

Original Research Paper

Vision-Based Real-Time Positioning and Autonomous Navigation System Using Mobile Robot in Indoor Environments

¹Diop Mamadou, ¹Lim Chot Hun, ¹Lim Tien Sze and ²Ong Lee Yeng

¹Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia

²Faculty of Information Science and Technology, Multimedia University, Melaka, Malaysia

Article history

Received: 14-03-2016

Revised: 10-05-2016

Accepted: 17-05-2016

Corresponding Author:

Diop Mamadou

Faculty of Engineering and

Technology, Multimedia

University, Melaka, Malaysia

Email: mdiop86@gmail.com

Abstract: Research toward unmanned mobile robot navigation has gained significant importance in the last decade due to its potential applications in the location-based services industry. The increase in construction of large space indoor buildings has made difficulty for humans to operate within such environments. In this study, a mobile robot's indoor navigation algorithm is developed with vision cameras. Using two monocular cameras (one looking forward and one looking downward), the developed algorithms make use of the salient features of the environments to estimate rotational and translational motions for real-time positioning of the mobile robot. At the same time, an algorithm based on artificial landmark recognition is developed. The artificial landmark is shaped arrow based signboards with different colors representing different paths. These algorithms are integrated into a designed framework for mobile robot real-time positioning and autonomous navigation. Experiments are performed to validate the designed system using the mobile robot PIONEER P3-AT. The developed algorithm was able to detect and extract artificial landmark information up to 3 m distance for the mobile robot guidance. Experiment results show an average error of 0.167 m deviation from the ideal path, signified the good ability and performance of the development autonomous navigation algorithm.

Keywords: Real-Time, Vision-Aided Navigation, Artificial Landmark

Introduction

Autonomous navigation for a ground-based mobile robot has become more and more desirable in these recent years in both indoor and outdoor environments. The usage of mobile robots in indoor environments (such as offices, warehouses, airports, etc.) appears to be more challenging since most the available technologies for positioning failed to operate reliably and accurately in indoors environments (Rivera-Rubio *et al.*, 2015). While Global Positioning System (GPS) is unavailable in indoor environments, the low-cost Micro-Electro-Mechanical System (MEMS) suffered from various stochastic errors. At the same radio frequency signals such as Radio-Frequency Identification (RFID) and Wireless Fidelity (WiFi) require dedicated and cost effective infrastructures (Zhao *et al.*, 2007; Atia *et al.*, 2015; Zhuang *et al.*, 2015).

For these reasons, the implementation of visual sensors (i.e., cameras) in mobile robot navigation applications are actively being studied (Yang *et al.*, 2012). For a mobile robot real-time positioning using vision, most of the proposed solutions are able to provide good accuracy in terms of measurement and robustness. However, at some points, these solutions have limitations. While monocular vision failed to operate in complete unknown environment (Zhang *et al.*, 2014), stereovision tends to be heavier in computation with limited range (Huang, 2013; Hong *et al.*, 2012) and vision aided with inertial sensor is the most costly configuration with delays (Hesch *et al.*, 2013). Despite these limitations, monocular vision appears to be the most suitable candidate for a vision-based navigation solution since it is able to provide richness of information for a high level of intelligence with a lower cost sensor (Ye *et al.*, 2012). Compared to other depth

sensing devices, the range of a vision perception is unlimited. Therefore, monocular vision is able to detect features at its sight, no matter how close or far features are located, allowing it to operate in small and large, indoor and outdoor environments (Engel *et al.*, 2012).

For a vision based mobile robot autonomous operations, various sensors are used to get information, in its surroundings, that could help it take appropriate actions to reach its destination (Bhattacharayya *et al.*, 2014). Many practical approaches rely on observations of artificial landmarks placed in strategic places within the environments. For examples, Ye *et al.* (2012) use line marks on the floor combined with drawn MR Codes, Li *et al.* (2012) uses line marks considering their poses and Ortega-Garcia *et al.* (2014) uses straight lines combined with angles and landmarks recognition on its environment. This method is simple and widely used. However, difficulties arise when several paths to different destinations are involved; whether intersecting or sharing same corridors at few occasions (Ye *et al.*, 2012). In these situations, it is difficult to represent all paths as a single line mark model that allows the mobile robot to navigate safely at the middle of these corridors (avoiding possible misdirection and crashes). Moreover, tapes line marks method is not easy to modify after being set. Line marks following technique is usually more convenient with smooth curves rotations, which is not practical in big rotation angles (Ortega-Garcia *et al.*, 2014). In more recent years, teach-and-replay methods have been studied and applied for autonomous navigation. For examples, Rojas Castro *et al.* (2015) developed an autonomous indoor navigation system based on the both prior analysis of a paper based floor plan of a building for reference and neuron-shaped artificial landmarks that help the robot on navigation sequences. Combining both teach-and-replay feature-based method and a segmentation-based approach, De Cristóforis *et al.* (2015) developed an autonomous navigation method, as an improved version of Chen and Birchfield (2009) method, capable of operating in both indoor and outdoor environment. Following the same logic, teach and replay based technique aided by local ground planarity is used by Clement *et al.* (2015) for an autonomous navigation system experimented in both indoor and outdoor environments. However, as mentioned by De Cristóforis *et al.* (2015), the main drawback of the teach and replay methods lies in the fact that the robot workspace is limited only to the regions mapped during the training step. The robot has to be guided throughout the entire path before performing autonomous navigation, which may represent a very tedious process.

This paper proposed an alternative solution for vision-based navigation in indoor environments using two cameras arranged in a unique way. With no prior learning nor learning of a structured and non-structured environment, this system of integrated algorithms is capable of providing an accurate real-time positioning by exploiting natural features of the environment, as well as directives using arrow based signboards as artificial landmark placed within the environment for an autonomous navigation. Several experiments are performed to highlight the system performance in terms of accuracy and robustness.

The organization of this paper is as follow: The techniques applied for the mobile robot positioning are presented in section 2, while the artificial landmark driven autonomous navigation methodology is elaborated in section 3. The system design and its functionalities are described in section 4 and the experimental studies and results are discussed in section 5. Lastly, the paper is concluded in section 6.

Positioning Algorithms Description

The developed positioning algorithms utilized two cameras, one looking forward and one looking downward, to detect and track salient features in the environment. Whenever a motion occurs, features patterns from successive image frames are exploited to estimate travelled distance and rotations motions. The algorithms are divided in two positions, where the forward-looking camera is used to estimate rotation motions (Diop *et al.*, 2015), while the downward-looking camera is utilized for travelled distance estimation (Diop *et al.*, 2014).

Using a forward-looking camera for rotations motions estimation, the magnitude A from a displacement of a feature point Γ between the previous image (pi) and the current image (ci) can be obtained from the coordinate points $\Gamma_{ci}(x, y)$ and $\Gamma_{pi}(x, y)$, as shown in Fig. 1, as follow:

$$A_{(x,y)} = \Gamma_{ci(x,y)} - \Gamma_{pi(x,y)} \quad (1)$$

Assuming that the *Width* of the image in pixel W is equivalent to the camera field of view following the x-axis in degree (FOV_x), the equation to calculate the yaw θ_i rotation angles between images is expressed as follow:

$$\theta_i = \frac{A_x \times \varphi_{FOV_x}}{W} \text{ and } \theta = \sum_{i=1}^n \theta_i \quad (2)$$

where, n is the number of pairs of images.

