Article

A Novel Method for 3D Absolute Localization and Orientation of a Stair Cleaning Robot using Staircase Geometry and Onboard Sensors

Ilyas Muhammad[1,2], Vinu Sivanantham[1], Manojkumar Devarassu[1], Balakrishnan Ramalingam[1], Mohan Rajesh Elara[1,*]

[1] Engineering Product Development Pillar, Singapore University of Technology and Design (SUTD), Singapore 487372.
[2] Department of Electrical Engineering, UET Lahore 54890, NWL Campus, Pakistan.

*Correspondence: rajeshelara@sutd.edu.sg; Tel.: +65-9023-4275

Abstract: The *sTetro* is a stair cleaning robot which can climb the staircase with its shapeshifting capabilities. As this robot is intended to traverse multi-floor environment autonomously, hence its localization/positioning information is an essential component of the overall system. Usually, the indoor mobile robots rely on some external system for localization information, e.g., WiFi, UWB, vision, RFID signals, or indoor Global Positioning System (GPS). This requires the installation of additional hardware and/or modification of the working environment for precise positioning information of the mobile platform. As the dimensions of the staircase are known a priori, this knowledge can be used to localize the *sTetro* robot on the stairs. In this article, the geometry of the staircase has been exploited to localize the robot in 3D space with measurements from the onboard time-of-flight (*ToF*) range sensors only. The heading angle of the robot is also estimated with two *ToF* sensors installed in front of the *sTetro* robot. Results achieved by conducting experiments on real robot prove the efficacy of the proposed approach.

Keywords: Indoor environment; 3D localization; staircase geometry; robotics; time-of-flight sensors.

1. Introduction

Any mobile robotic system comprises of many systems and subsystems for its proper operation, however, some significant modules are essential for the autonomous operation of the mobile platforms [1], [2]: *control system module*, *path planner module* and *localization/positioning module*. Among them, the localization module is responsible for providing correct information of position and orientation [3]. The other significant modules of the robotic system mainly depend on the accuracy of the localization system.

Mainly there are two types of localization/positioning systems, 1) Absolute positioning system, 2) Incremental/relative positioning system [4],[5]. In absolute positioning system the position information is available with respect to some global reference coordinate frame and usually does not depend on time and space, i.e., position in not drifting with time, e.g., GPS [6], [7]. However, in the incremental or relative positioning system, the position update is obtained from the previous step increments. The drawback of later is that any errors in the previous position and the sensors are also aggregated, i.e., position drift with time, e.g., IMU and wheel encoder-based position calculation [8], [9].

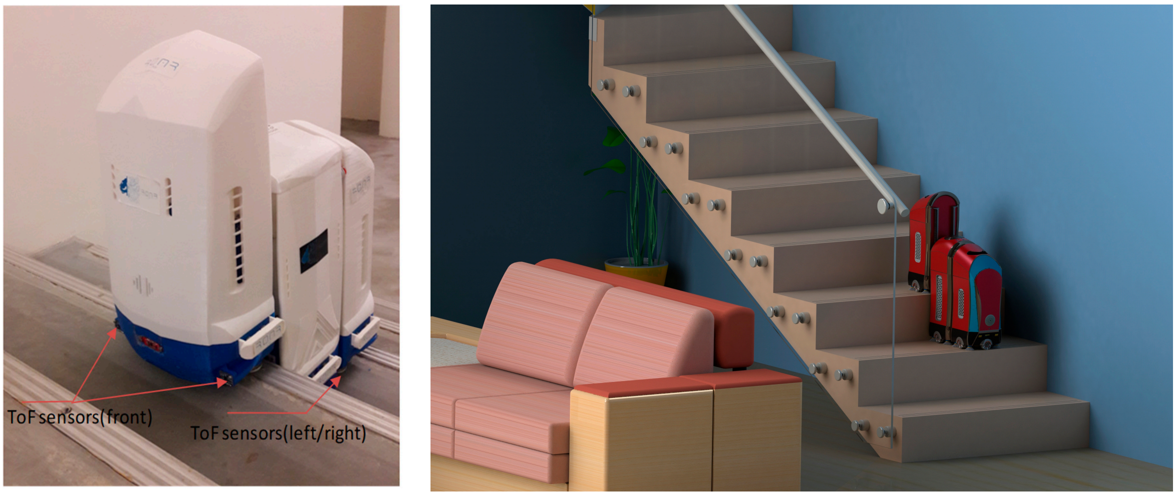
For outdoor mobile robots, the localization information mainly comes from GPS alone or combined with other sensors/systems, e.g., inertial measurement unit, camera, Lidar, vision sensors etc. [10], [11]. However, for indoor mobile robotic platforms, GPS is not readily available, hence other forms of absolute positioning are employed, e.g. *WiFi* signals are used to estimate absolute position ofmobile devices[12]. Similarly, other beacons-based absolute positioning systems are employed to get the absolute position of a mobile platform in an indoor environment [3], [6], [13]–[18]. However, this requires some additional hardware modifications in the working environment, which adds additional cost and/or complexity of the overall system.

