truth for the staircase location is known by placing two VICON markers on the corners of the first stair nosing. Several similarity measures can be investigated.

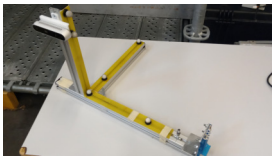
First, the mean of these two points can be defined as the staircase location.

The location error d_l can be measured as:

$$d_l = \|X_r - X_g\| \tag{12}$$

where :

X_r	Staircase location estimation found by our RANSAC-based algorithm
X_g	Staircase location ground truth



(a) Camera setup with VICON markers

(b) Example of experimentation

Fig. 6: Setup used in our experiments

Secondly, the staircase location error can be seen as the distance between the two segments d_s. This distance can be computed as follows :

$$d_s = \begin{cases} \frac{\vec{n}.(r_r - r_g)}{||\vec{n}||}, & \text{if } n \neq 0 \\ \frac{||\vec{e_{rg}} \times \vec{e_g}||}{||\vec{e_g}||}, & \text{otherwise} \end{cases} \tag{13}$$

with $\vec{n} = \vec{e_r} \times \vec{e_g}$ and $\vec{e_{rg}} = p_r - p_g$.

Thirdly, the staircase direction can be associated with the unit vector corresponding to the difference between the two markers. Any points p_g belonging to 3D segment corresponding to the ground truth stair nosing is defined as:

$$p_g = r_g + k\vec{e_g} \tag{14}$$

where:

r_g	location of the ground truth marker used as the segment origin
e_g	segment direction
k	scalar value between 0 and 1 included

The equation for the estimated segment has already been defined in Equation II-A.

The angle α between the two segments is then given as:

$$\alpha = \cos^{-1}\left(\frac{\vec{e_g} \cdot \vec{e_r}}{\|\vec{e_g}\| \cdot \|\vec{e_r}\|}\right) \tag{15}$$

III. Results

A. Experimental Setup

For the ease and completness of the experiments, we used the setup shown in Figure 6a. The RGBD camera is placed at the same height as the robot shown in Figure 1. This setup especially ensures that the VICON markers are fully visible by our VICON setup. Our VICON setup is made of 20 cameras placed in a 10x15m room. An Intel RealSense D415 was used in our experiments. The calibration pattern is an A0 7x5 square checkerboard.

Several dataset were collected. We split datasets used to estimate $H_{r\to c}$ and the dataset used to benchmark 2D and 3D approaches.

First, we will present the validation results for the estimation of the matrix H. In a second step, we will see the results obtained for the staircase detection with the matrix H estimate.

B. $H_{r\to c}$ estimation

We have used 10 samples for $H_{r\to c}$ estimation and 10 samples for $H_{r\to c}$ quality estimation. The samples corresponds to checkerboard with VICON markers placed on

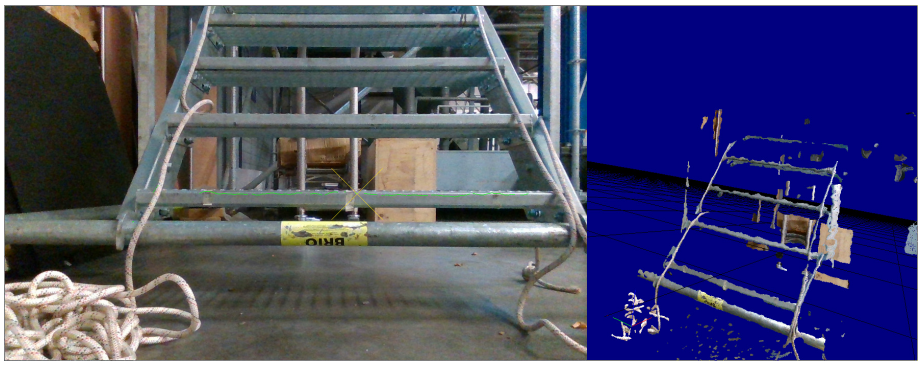


Fig. 7: Industrial stair data from the RGBD camera
The thin green line corresponds to the located stair in the RGB image

Method	Mean	Standard Deviation
2D-2D	0.0381	0.0015
3D-3D	**0.0140**	**0.0017**

TABLE I: $H_{r\to c}$ estimation benchmark
Out of 10 samples

Dataset	Distance IRON	Angle IRON
Distance (m)	$d \in \{0.6, 0.7, \cdots, 1.9\}$	1.0
Angle (°)	0.0	$a \in \{-20, -10, \cdots, 20\}$