On the other hand, the downward-looking camera is used to estimate the travelled distance from ground features. As shown in Fig. 2, the travelled distance

estimation required information of the $\phi_{FOV,y}$, the height of the image resolution H (in pixels), the height of the camera Z with respect to the ground (in centimeter) and the camera focal length F_y following the y-axis (in pixels). From the $\phi_{FOV,y}$, the focal Length F_y is formulated as follow:

$$F_y = \frac{H \times \frac{1}{2}}{\tan\left(FOV_y \times \frac{\pi}{180} \times \frac{1}{2}\right)} \quad F_y = \frac{0.5 \times H}{\tan\left(\frac{\pi}{360} \times \phi_{FOV,y}\right)} \quad (3)$$

Given pairs of identical features (Γ_p, Γ_c) between pi and ci of ground, the magnitude A_y (in Pixel) of the features' displacement can be computed using Equation 2. By correlated with the magnitude pairs of features from the image, the travelled distance d_{TD} of a mobile robot can be estimated between successive images, using following equation:

$$d_{TD} = \frac{2 \times Z \times A_y \times \tan\left(\frac{\pi}{360} \times \phi_{FOV,y}\right)}{H} \quad (4)$$

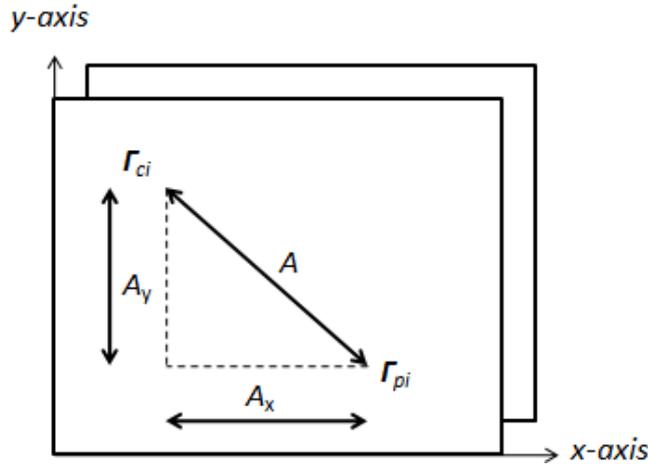


Fig. 1. Geometrical representation of features displacements on image frame of 3-DOF orientation estimation

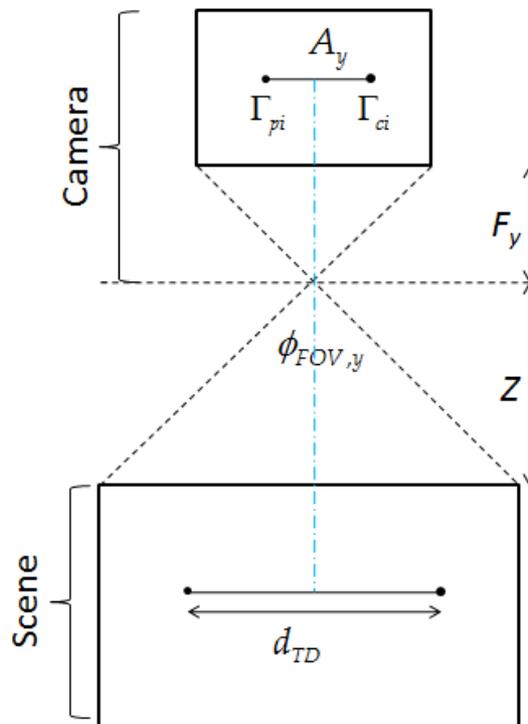


Fig. 2. Geometrical representation of features displacements on image frame for travelled distance estimation

Artificial Landmark Driven Autonomous Navigation Algorithm

The autonomous navigation algorithm recognized colored arrow on sign boards as artificial landmarks, where artificial landmarks are specially designed artificial landmark that aid the autonomous navigation. From the recognized landmark, the algorithm extracted necessary information (such as color, pointing direction, pose and depth) that allowed a mobile robot to guide itself through a specific path toward a destination.

In between successive captured frames, the algorithm first tries to detect the presence of the artificial landmark shapes within the frame scene by matching all identified contours with a predefined template. The template matching process consisted of matching all contours with an arrow shape template using HuMoments technique (Hu, 1962; Mercimek *et al.*, 2005; Bradski and Kaehler, 2008). Once the shapes are detected, the colour of the artificial landmarks will be identified. The proposed algorithm is operated in such a way that different predefined colors represented different path to a destination. Each Artificial landmark's color is verified, whether it corresponds to the preselected one to follow or not. Colored image can be represented in several color models among such Red blue Green (RGB) and Hue Saturation Value (HSV) (Ibraheem *et al.*, 2012). As an approximation way of humans' perception and interpretation of colors, HSV offered robustness in detecting lighting changes and shadows with the ability of separating image intensity from color information. HSV used the combination of three components (Hue H , Saturation S , Value V) to define a specific color. Each color brightness, from the lightest to the darkest, is defined within the HSV combination range from its minimum ($H_{min}, S_{min}, V_{min}$) o its maximum ($H_{max}, S_{max}, V_{max}$) value. Since specific path are represented with specific colour in the proposed algorithm, these colours are calibrated within the specific range of ($(H_{min}, S_{min}, V_{min}), (H_{max}, S_{max}, V_{max})$).

Once the preselected colour is detected, information such as the depth, the orientation to the centre of the frame scene, the pointing direction and the distance to travel are extracted from the artificial landmark in order to provide directive to reach the desired destination.

Arrow Tip Detection

The arrow shape used in this study is a heptagonal arrow shape with seven corner nodes. Each of these corner nodes formed an angle from the intersection of any two successive vertices. As shown in Fig. 3, the

tip of the arrow N_i can be identified by acquiring the arrow tip TIP through comparing the sum of all three successive corner node angles, and find the smallest summation angle. Thus, the tip of the arrow N_i can be identified through Equation 5 to 7, as shown below:

$$TIP = N_i(\widehat{N}_i) \quad (5)$$

where:

$$\widehat{N}_i = (N_{(i-1+7) \bmod 7}, \widehat{N}_i, N_{(i+1+7) \bmod 7}) \quad (6)$$

and:

$$\begin{aligned} & \sum (N_{(i-1+7) \bmod 7}, \widehat{N}_i, N_{(i+1+7) \bmod 7}) \\ & = \min \left(\sum_{x=0}^6 (N_{(x-1+7) \bmod 7}, \widehat{N}_x, N_{(x+1+7) \bmod 7}) \right) \end{aligned} \quad (7)$$

Arrow Depth Information

The Depth information D of a detected arrow is defined as the estimated distance between the front-looking camera and the detected arrow in the image frame. The derivation of depth data required information such as the $\phi_{FOV,x}$, the height of the image resolution H , the camera focal length F_x in pixel, the real height RH of the arrow in centimeter and the image height IH of the arrow in pixel, as shown in Fig. 4.