In this work, we present a novel yet very simple technique to estimate the 3D absolute localization information of an indoor mobile robot, called *sTetro*, developed at *RoAR* Laboratory, SUTD [19]. Sensor installation and typical working environment of this robot is shown in Figure 1. As the working environment of the *sTetro* robot is a staircase which has fixed and known dimensions of each step's tread and riser. This a prior knowledge of staircase geometry is exploited for localization information estimation along with onboard sensors measurements only. Some of the terminologies related to staircase geometry are given in Figure 2. Two time of flight (*ToF*) sensors are installed on the left and right side of the robot. The distance measurements obtained from two *ToF* sensors are used to calculate y-axis position on the tread. As the dimensions of stair's tread and riser are known, the x-axis and z-axis position is derived from the known geometry of the staircase as robot climbs on the stair steps. Two *ToF* sensors are also installed in front of the robot facing front riser. The range/distance measurements of these two *ToF* sensors are used to calculate the heading angle of the *sTetro* when it traverses left and right on the tread of a step.

The remaining article is organized as follows. In section 2, a brief introduction of the *sTetro* robot is given. We present 3D localization calculation method of the *sTetro* in section 3, and in section 4 the orientation estimation scheme is presented. Section 5 gives the experimental setup and results discussion. Section 6 concludes the work and proposes some future directions.

2. Introduction of *sTetro*: A Staircase Cleaning Robot

The *sTetro*, a staircase cleaning robot, is a reconfigurable robot which uses a vertical conveyor belt mechanism to climb the staircase. The working principle of *sTetro* is borrowed from the Tetris game [20]. The *sTetro* robot's body consists of three cuboids connected with two sliders attached to each side of the central cuboid. The hollow space inside each parallelepiped block encloses the electronic units, vacuum/suction tools and sweeping mechanism (planned). The blocks have a modular design, which allows reusing many pieces in the system. A GUI has been designed which allows the user to control the robot's motion, as desired. The reconfiguration mechanism allows the *sTetro* robot to efficiently traverse both the floor and stairs, which is almost impossible in presently available home cleaning robots in the market. The newer version of the *sTetro* has capabilities to autonomously navigation on the staircase without user intervention. Pictures of old and newer versions of *sTetro* robot are shown in Figure 1 bellow.



(**a**) (**b**)

Figure 1. (**a**) *sTetro (new ver.) robot* with sensors installed for this work, (**b**) *sTetro (old ver.)* robot's typical working environment: A staircase.

3. Absolute 3D Position of *sTetro*

In order to get absolute position information on all three axes (i.e. x-axis, y-axis, and z-axis), only the y-axis position is measured by *ToF* sensors in this work. The position along other two axes is calculated from the known geometry of the staircase. For instance, the width of a tread of stair step and height of riser is fixed on any staircase and can be measured manually. When the robot climbs the staircase, the distance traveled along x-axis and z-axis can be calculated if the step number is known on a staircase. The 3D absolute position of the *sTetro* is calculated as in Equation (1).

$$\begin{aligned} x_i &= t_l \times step_i - \left(\frac{t_l}{2}\right) \\ z_i &= r_l \times step_i \qquad ;\ i = 1, 2, \ldots.., N. \\ y_i &= \begin{cases} d_j & ; if\ \ sensor\ output < 120\text{cm} \\ W - d_j & ;\ \ otherwise \end{cases} \end{aligned} \tag{1}$$

Where t_l and r_l are lengths of tread and riser of a staircase, x_i and z_i are position along the x-axis and z-axis respectively, d_j is *ToF* sensor measurement along the y-axis, W is the width of staircase and i is the step number counter.

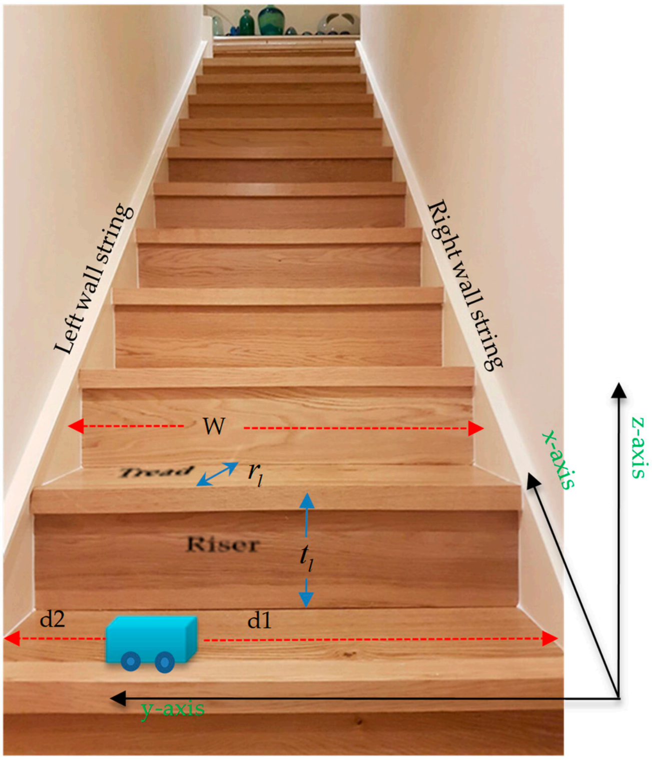


Figure 2. Staircase geometry to calculate 3D position and orientation of *sTetro* robot.

To calculate the robot position along the y-axis, the measurements from distance sensors installed at both sides of the robot are used. The reason that two *ToF* sensors are installed on left and right side of *sTetro* is that these sensors have limited raging capability ($\sim 120cm$), so when the robot is on extreme left or right side of the stair, the distance measurements of the sensor become unreliable. In this case, the measurements of the sensor on the other side of the robot are used to calculate position on the stair along y-axis. The complete pseudo-code for 3D position and orientation (heading) estimation is given in Algorithm I below.

Algorithm I: Pseudo code for 3D localization and heading estimation of *sTetro*.