TABLE II: Stair localization dataset

it. RGBD camera and the VICON system simultaneously record the checkerboard. Marker locations are known in the checkerboard space and the VICON space. *PnP* gives the checkerboard pose with respect to the camera. As a result, marker location P_c are given in the camera reference frame. The 3D-3D method based on marker selection in the camera image directly given marker locations in the camera reference frame. As a consequence, we can estimate marker locations in the VICON reference frame ${}^{V}\hat{P}_c$ as follows:

$$ {}^{V}\hat{P}_c = H_{v\to r} \cdot H_{r\to c} \cdot P_c \tag{16} $$

where:

$H_{v\to r}$ Transformation from the robot frame to the VICON frame provided the VICON system

$H^i_{r\to c}$ Transformation from the camera frame to the robot frame provided by either the 2D or 3D method with $i = 2D$ or $i = 3D$ respectively.

The error is then computed as the mean and standard deviation of the 3D error between the estimated and true VICON marker location on the test set. Table I reports the results. It can be seen that the 3D-3D method gives the best result with an average error of 1.4cm and a low standard deviation of less than 2mm. The standard deviation is similar for the *PnP*-based estimation is about 3 times larger. This can be due to a biased estimation of the checkerboard pose computed by the *PnP*. The 3D-3D method relies on noisy depth data from the RGBD camera and VICON data. This approach seems more accurate than the 2D-2D method. Consequently, $H^{3D}_{r\to c}$ will be used for the rest of this paper.

C. Stair localization evaluation

Stair localization evaluation was performed with two dataset. First, the setup is centered with respect to the staircase center as defined earlier. Then the distance is increased by steps of 10cm. Sensitivity to the distance is consequently investigated. Secondly, the setup is placed 1m away from the staircase. The setup is then moved about a circle to change the angle of the camera with respect to the staircase direction. The dataset is freely available[1] for benchmark purposes.

Figure 8a presents the results to the distance sensitivity study. It measures the 3D distance to the ground truth staircase center d_l. We can see that the mean error over the entire range is less than 5cm. The error is less than 5cm from 0.6m to 1.7m. Then, it increases significantly. This error is explainable by an increase in the depth uncertainty provided by the RGBD camera.

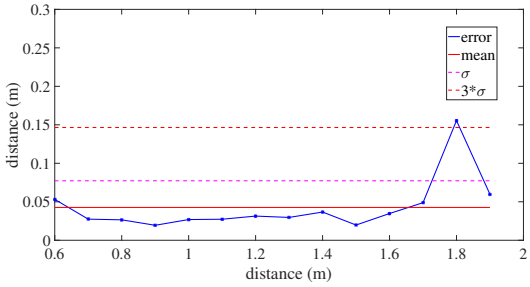
Figure 8b presents the results to the angular sensitivity study. The mean error is about 5cm. We can that the error is symmetric about zero. It tends to increase as the angle increases. This phenomenon is caused by the perspective projection. As the viewpoint angle of the camera increases, the right-most or left-most edge of the stair nosing tends to be under detected as the gap threshold conditions are not met anymore. It induces a shift in the stair center location.

We also investigated the influence of camera angle on the d_s (Equation 13). The result is shown in Figure 9a. We can see that the distance between the stair nosing segment and the estimated stair nosing location is rather small. It is actually smaller than the RGBD sensor noise. Variations with respects to angle are not significant.

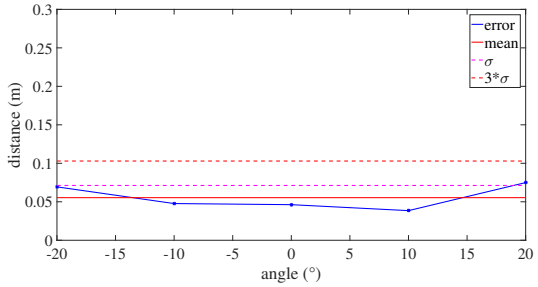
Finally, we compared the estimated noising direction with the ground truth based on the α measure (Equation 15). The angles are small over the entire angular span. It means that the direction is correctly estimated.

To sum up, we can see that the error found on the staircase center is a shift along the stair nosing direction. As a result, the impact for robot control is rather limited. The robot may be slightly shifted about the staircase center. The robot heading will be correct and so is the control to reach the staircase.