From the $\phi_{FOV,x}$ and H , the focal Length F_x is formulated in Equation 8, as shown below:

$$F_x = \frac{0.5 \times H}{\tan \left(\phi_{FOV,x} \times \frac{\pi}{360} \right)} \quad (8)$$

By referring to Fig. 4, IH corresponded to the line from the tip of arrow P_1 to the center of the line formed by the edges P_4 and P_5 . Therefore, the depth information D can be derived as follows:

$$D = \frac{F_x \times RH}{IH} \quad (9)$$

Arrow Orientation Information

The orientation of the arrow shape (ω_i) can be illustrated as the arrow's position towards the x-axis, from the central line of the image frame. Fig. 5 showed an example of the orientation representation of the arrow shape in 2D postures. Consider an image

frame with resolution of H (Height) and W (Width) in pixel, it is possible to compute the central line of the image frame through the coordinates $(H/2, 0)$ and $(H/2, W)$. Given the centre of an arrow $C_i(x, y)$ recognized in the image, the line that is parallel to the x -axis and intersects the central line of the image in $P_i(x, y)$ represented the arrow position expressed in pixel from the image centre; as shown in Equation 10:

$$\|C_i P_i\| = C_{i,x} - P_{i,x} \quad (10)$$

Given $\phi_{FOV,x}$ and H , the arrow orientation expressed in degree is formulated and shown in Equation 11:

$$\omega_i = \frac{(\|C_i P_i\| * \phi_{FOV,x})}{H} \quad (11)$$

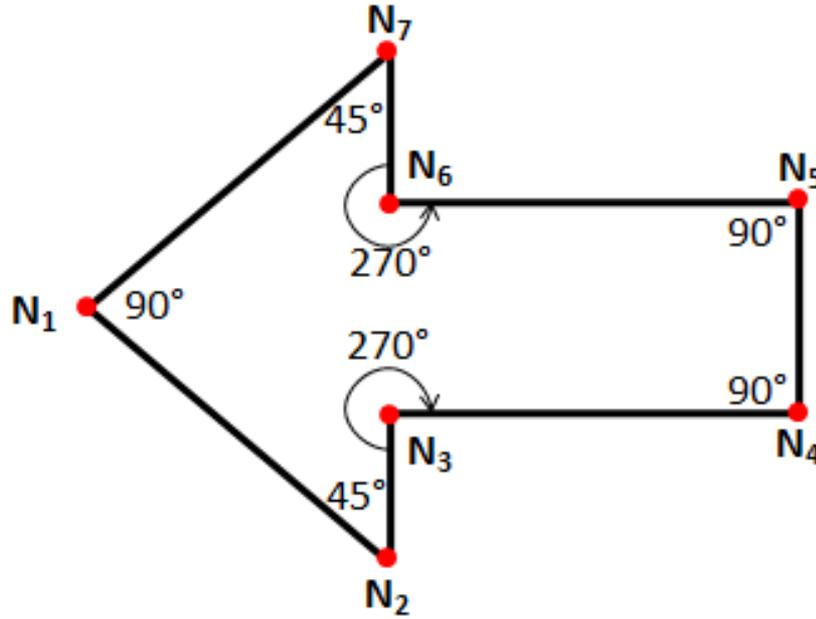


Fig. 3. Geometrical representation of the artificial landmark based arrow for the mobile robot autonomous navigation

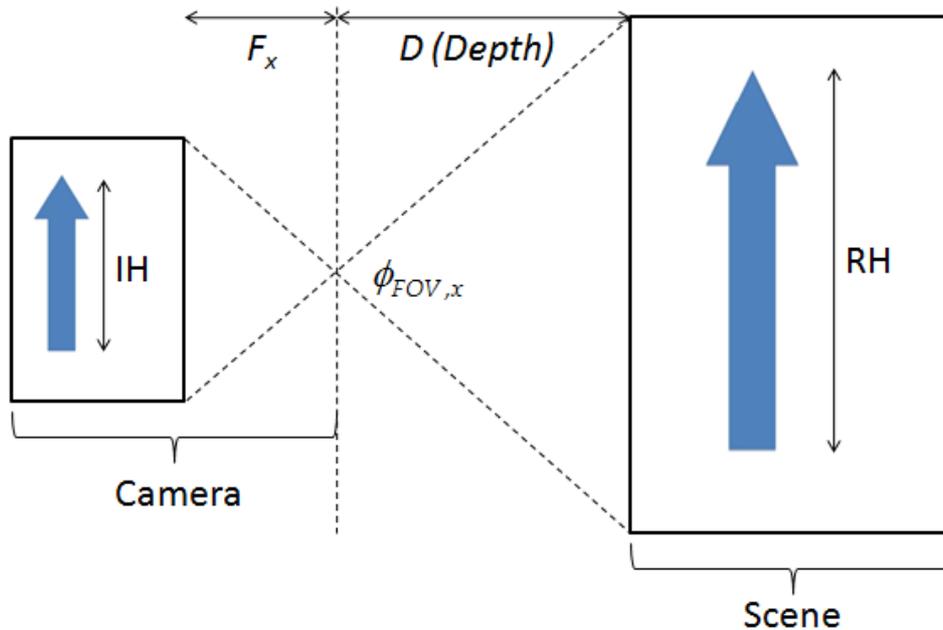


Fig. 4. Geometrical representation of the arrow depth

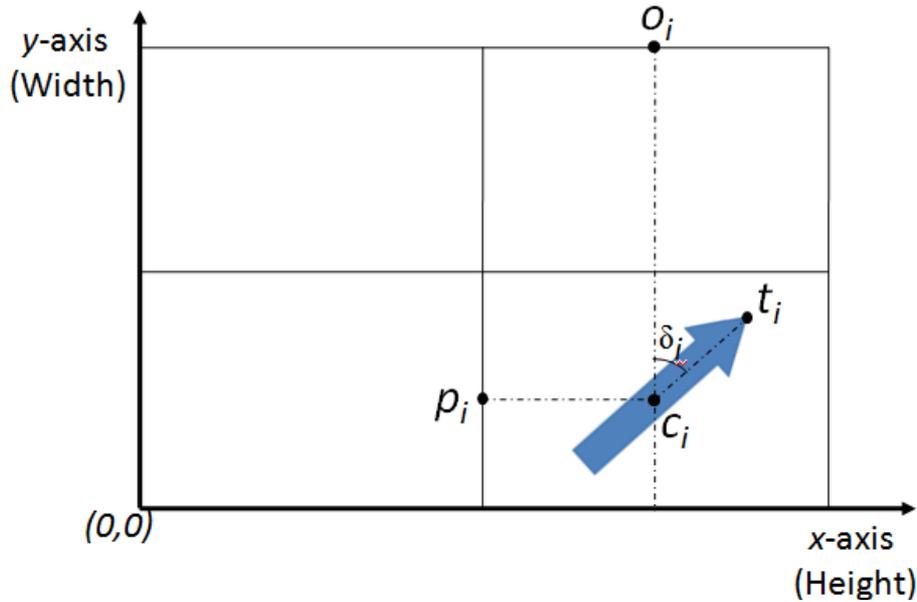


Fig. 5. Geometrical representation of the arrow Orientation in an image frame

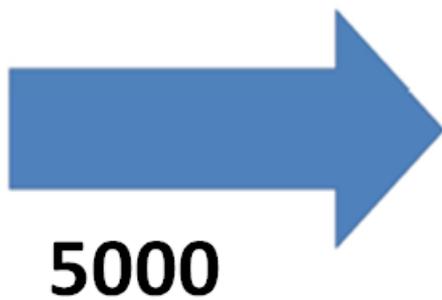


Fig. 6. Extraction of an arrow based signboard with distance to travel information

Arrow Pointing Direction Information

The pointing direction of the artificial landmark arrow conveyed information on how many degrees, in yaw angle, a mobile robot were to turn. The pointing angle δ_i is formed between the point o_i , the point C_i and the tip t_i of that arrow, as illustrated in Fig. 5. Note that δ_i can be obtained using Equation 12:

$$\delta_i = \angle o_i, \widehat{C_i}, t_i \quad (12)$$

Distance to Travel Information

The distance to travel information provides input on how far a mobile robot needed to travel in order to reach the next check point. This information is the result of extraction and recognition of the numerical characters, appeared next to the arrow sign, as shown in Fig. 6. Extraction of the numerical characters

required prior detection of the colored arrow in order to crop the signboard area. This function extended the height and width delimitation of the arrow. Here, the centre of the arrow is used as the starting point of all dimension sides for border extension.