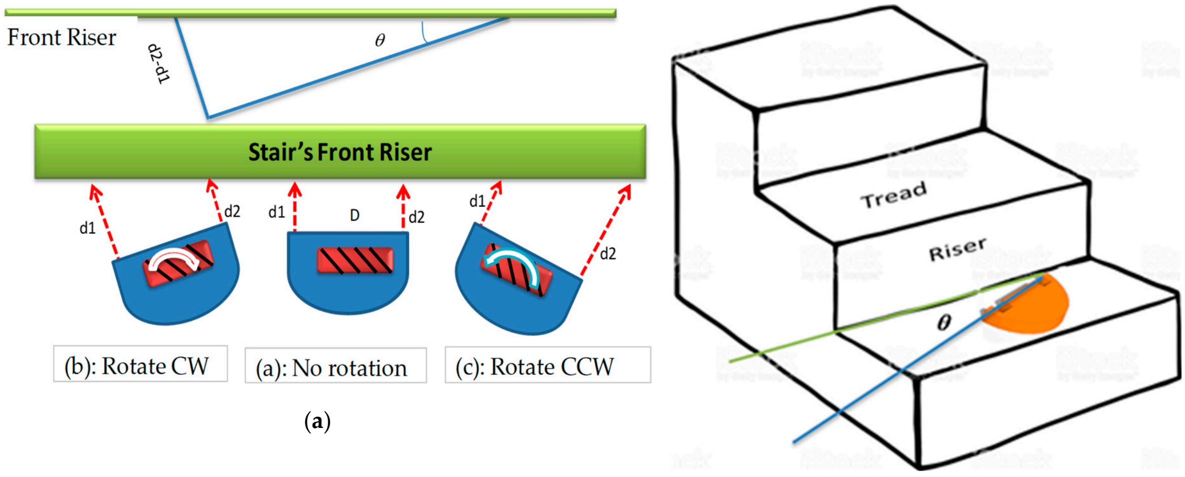
```
Initializ: $x_0 \leftarrow 0, y_0 \leftarrow Y_0, z_0 \leftarrow 0$
$i = 1$    //step number
  Loop : while $(i \le N)$
      $x_k \leftarrow t_l \times (i) - 0.5t_l$
      $z_k \leftarrow r_l \times (i)$
      if(sensor output <limitMax)   //limitMax = 120cm
          $y_k \leftarrow d_k$
      else
          $y_k \leftarrow W - d_k$
      $\theta_k \leftarrow \tan^{-1}\left(\frac{d_2 - d_1}{D}\right)$
      if$(\theta_k > 0)$
          rotateCW()
      if$(\theta_k < 0)$
          rotateCCW()
     $i \leftarrow readStep()$
  end loop
```

4. Orientation Calculation and Misalignment Correction

As the working environment of *sTetro* is staircase which has almost flat treads, hence the roll and pitch angle of *sTetro* are nearly zero in normal operation of the robot. Only heading angle (rotation about the z-axis) may change while traveling left/right on the treads. Two *ToF* sensors are installed at the front end of the *sTetro* robot to keep track of the heading angle. The role of these two sensors is to calculate the heading angle of the robot from the range measurements of the two sensors. Time of flight sensors give direct distance measurements with respect to the front riser of the step. The heading angle is calculated as follows in Equation (2),

$$\theta_k = \tan^{-1}\left(\frac{d_2 - d_1}{D}\right) \tag{2}$$

where d_1, d_2 are distance measurements from *ToF* sensors installed in front of the robot, and D is the width of the *sTetro* robot. If both distances are nearly equal, it leads to the conclusion that the robot is aligned with the front riser and heading angle is almost zero. If two ranges are not equal, a positive or negative heading angle indicates the misalignment of the robot with the front riser. Then the control system sends signals to actuators to turn *sTetro* clock-wise or anti-clock-wise in order to keep it aligned with the front riser for smooth motion. Heading angle estimation and correction scheme in this work is illustrated in Figure 3 below.



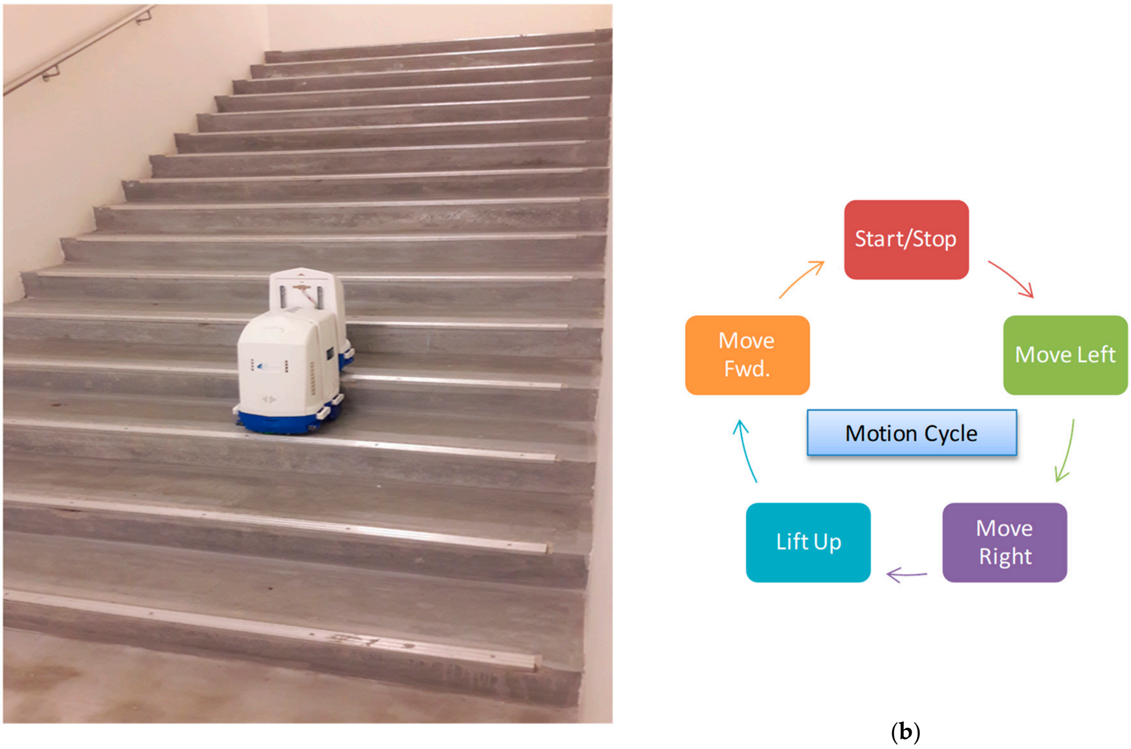
(a)