[1]https://github.com/vauchey/StaircaseLocalization

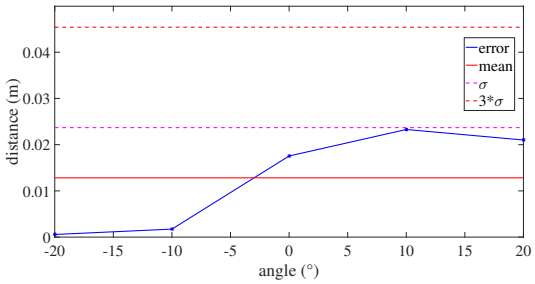


(a) d_l and sensor distance

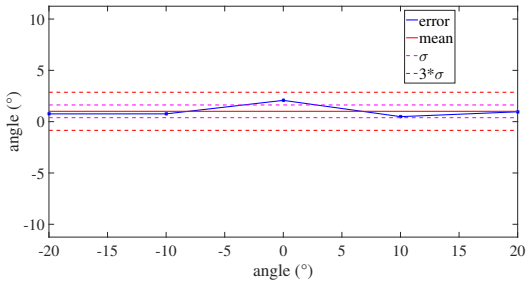


(b) d_l and sensor angle

Fig. 8: Errors on the staircase center



(a) d_s and sensor angle



(b) α and sensor angle

Fig. 9: Staircase localization sensitivity to angle

IV. CONCLUSION

In this paper, we have presented an algorithm able to localize a staircase with respect to a RGBD camera and a robot. Contrary to the state-of-the art papers, a quantitative evaluation is carried out with a motion capture setup. Our results suggests our algorithm can accurately localize an industrial staircase from tracked robots. The staircase first nosing is detected within 2cm. The nosing direction error is less than 1°. Future works will focus on several topics. First, we will evaluate our algorithm on plain staircases. We will test other point of viewer. Such point of views might allow merging the decision from several steps in order to make the localization more robust.

REFERENCES

[1] S. Carbonara and C. Guaragnella. Efficient stairs detection algorithm assisted navigation for vision impaired people. In *2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings*, pages 313–318, June 2014.

[2] M Fair and David Miller. Automated staircase detection, alignment, & traversal. In *Proceedings of the IASTED Int'l Conference on Robotics and Manufacturing*, pages 218–222, 01 2001.

[3] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[4] J. . Gutmann, M. Fukuchi, and M. Fujita. Stair climbing for humanoid robots using stereo vision. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 2, pages 1407–1413 vol.2, Sep. 2004.

[5] Bert M Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International journal of computer vision*, 13(3):331–356, 1994.

[6] J. A. Hesch, G. L. Mariottini, and S. I. Roumeliotis. Descending-stair detection, approach, and traversal with an autonomous tracked vehicle, Oct 2010.

[7] F. Landis Markley, Yang Cheng, John L. Crassidis, and Yaakov Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007.

[8] E. Mihankhah, A. Kalantari, E. Aboosaeedan, H. D. Taghirad, S. Ali, A. Moosavian, E. Mihankhah, A. Kalantari, E. Aboosaeedan, H. D. Taghirad, S. Ali, and A. Moosavian. Autonomous staircase detection and stair climbing for a tracked mobile robot using fuzzy controller. In *2008 IEEE International Conference on Robotics and Biomimetics*, pages 1980–1985, Feb 2009.

[9] Anastasios I. Mourikis, Nikolas Trawny, Stergios I. Roumeliotis, Daniel M. Helmick, and Larry Matthies. Autonomous stair climbing for tracked vehicles. *International Journal of Computer Vision & International Journal of Robotics Research - Joint Special Issue on Vision and Robotics*, 2007.

[10] Alejandro Perez-Yus, Daniel Gutiérrez-Gómez, Gonzalo Lopez-Nicolas, and JJ Guerrero. Stairs detection with odometry-aided traversal from a wearable rgb-d camera. *Computer Vision and Image Understanding*, 154:192–205, 2017.

[11] T. Schwarze and Z. Zhong. Stair detection and tracking from egocentric stereo vision, Sep. 2015.

[12] Somayeh Shahrabadi, Joao M. F. Rodrigues, and J. M. Hans du Buf. Detection of indoor and outdoor stairs. In João M. Sanches, Luisa Micó, and Jaime S. Cardoso, editors, *Pattern Recognition and Image Analysis*, pages 847–854, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[13] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, April 1991.

[14] Shuihua Wang and Yingli Tian. Detecting stairs and pedestrian crosswalks for the blind by rgbd camera. pages 732–739, 10 2012.