Once the sign-board is extracted, the numerical characters information can be identified by isolating its written color with the color detection technique described in previously. Next, morphological image processing techniques are applied to gather all blobs, near to each other, into in a single larger blob representing the value area. From that perspective, smaller and larger blobs excluded within a range are considered as noises and are removed from the image. That allowed the numerical characters area appeared as the only candidate within a range nearest to arrow centre, to be extracted and processed for optical character recognition. Next, segmentation and recognition using Optical Character Recognition (OCR) Technique is performed. The segmentation of the numerical characters involved the same procedure of the contours detection techniques described in previously, except here all contours are taken into consideration since each one represents a number. Each of the segmented images is processed using OCR technique (Smith, 2007) to convert the image contents into characters. Figure 7 illustrates the overall conversion process to obtain the travelled distance information from the numerical characters.

Pose Correction Handling

Assuming the mobile robot is initially aligned perpendicularly to the artificial landmark, then the

only reason an arrow encounter a yaw orientation pose is because the mobile robot is heading through its centre at a certain angle during navigation. If the mobile robot approached an arrow with a certain angle, it means that few maneuvers (yaw rotations) were performed throughout the path between one arrow signboard to another, as shown in Fig. 8. The correction value θ_c , which represented the right angle to rotate, can be obtained by subtracting the pointing direction information α , with the sum of all rotation angles θ_i throughout the way; as expressed in Equation 14:

$$\theta_c = \delta - \sum_{i=0}^n \theta_i \quad (14)$$

where i is the number of rotations and n is the total number of rotations.

Algorithm Validation

Three experiments were setup to validate the proposed algorithms. The first experiment aimed to test the algorithm robustness and accuracy. A set of 18 images, each contained an arrow of 22 cm height within the signboard, are captured. The signboard is positioned at two different depths (up to 3 m) with three different poses (yaw, pitch and roll orientation), as illustrated in Fig. 9. The arrow information extracted from the image are the matching value, the depth, the pointing direction, the position to the centre of the image, the size area in the image, the length and all corner point angles as structured in Table 1.

In this experiment, the information is analyzed on different signboard posture in order to define the algorithm constraint settings for robustness. The results obtained from the experiment are illustrated in Fig. 9 and Table 1. Results demonstrated the algorithm accuracy and robustness in detecting and extracting arrow information over a distance up to 3 meters. The accuracy of the algorithm is evaluated by using the up, down, left, right directions as a reference for ground truth measurements, representing 0, 180, -90 and 90°C respectively. Data collected from Table 1 on Images 1, 7, 9, 11, 13, 16-18, the pointing direction (direction in Table 1) measurements generates an average error of 0.6°C, with the highest errors occurring on farthest detections. However, the accuracy of the depth information is highly depends on the arrow poses in the scene, where yaw-oriented poses have an effect on the arrow length which is used for depth measurement.

The robustness of the evaluated by its ability to detect the arrow up to 3 m distance and at any poses (yaw, pitch, roll orient) up to some extent. Several extreme case were set in the experiment, involving the detection of the arrow when its pose is yaw-oriented up to $\pm 30^\circ$ (as shown in Image 2 and 3 in Fig. 9), pitch-oriented up to $\pm 30^\circ$ (as shown in Image 4 and 5 in Fig. 9) and roll-oriented up to $\pm 360^\circ$ (as shown in Image 6, 8, 9, 10, 11 and 12 in Fig. 9). In addition, the algorithm robustness is illustrated in the accurate recognition of the arrow tips. The findings in this experiment have proven that on arrow angles data collected in Table 1, where the application of Equation 7 on detected arrow always recognizes the tip.