(b)

Figure 3. (a) Heading angle calculation from *ToF* sensor measurements, (b) Depiction of misalignment of *sTetro* on stair tread.

5. Experimental Setup and Results discussion

The *sTetro* robot's design and working principle are described in full detail in our previous work[19]. The *sTetro* robot is equipped with four additional *ToF* sensors for experiments conducted in this work. The distance sensor (*VL53L0X*) measures absolute range up to 2 meters [21]. An experimental setup for this work is shown in Figure4.



(b)

(a)

Figure 4. Experimental setup, (a) *sTetro* robot traversing the staircase, (b) One motion cycle of *sTetro*.

The *sTetro* robot is driven to traverse staircase as shown in Figure 4(a). The motion cycle for covering one stair step is given in Figure 4(b). Left and right wall strings restrict the motion of the robot. Two *ToF* sensors are installed on the left and right side of the central block and 2×*ToF* sensors are fitted in front of the first block of *sTetro*. The robot is started to move on the tread of the first step from right to left. After reaching extreme left of the tread, the left bumper touches the left wall string, and robot starts to move towards right on the tread. When the robot reaches the rightmost side, the right bumper touches the right wall string, and the robot starts to move towards left. After reaching midway on the tread, it stops and lifts central and last block to climb up to the next level. The counter in the software is incremented each time the robot climbs to the next step, thus counting the number of steps. This motion pattern repeats on the second step and so on, as before on the first step. During left/right motion, all four sensors measure the distances $(d_1 \sim d_4)$ which are used to calculate y-position and heading angle of the robot.

5.1. Heading Angle: Static test

In the first test, the two front sensor's range data (d_3, d_4) is used to calculate the heading angle of the robot. To assess the accuracy of the proposed method, a static test is performed first. The robot facing the front riser of the stair is rotated from +45° to -45° in continues rotation. Straight lines are marked with the help of angle protector at designated angles: 0°, -45°, +45°. Starting from facing the riser (i.e. heading angle ~0°), the robot is turned leftwards (CCW), thus increasing the distance (d_3) of the left sensor as compared to the distance (d_4) of right sensor. The heading angle starts increasing from 0° to +45°. At about +45°, the robot is held stationary for a while and then begins rotation rightwards (CW). Now the distance (d_4) starts decreasing, and at about zero degrees, both distances (d_3, d_4) are almost equal again, thereby measuring approximately 0° heading angle. After this, the robot is rotated CW, and now distance d_3 is creasing more than d_4, thereby increasing heading angle to -45°. At about -45°, the robot is held stationary for a while and then leftwards rotation takes place towards zero degrees, hence completing one rotation cycle. The results of this static experiment are shown in Figure 5, 6. The results of heading estimation using *ToF* sensors are compared with commercially available off-the-shelf *Attitude Heading Reference System (AHRS)*sensor from *VectorNav Inc*. This sensor has an accuracy of the absolute heading angle of ±2° RMS [22]. The zoom-in view of the heading angle estimation at different rotation states is given in Figure 6(a-d).