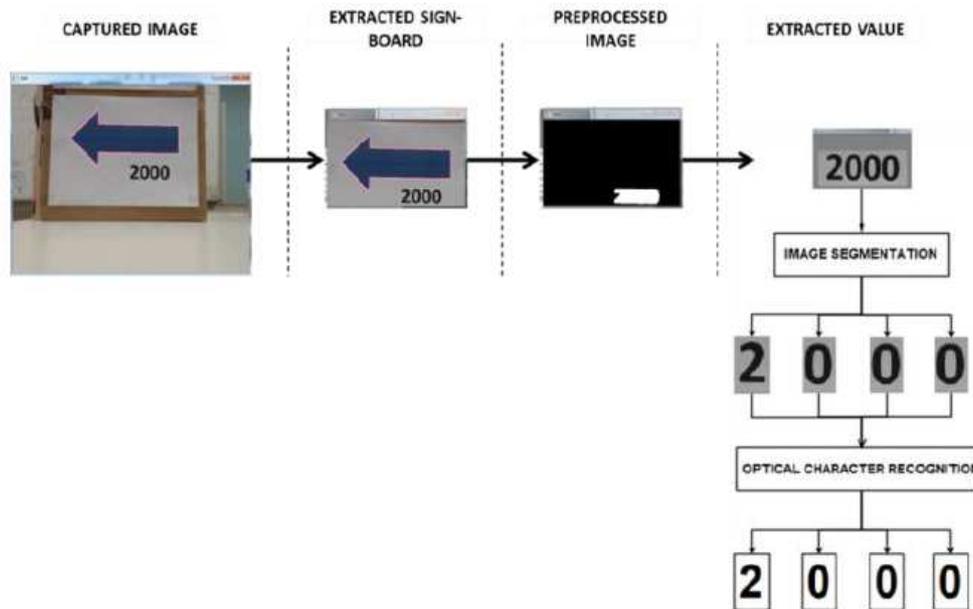


Fig. 7. Process of distance to travel information extraction and recognition

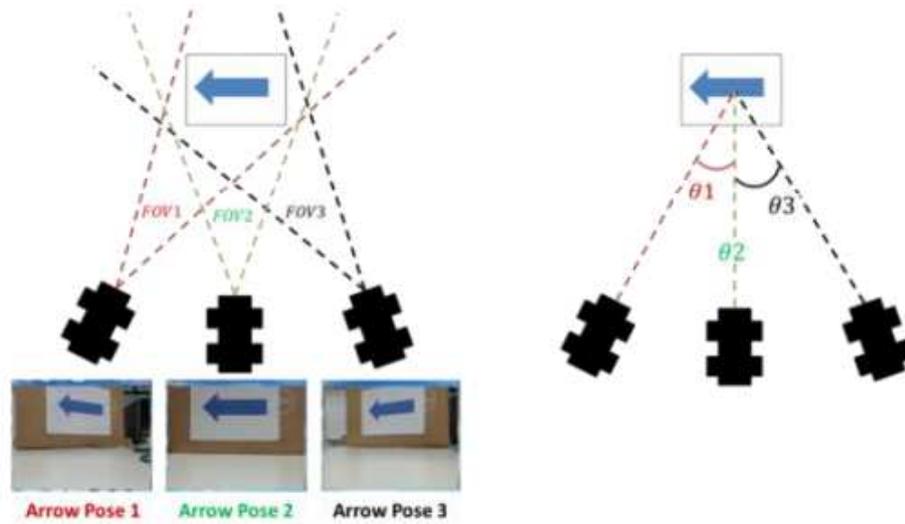


Fig. 8. Examples of rotation manoeuvres on mobile robot

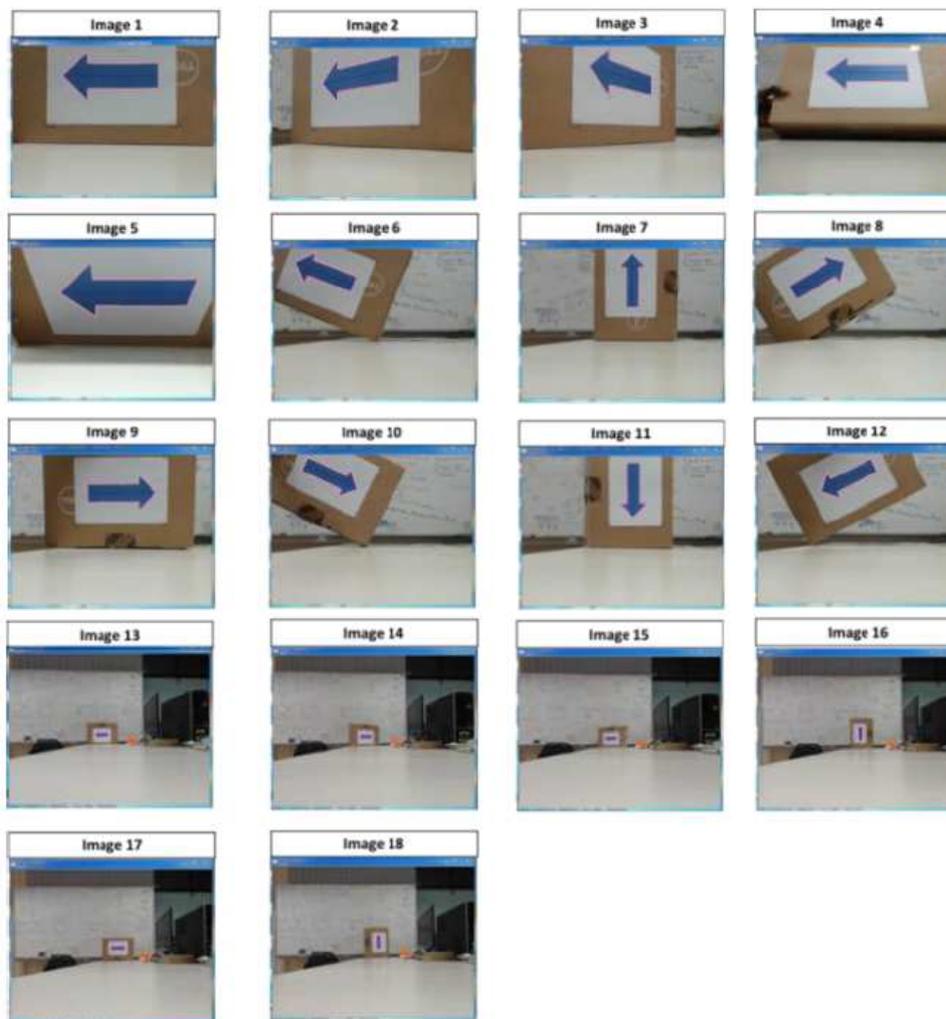


Fig. 9. A set of 18 images with different sign-board poses

Table. 1. Extracted Arrow Information from Fig. 9

		IMAGES									
		Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	
Arrow Information	Match	0.018	0.082	0.242	0.288	0.264	0.024	0.034	0.012	0.013	
	Depth (cm)	47.35	58.12	70.35	55.76	34.55	72.41	83.96	75.05	66.24	
	Direction (q°)	-90.18	-100.9	-72.71	-89.77	-90.27	-68.83	0	59.03	89.21	
	Position (ω°)	-0.65	-3.006	-0.24	2.51	4.14	-12.35	2.76	-10.07	2.43	
	Size (pixel)	24042	18700	14231.5	13577.5	36098	10786	7658	9862	12635	
	Length (pixel)	305.002	248.485	205.273	259.002	418.005	199.449	172	192.421	218.021	
Arrow Angles (°)	P1	89.62	108.451	102.12	76.1	79.41	93.28	88.69	91.41	92.08	
	P2	48.09	40.81	49.48	57.69	54.39	53.13	53.13	50.51	48.06	
	P3	93.59	85.82	113.7	99.32	100.008	99.28	98.58	97	93.36	
	P4	89.01	72.94	109.72	80.37	69.37	91.32	89.08	88.08	92.35	
	P5	91.48	102.374	77.94	100.2	110.63	88.89	91.82	92.26	87.98	
	P6	93.09	103.426	83.49	92.48	88.31	100.2	98.59	93.88	99.48	
	P7	48.46	44.65	37.925	57.44	54.5	52.84	54.43	48.61	52.36	
		Image 10	Image 11	Image 12	Image 13	Image 14	Image 15	Image 16	Image 17	Image 18	
Arrow Information	Match	0.026	0.009	0.045	0.128	0.214	0.211	0.057	0.089	0.109	
	Depth (cm)	77.44	83	74.58	320.946	379.936	390.34	320.867	313.895	320.867	
	Direction (q°)	116.42	180	-116.69	-90	-91.5	-90	1.27	88.75	178.72	
	Position (ω°)	-10.88	3	-2.11	-2.031	-1.95	-2.27	0.4	1.7	1.38	
	Size (pixel)	9232.5	8058.5	10146.5	627	525	517.5	610.5	643	620.5	
	Length (pixel)	186.489	174	193.644	45	38.01	37	45.01	46.01	45.01	
Arrow Angles (°)	P1	90	90.599	92.59	94.76	100.3	106.64	89.56	87.72	92.7	
	P2	50.21	48.92	54.52	71.2	65.65	56.49	61.5	81.17	76.67	
	P3	94.96	96.08	101.03	124.54	123.08	116.2	101.31	130.45	130.33	
	P4	91.11	88.28	92.85	84.11	82.65	99.16	95.19	86.59	83.16	
	P5	90	92.15	88.41	99.46	99.46	83.11	88.27	97.04	102.21	
	P6	101.87	94.99	97.66	111.8	111.8	118.84	115.27	113.64	113.71	
	P7	55.49	51.11	50.31	66.8	66.8	69.62	62.03	71.56	69.29	

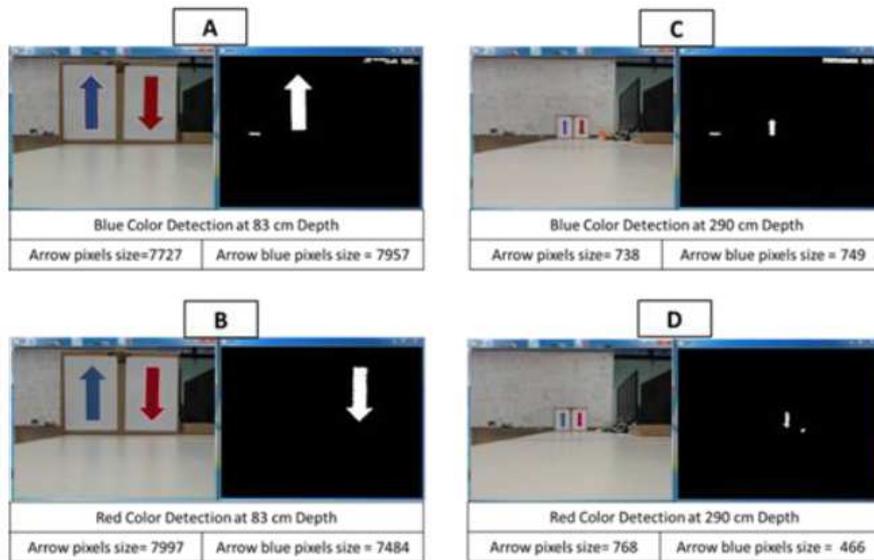


Fig. 10. Colored arrow detection using HSV color model

In the second experiment, a signboard with a colored arrow (blue and red) is adopted to represent specific path by using color detection. Using HSV color model for color detection, blue colored arrow is predefined within the range ((88, 80, 52), (115, 255, 255)) and red colored arrow within ((0, 150, 89), (12, 255, 255)). Applying these color ranges in the color detection

algorithm gave results as illustrated in Fig. 10. Results showed good robustness and accuracy of the algorithm in identifying the arrow color. The corresponding color of a detected arrow with a predefined color range would not only help in identifying the path to follow, but also aid in discarding the arrow that do not meet the predefined color range requirements.