doi:10.20944/preprints201811.0295.v1

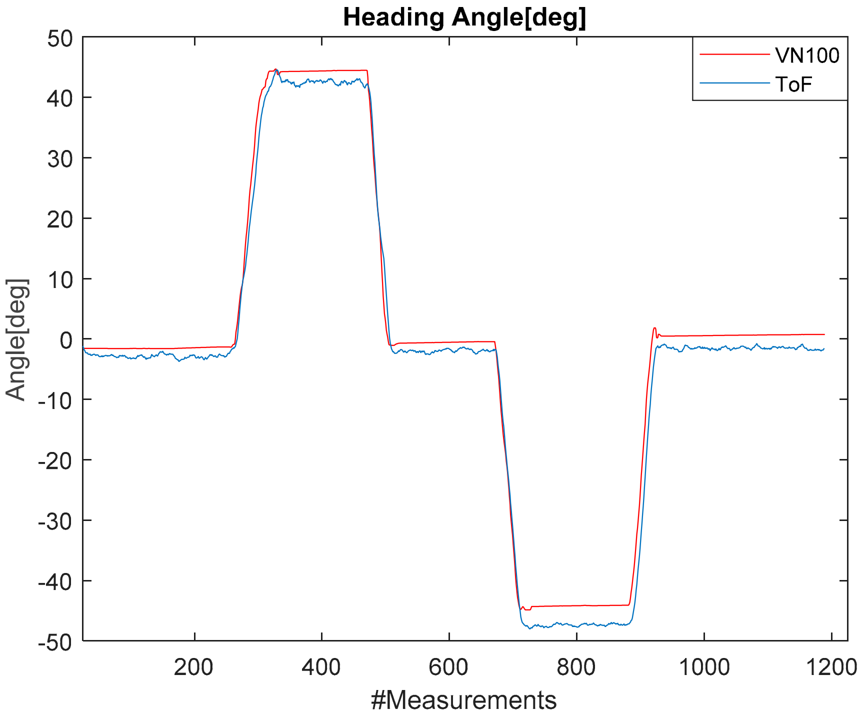
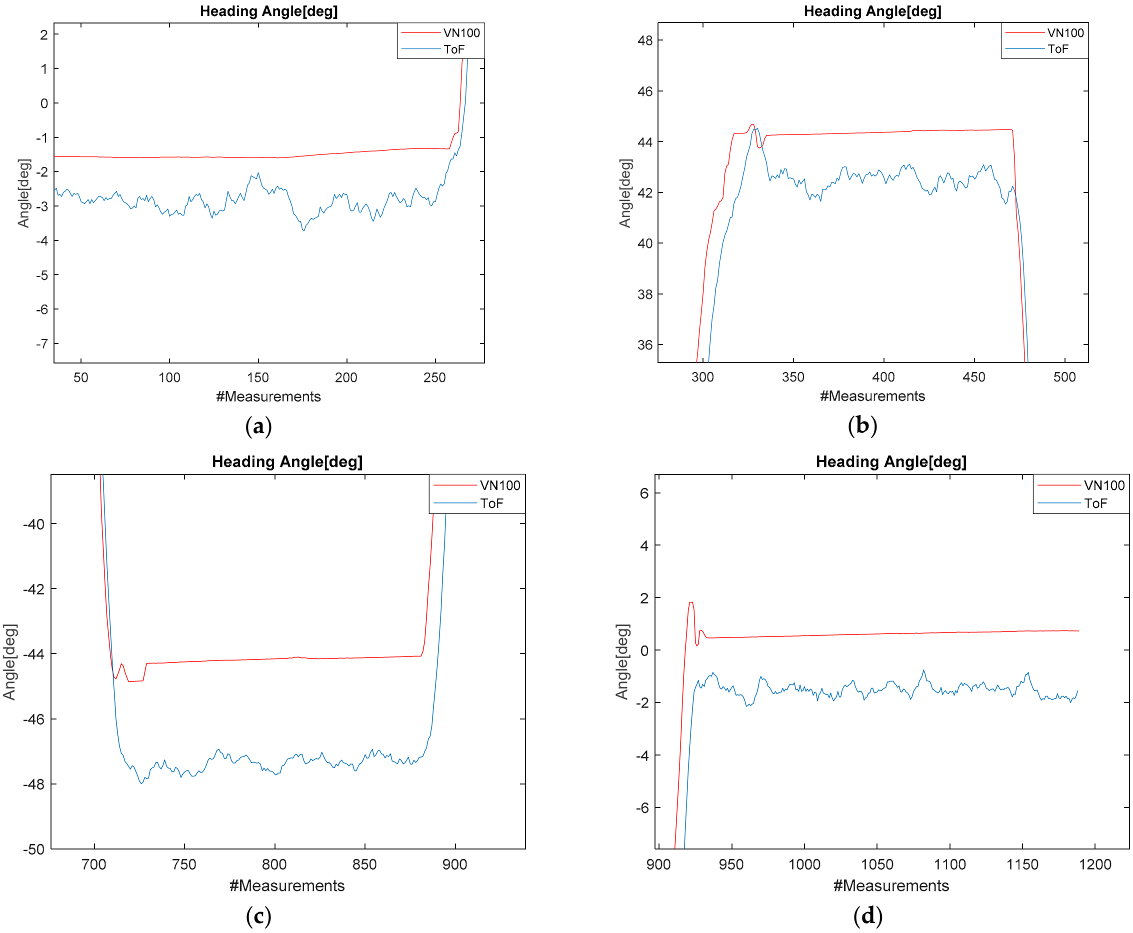


Figure 5. Comparison with VN100 AHRS: From 0° to +45°, back to 0° and then to -45°.



(**a**) (**b**)

(**c**) (**d**)

193 **Figure 6 (a-d).**Zoom-in view of heading angle aproximately at 0°, +45°, -45°.

194

The error in heading angle is calculated by comparing angle readings from reference *AHRS* and that of the proposed method and given in Table 1 below. It is observed that the proposed method of heading angle estimation is more accurate when rotation about the z-axis is minimal. At higher rotation angles (~ ±45°), the error in heading angle using *ToF* sensors is more. The reason is that at oblique angles, the error in range measurements obtained from *ToF* sensors increases, hence resulting in more error in heading calculation at large rotation angles about vertical.

Table 1. Accuracy of proposed scheme.

Rotation state	VN100 [deg]	*ToF* [deg]	Error [deg]
$\approx 0^{\circ}$	-1.80	-2.80	1.00
$\approx +45^{\circ}$	-44.2	-47.1	2.90
$\approx 0^{\circ}$	0.60	-1.40	0.80
$\approx -45^{\circ}$	44.40	42.60	1.80
$\approx 0^{\circ}$	-1.80	-2.80	1.00

5.2.3D Position and Orientation: Dynamic test

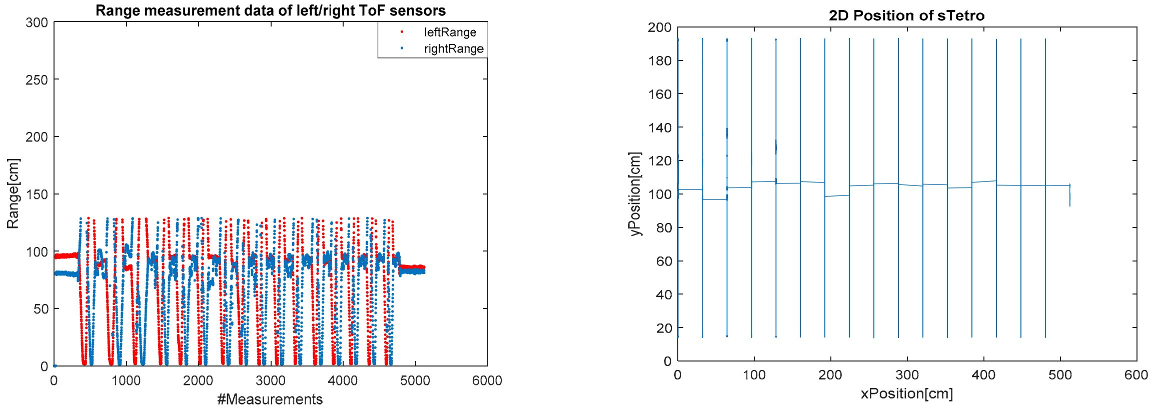
The 3D localization of *sTetro* is calculated with staircase geometry and measurements from onboard distance/range sensors only. The geometry of the staircase used and some parameters of *ToF* sensors and the *sTetro* robot are given in Table 2 below.

Table2. Staircase geometry and *sTetro's* main parameters

Parameters	Value(cm)	Remarks
Riser height(rl)	14	
Tread length(tl)	32	
Reliable range (max)	120	
sTetro size (l,w,h)	45×20×40	
No. of steps(N)	16	
Staircase size	480×205×165	(L×W×H)

It is observed that reliable ranging from left and right *ToF* sensors is only about 120 cm. Above this, the range data is not reliable for position calculation in this experiment, as confirmed from the range measurement data in Figure 7 (a). Starting from almost middle of the first step of staircase, the robot's initial position is set as $(x_0 = 0, y_0 = Y_0, z_0 = 0)$. The robot is then moved rightwards until its right bumpers touch right wall string of the staircase. The distance measurements from right side *ToF* sensor give y-position as the robot moves rightwards, keeping x-position and z-position constant $(i.e.\ x = \text{const.}, y = d_j, z = \text{const.})$ during rightwards motion. After touching the rightmost wall, the robot starts moving towards leftwards until the left bumper touches the left wall string of the staircase. After this, the robot comes back to the middle of the stair step. This completes the one traverse of *sTetro* on the stair step. Then the robot starts lifting up its 2nd block and then 3rd block to climb to the next level. During lifting up, its z-position changes while y-position and x-position remain constant $(i.e.\ x = \text{const.}, y = \text{const.}, z = z_j)$. After the lift-up action is complete, the robot moves forward to the middle of the tread to start next traverse on the stair step. During this forward motion, it's x-position changes while y-position and z-position remain constant $(i.e.\ x = x_j, y = \text{const.}, z = \text{const.})$. Note that during the left/right travel on the stair, if range measurement increases the maximum limit, then y-position is calculated using the other side's *ToF* sensor measurement, by subtracting its range measurement from total width (W) of the staircase.

The raw range measurements of the left and right *ToF* sensors are given in Figure 7 (a). These range measurands become unusable after about 120 cm. Measurements from both the left and right range sensor are used to calculate y-position of the robot. The 2D position (x,y) of *sTetro* is shown in Figure 7 (b). The robot travels about 480 cm in the x-axis and 180 cm in the y-axis (excluding robot width from y-axis distance calculation). The 3D position plot is given in Figure 8. It is evident that all axis position data is very smooth and highly accurate when compared with actual geometry(dimensions) of the staircase. In z-axis, the robot covers approximately 165 cm, as shown in this figure. The heading angle estimation is shown in Figure 9. It is seen that that absolute orientation remains within ±1° while traversing the entire staircase. The standard deviation calculated of the heading error is approximately 0.9 degree. The misalignment correction was not implemented in this experiment.



(**a**) (**b**)

Figure 7. (a) Raw range measurements of left and right distance sensor. (b) 2D position (x,y) of *sTetro*.