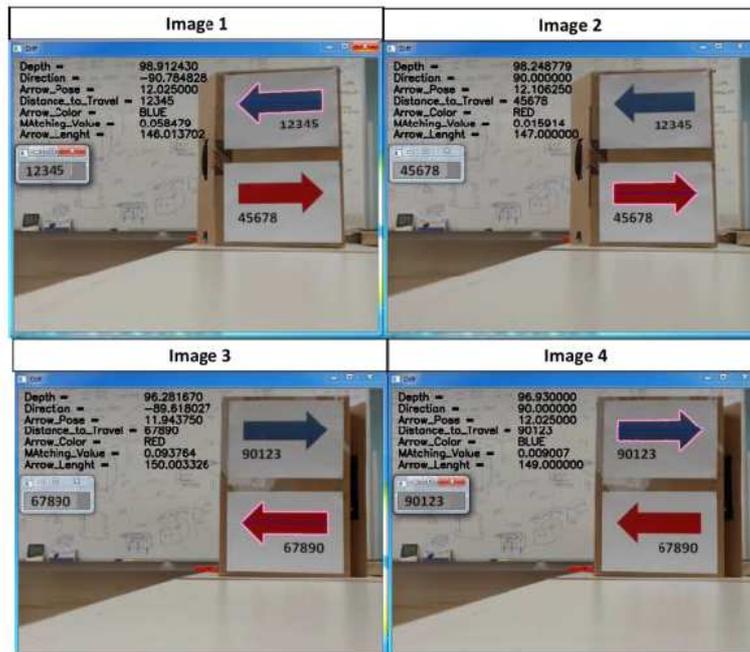


Fig. 11. Distance to Travel Information Detection and Recognition

In the third experiment, a set of colored arrow based signboards (red and blue), with different distance to travel information on each signboard, is captured with different poses over different distances. In addition of accurately identifying specific colored arrow within the environment, this experiment is also designed to evaluate the accuracy in detection and recognition of the numerical characters representing the distance to travel. The experiment results, in Fig. 11, show that the algorithm is able to detect and recognize the distance to travel information from the defined colored arrow up to a meter in this case study.

System Design and Implementation

This section outlines overall hardware setup and system framework design. The setup utilized the proposed algorithm illustrated in section 3.

System Hardware Setup

The system hardware consisted of a mobile robot, a laptop and two cameras (as shown in Fig. 12). The mobile robot is a PIONEER P3-AT from Adept. The mobile robot is attached to the laptop, LENOVO core i7 and 8GB ram, that hosted the algorithms. The laptop is connected to two cameras, labeled as CAMERA1 and CAMERA2. CAMERA1 is a Logitech HD c920, set with 640×480 (width × height) pixels resolution, 52×42 degrees (horizontal × vertical) field of view after calibration and capturing 30 Frames Per Second (FPS). CAMERA1 is positioned at forward-looking direction. This camera is responsible for detecting and following the arrow and

measuring the yaw rotation motions of the mobile robot. CAMERA 2 is a Logitech HD c390, set with 320×240 (width × height) pixels resolution, 42×42 degrees (horizontal × vertical) field of view after calibration and capturing 30 FPS. CAMERA2 is positioned at downward-looking direction. This camera's responsibility is to measure mobile robot's translational motions and to estimate the mobile robot's travelled distance.

System Framework

A system framework is designed to use data provided from the developed algorithms to derive the actual location of the mobile robot and the appropriate moves to execute. Given these three algorithms (yaw motion estimation, travelled distance estimation and artificial landmark recognition) running in parallel as threads, yaw estimation and travelled distance algorithms are used for the mobile robot real-time positioning while the artificial landmark recognition algorithm is used for the mobile real-time autonomous navigation. The framework real-time positioning consisted of retrieving yaw estimation and travelled distance estimation data, every 60 milliseconds interval, to compute and show on the graphical user interface map. The current position of the mobile robot $P(x,y)$ is represented using the Equation 15:

$$P(x,y) = \begin{cases} x = \cos \theta * TD \\ y = -\sin \theta * TD \end{cases} \quad (15)$$

where, TD is the travelled distance and θ is the yaw rotation.



Fig. 12. Two cameras setup for a mobile robot real-time positioning and autonomous navigation system

As for the real-time artificial landmark detection algorithm, it is used to guide the mobile robot through a specific path based on the colored landmark. Once a landmark describing a selected path is detected, depth and yaw pose information of the landmark are extracted to instruct the mobile robot to move toward the landmark centre at a specific speed until it reaches a certain predefined distance. For this experiment, the mobile robot adjusted itself within $\pm 5^\circ$ to the sight of the landmark centre and moved forward until it reaches 0.5 m to the landmark. The specifications are applied with respect to data integrity involving the measurements from yaw and travelled distance estimation algorithms obtained from Diop *et al.* (2014; 2015). After the mobile robot reached 0.5 m distance to the landmark, it stopped and retrieved the landmark pointing direction and together with the distance to travel information from the signboard. Once that information is retrieved, the mobile robot rotated in consideration with the accumulation of rotations made throughout the way; and travel to the distance as stated until another landmark is detected. If no landmark is detected, the mobile robot will stop at the distance specified by the last signboard.

Experimental Studies

This section illustrates the experiment studies of the proposed system with results and discussions.

Experimental Setup

The real-time positioning and autonomous navigation system experiment consisted of four phases. The first phase of the experiment involved testing the robustness and accuracy of the system in detecting a specific landmark navigated through and rotated in the direction

indicated. Three shuffled pieces of artificial landmarks are used in the first phase of the experiment. Two of the artificial landmarks are in blue color while the third artificial landmark is in red color. The direction of these landmarks is arranged in such way that different color landmarks are always pointed toward opposite direction. The mobile robot is placed at 2 m distance to these landmarks and it is instructed to navigate by following the red colored landmark with respect to the designed framework. The results of the mobile robot navigation, when the red colored landmark located at a different position on scene scenarios, are illustrated in Fig. 13. The second phase of the experiment involved testing the robustness and accuracy of the system in handling the error and travelled at the specified distance mentioned by the numerical characters image below the landmark. The mobile robot is located at 2 meters from the landmark. Three different experiment scenes were set. In each scene, the landmark is placed at different locations. To be more specific, the landmark is placed at the middle, the extreme left and the extreme right with respect to the captured image frame, respectively. For each scene, two different colored landmarks (blue and red) are used in which 2 and 5 m of distance to travel are specified on the blue and red colored landmarks, respectively. Results of the mobile robot navigation are illustrated in Fig. 14.

The third phase of the experiment involving testing the system's ability to stay on path in real-time navigation and to illustrate various navigation path error handling in real-time. To demonstrate this experiment, a specific navigation path consisted of four-sided polygon path, as illustrated in Fig. 15, is set. Red colored artificial landmarks with specific labels (1, 2, 3 and 4) and distance to travel are placed at each corner of the path to guide the mobile robot, as shown in both Fig. 15 and 16. Landmarks

are placed at the corners of the 4-sided polygon path, where each landmark is placed at the 50 cm lane to the previous one to create a rectangular path to navigate. To illustrate the error handling capability, different scenarios (A, B and C) were set in this experiment and are described as followed (with reference to Fig. 15):

- In scenario A, the landmark 2 is aligned with the landmark 3 to create an obvious error handling situation
- In scenario B, the landmark 1 is aligned with the landmark 2 to create an obvious error handling situation
- In scenario C, landmark 1 and 2 are aligned and landmark 2 and 3 are aligned to create multiple error handling situations

The final phase of the experiment was performed to evaluate the mobile robot's ability to stay on path for a long run. The experiment consisted of instructing the mobile robot to repeatedly navigate in a 4-sided rectangular path, as illustrated in Fig. 17. In this experiment, the mobile robot was left navigating for more than half hour, performing 8 complete rounds of the rectangular path.