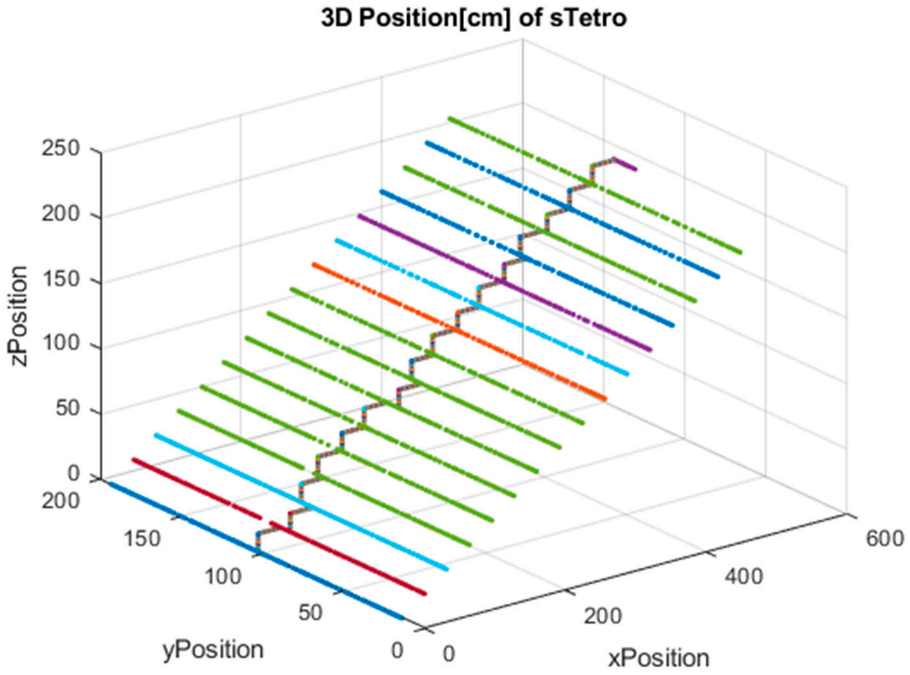


Figure 8. 3D absolute position (x,y,z) plot of *sTetro* on 16-step staircase.

5.3. Error analysis

Error in heading angle is due to the error in range measurements of the sensor itself. It is noted that range error is about ±3cm which contribute to an error in heading angle about ±1°. We compared heading angle accuracy with off-the-shelf *AHRS* sensor (VN-100) from *VectorNav Inc.* in experiments conducted in this work. The absolute heading angle estimation with the proposed scheme and its error w.r.t reference *AHRS* sensor is given in Figure 9 below.

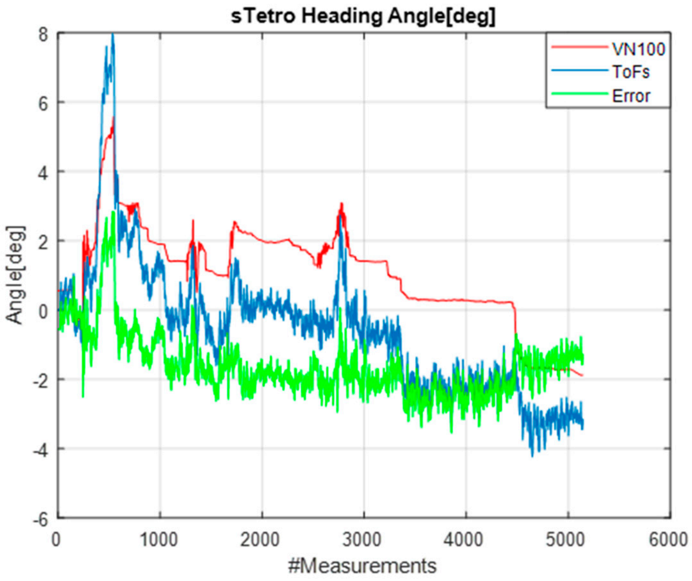


Figure 9. Absolute heading angle estimation of *sTetro.*

The error in y-position is calculated as in Equation (3). Here we calculated the error in total distance estimated along the y-axis as compared to width of the staircase for every step. The results are given in Table 3 below.

$$Error = W - [(y_{\max} - y_{\min}) + R_W] \tag{3}$$

Here W is the width of the stair step, R_W is the width of robot and $y_{\min}, y_{\max}$ are the minimum and maximum range measurements of left/right *ToF* sensors on a particular step. On each step, the difference between the maximum and minimum range measurements (by adding robot width as well) gives distance covered on a tread. The error is calculated by subtracting this difference from the manually measured width of the stair.

Table 3. Accuracy of proposed scheme.

Staircase Step#	$y_{\max}(cm)$	$y_{\min}(cm)$	$Error[cm]$	Heading angle error
1	193.1	14.0	2.9	
2	193.0	14.1	3.1	
3	193.2	14.3	3.1	Mean=0.99°
4	193.1	14.1	3.0	StdDev.=0.93°
5	192.9	14.2	3.3	
6	193.1	14.3	3.2	
7	193.0	14.3	3.3	
8	193.1	13.9	2.8	
9	193.1	14.2	3.1	
10	193.2	14.1	3.1	
11	193.3	14.3	3.0	
12	193.1	14.2	3.1	
13	192.9	14.0	3.1	
14	193.1	14.3	3.2	
15	193.2	14.3	3.1	
16	193.0	14.0	3.0	
			$Mean = \frac{1}{N}\sum_{i=1}^{N} Error_i = 3.1\text{cm}$	
			$stdDev(Error_i) = 0.13\text{cm}$	

6. Conclusion

In this work, a novel and a straightforward method is presented to calculate 3D absolute position and orientation using staircase geometry and minimal onboard sensors only. As the stair cleaning robot is assumed to travel in a known environment, so geometry of the staircase is exploited to calculate its position and orientation. The proposed method is very accurate as manual measurements of stair's riser and tread are provided to the position calculation algorithm. The only uncertainty arrives in y-axis because of the range sensor's random noise. This noise is minimized by taking the moving average of the measurements of *ToF* sensors. The error in y-axis only arrives due to the error in sensor measurement itself (~3 cm as per datasheet). Mean error in position in 3.1 cm with a standard deviation of 0.13 cm only. The mean error in heading angle is 0.99 degree with a standard deviation of 0.93 degree. Ideally, the heading angle of *sTetro* robot should remain within ±5 degrees in order to keep aligned with the front step riser. The proposed method faithfully fulfills this requirement. As a future work, this method may be integrated with other indoor positioning systems for overall system redundancy and reliability enhancement.

Author Contributions: Muhammad Ilyas conceived the main idea presented in this article and wrote most of the part of this paper.**Vinu Sivanantham and Manojkumar Devarassu** helped in setting up experiments and provided some of the figures. **Balakrishnan Ramalingam** helped to conduct experiments and result analysis. This worked in done under the supervision of A/Prof. **Mohan Rajesh Elara**.

Conflicts of Interest: The authors declare that there is no conflict of interests regarding the publication of this paper.

Funding: This work is financially supported by the National Robotics R&D Program Office, Singapore, under the Grant No. RGAST1702.

References

[1] M. Ilyas, K. Cho, S. H. Baeg, and S. Park, “Drift reduction in IMU-only pedestrian navigation system in unstructured environment,” *2015 10th Asian Control Conf. Emerg. Control Tech. a Sustain. World, ASCC 2015*, 2015.

[2] M. B. K. W. George, “A Navigation Control System for an Autonomous Mobile Robot,” in *Robotics and Factories of the Future ’87*, Springer, Berlin, Heidelber, 1988, pp. 678–686.

[3] F. A. X. Da Mota, M. X. Rocha, J. J. P. C. Rodrigues, V. H. C. De Albuquerque, and A. R. De Alexandria, “Localization and navigation for autonomous mobile robots using petri nets in indoor environments,” *IEEE Access*, vol. 6, pp. 31665–31676, 2018.

[4] T. S. Wolfram Burgard, Dieter Fox, Daniel Hennig, “Estimating the absolute position of a mobile robot using position probability grids,” *Proc. Thirteen. Natl. Conf. Artif. Intell.*, no. March, pp. 1–15, 1996.

[5] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme, “Robust localization using relative and absolute position estimates,” *Proc. 1999 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 2, pp. 1134–1140, 1999.

[6] E. M. Gorostiza, J. L. L. Galilea, F. J. M. Meca, D. S. Monzú, F. E. Zapata, and L. P. Puerto, “Infrared sensor system for mobile-robot positioning in intelligent spaces,” *Sensors*, vol. 11, no. 5, pp. 5416–5438, 2011.

[7] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, “Mobile robot positioning: Sensors and techniques,” *J. Robot. Syst.*, vol. 14, no. 4, pp. 231–249, 1997.

[8] M. B. Alatise and G. P. Hancke, “Pose estimation of a mobile robot based on fusion of IMU data and vision data using an extended kalman filter,” *Sensors (Switzerland)*, vol. 17, no. 10, 2017.

[9] B.-S. Cho, W. Moon, W.-J. Seo, and K.-R. Baek, “A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding,” *J. Mech. Sci. Technol.*,

doi:10.20944/preprints201811.0295.v1

vol. 25, no. 11, pp. 2907–2917, 2011.

[10] E. North, J. Georgy, M. Tarbouchi, U. Iqbal, and A. Noureldin, "Enhanced mobile robot outdoor localization using INS/GPS integration," *Proc. - 2009 Int. Conf. Comput. Eng. Syst. ICCES'09*, no. December 2013, pp. 127–132, 2009.

[11] M. Hammouche, S. Sakhi, M. Belhocine, A. Elourdi, and S. Bouaziz, "A Fuzzy Controller for GPS/INS/Odm Integrated Navigation System," *Proc. 13th Int. Conf. Informatics Control. Autom. Robot.*, no. September, pp. 390–397, 2016.

[12] J. Biswas and M. Veloso, "WiFi localization and navigation for autonomous indoor mobile robots," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4379–4384.

[13] J. Lim, S. Lee, G. Tewolde, and J. Kwon, "Indoor localization and navigation for a mobile robot equipped with rotating ultrasonic sensors using a smartphone as the robot's brain," *IEEE Int. Conf. Electro Inf. Technol.*, vol. 2015–June, pp. 621–625, 2015.

[14] D. Xu, L. Han, M. Tan, and Y. F. Li, "Ceiling-based visual positioning for an indoor mobile robot with monocular vision," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1617–1628, 2009.

[15] T. J. Lee, C. H. Kim, and D. I. D. Cho, "A Monocular Vision Sensor-Based Efficient SLAM Method for Indoor Service Robots," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 318–328, 2019.

[16] S. Park and S. Hashimoto, "Autonomous navigation system for mobile robot using randomly distributed passive RFID tags," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E93–A, no. 4, pp. 711–719, 2010.

[17] C. Falsi, D. Dardari, L. Mucchi, and M. Z. Win, "Time of arrival estimation for UWB localizers in realistic environments," *EURASIP J. Appl. Signal Processing*, vol. 2006, pp. 1–13, 2006.

[18] A. Tedeschi, S. Calcaterra, and F. Benedetto, "Ultrasonic RAdar system (URAS): Arduino and virtual reality for a light-free mapping of indoor environments," *IEEE Sens. J.*, vol. 17, no. 14, pp. 4595–4604, 2017.

[19] M. Ilyas, S. Yuyao, R. E. Mohan, M. Devarassu, and M. Kalimuthu, "Design of sTetro : A Modular , Reconfigurable , and Autonomous Staircase Cleaning Robot," vol. 2018, 2018.

[20] "Tetris game," *https://en.wikipedia.org/*. ((accesses on 2 Feb. 2018).

[21] "Vl53L0X", https://www.pololu.com/product/2490, (accesses on 10 April. 2018).

[22] "VN100 MEMS IMU", https://www.vectornav.com/products/vn-100, (accesses on 8 Oct, 2018).