Analysis and Discussions

Based on the experiments, the real-time positioning and autonomous navigation system and framework had shown a good response to the directives provided by the artificial landmarks, as illustrated in the 1st two phases of the experiment. Good responses were observed in the third phase of experiment in handling the error correction on rotations and translation over specify distance specified by the landmark. In the final phase of the experiment, the mobile robot is able to navigate through the specified path scenario for half an hour with an average error of 0.167 m from the ideal path and 0.080 m average deviation from the first round of the navigation, as shown in Table 2. The system accuracy appears to be acceptable as compared to some existing vision-based solutions for indoor environments, as illustrated in Table 3. The low navigation path error generated from this system allows the mobile robot to navigate in a narrow indoor environment. A real-time indoor navigation experiment is setup and shown in the following link. The video illustrated the capability of the system to navigate the mobile robot in a narrowed, indoor environment. <https://drive.google.com/open?id=0B3VoMqHLevubRWxUSDZLcmlRZEU>.

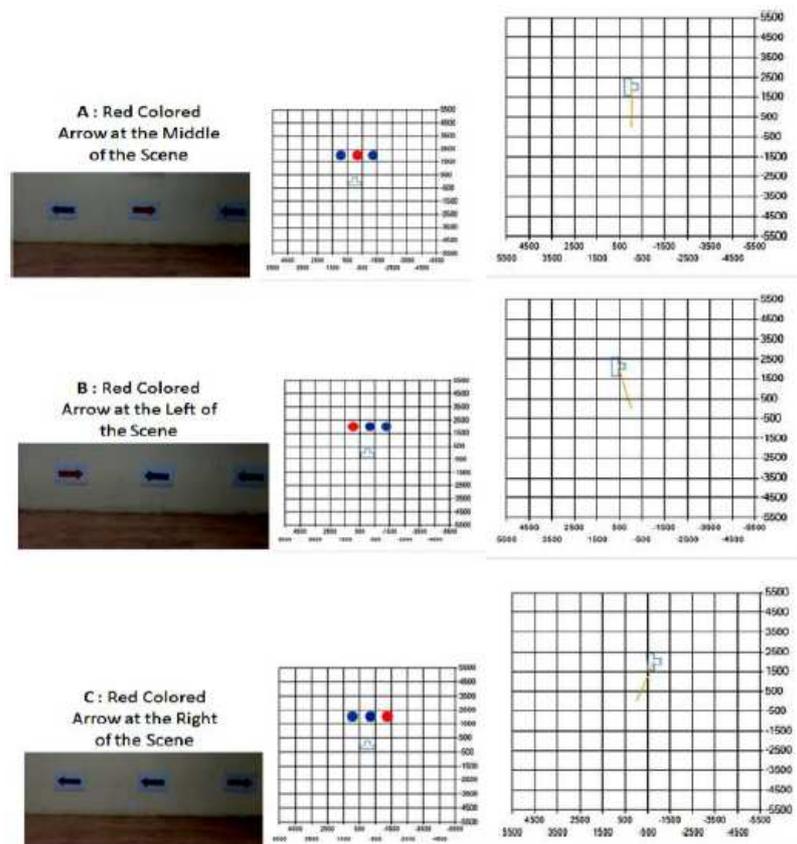


Fig. 13. Real-time positioning and autonomous navigation system framework

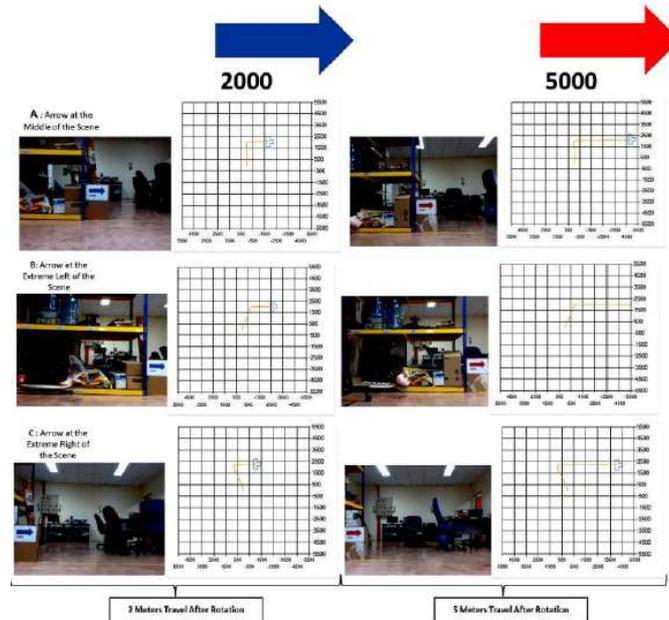


Fig. 14. Mobile robot response on driven artificial landmark directives

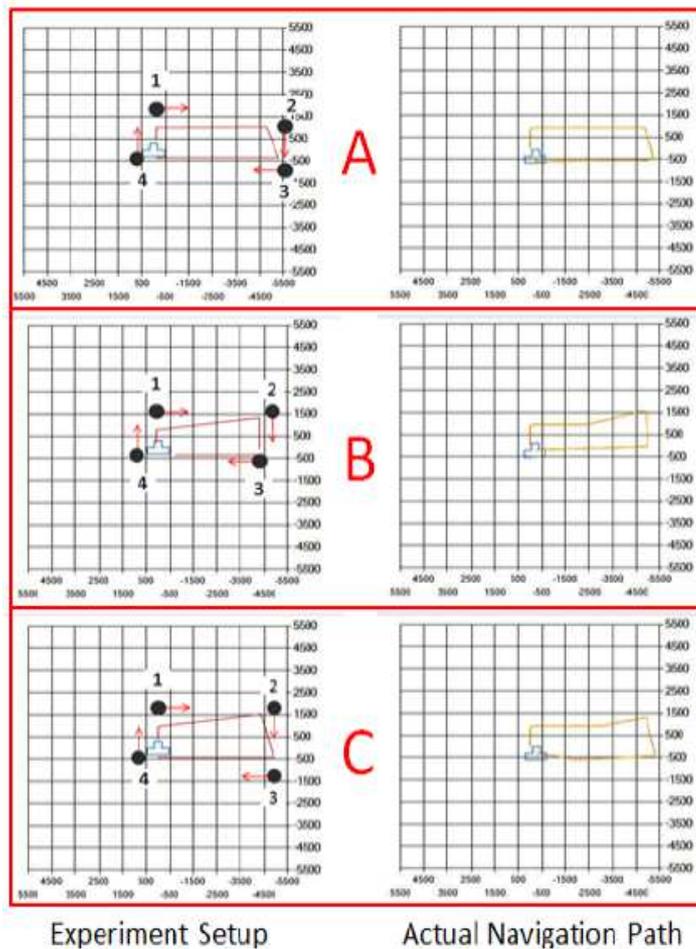


Fig. 15. Errors handling during mobile robot navigation

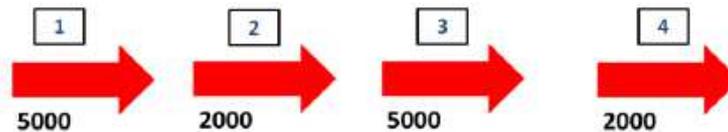


Fig. 16. Sets of signboards labels

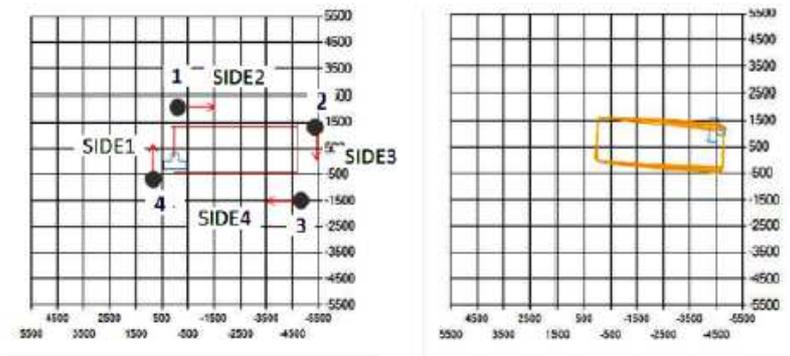


Fig. 17. Path following in mobile robot navigation

Table. 2. The RMSE of the navigation experiment from Fig. 17

	Root Mean Square Deviation (RMSD) of the mobile robot (in meter)				Average (mean)
	SIDE1	SIDE2	SIDE3	SIDE4	
Compared to the ideal path	0.031	0.119	0.263	0.254	0.167
Compared to the first round	0.02	0.141	0.027	0.134	0.08

Table. 3. Comparative table of few existing vision based indoor positioning solutions

Few existing vision based positioning solutions accuracy (in meter) Experimented in indoor Environments				
Chen and Birchfield (2009)	De Cristóforis <i>et al.</i> (2015)	Clement <i>et al.</i> (2015)	Atia <i>et al.</i> (2015)	Rivera-Rubio <i>et al.</i> (2015)
0.14	0.19	0.2	0.99	1.3

A few limitations can be observed from this system. Firstly, the system is very sensitive to the landmark's pose and pointing direction. Landmarks that are not well positioned in the environment could generate errors in depth estimation and pointed direction rotation. These errors, particularly in pointing direction information, could lead to a large deviation of the mobile robot from its path at a long run. Secondly, since the color is used to represent specific paths, the system had become sensitive to extreme lighting environments (very dark or very bright). This constraint, however, can be avoided with constant ambient indoor environment.

Conclusion

This paper aimed to address the potential of vision sensors to accurately estimate the real time position and to guide a mobile robot throughout navigation at indoor environments. The mobile robot's real-time positioning and autonomous navigation system, aided by vision, were achieved using features points and visual

landmarks as artificial landmark. This research has uncovered a significant potential in using vision for real-time positioning and autonomous navigation in the indoor environments. Even though the results obtained from developed algorithms seems acceptable, there are yet lots of rooms for improvement in measurement accuracy for a complete autonomous navigation system. Improvement of this system involves adding an Inertial Measurement Unit (IMU) or Lidar as an additional sensor to aid vision for a better accuracy in indoor positioning. Additional module could be integrated along with obstacle avoidance, clear path recognition and path planning in order to handle busy environment, reduce path deviation in long runs and offer more flexibility during navigation.

Acknowledgment

Authors would like to thank all Center of Remote Sensing and Surveillance Technology (CRSST) members for their support through the realization of this project.

Funding Information

This research is partially funded by Telekom Malaysia, under the TM R&D Funding, under the project number RDTC/130824 and partially supported by Ministry of Higher Education Malaysia (MOHE) under Grant MMUE/130103 and Grant 203/PJJAUH/6711279.

Author's Contributions

Diop Mamadou: Involving in carried out study, algorithm and framework development, data collection and manuscript writing.

Lim Chot Hun: Developed the overall plan of research and involved in carried out study, analysis and provided critical reviewing on the data.

Lim Tien Sze: Participated in the overall research and involved in drafting the manuscript.

Ong Lee Yeng: Participated in the overall research and provided critical reviewing on the manuscript for significant intellectual content.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

- Atia, M.M., S. Liu, H. Nematallah, T.B. Karamat and A. Noureldin, 2015. Integrated indoor navigation system for ground vehicles with automatic 3-d alignment and position initialization. *IEEE Trans. Vehicular Technol.*, 64: 1279-1292. DOI: 10.1109/TVT.2015.2397004
- Bhattacharayya, N., A.K. Singh, S. Kapil, H. Jain and A. Jain, 2014. Traffic light solution for UGV using digital image processing. *Int. J. Soft Comput. Eng.*, 4: 25-27.
- Bradski, G. and A. Kaehler, 2008. *Learning OpenCV: Computer Vision with the OpenCV Library*. 1st Edn., O'Reilly Media, Inc., ISBN-10: 0596516134, pp: 555.
- Chen, Z. and S.T. Birchfield, 2009. Qualitative vision-based path following. *IEEE Trans. Robot.*, 25: 749-754. DOI: 10.1109/TRO.2009.2017140
- Clement, L.E., J. Kelly and T.D. Barfoot, 2015. Monocular Visual Teach and Repeat Aided by Local Ground Planarity. In: *Field and Service Robotics*, Wettergreen, D.S. and T.D. Barfoot (Eds.), Springer, pp: 547-561.
- De Cristóforis, P., M. Nitsche, T. Krajník, T. Pire and M. Mejail, 2015. Hybrid vision-based navigation for mobile robots in mixed indoor/outdoor environments. *Patt. Recogn. Lett.*, 53: 118-128. DOI: 10.1016/j.patrec.2014.10.010

- Diop, M., C.H. Lim, T.S. Lim and L.Y. Ong, 2014. A vision-based travelled distance estimation algorithm in an indoor environment. *J. Comput. Sci.*, 10: 2469-2480. DOI: 10.3844/jcssp.2014.2469.2480
- Diop, M., L. Chot-Hun, L. Tien-Sze and O. Lee-Yeng, 2015. A vision-aided motion detection: 3-DOF orientation estimation algorithm using mobile robot. *Int. J. Control Automat. Measure.*, 8: 99-116. DOI: 10.14257/ijca.2015.8.5.10
- Engel, J., J. Sturm and D. Cremers, 2012. Accurate figure flying with a quadcopter using onboard visual and inertial sensing. *IMU*, 320: 240-240.
- Hesch, J.A., D.G. Kottas, S.L. Bowman and S.I. Roumeliotis, 2013. Camera-IMU-based localization: Observability analysis and consistency improvement. *Int. J. Robot. Res.*, 33: 182-201. DOI: 10.1177/0278364913509675
- Hong, S., J.B. Song, J.H. Baek and J.K. Ryu, 2012. Visual odometry for outdoor environment using a downward-tilting camera and self-shadow removal algorithm. *Proceedings of the 12th International Conference on Control, Automation and Systems*, Oct. 17-21, IEEE Xplore Press, JeJu Island, pp: 960-963.
- Huang, B., 2013. Floor plan based indoor vision navigation using smart device. Doctoral dissertation, University of Calgary.
- Hu, M.K., 1962. Visual pattern recognition by moment invariants. *IRE Trans. Inform. Theory*, 8: 179-187. DOI: 10.1109/TIT.1962.1057692
- Li, X., S. Jia, J. Fan, L. Gao and B. Guo, 2012. Autonomous mobile robot guidance based on ground line mark. *Proceedings of the SICE Annual Conference*, Aug. 20-23, IEEE Xplore Press, Akita, pp: 1091-1095.
- Yang, K.H., W.S. Yu and X.Q. Ji, 2012. Rotation estimation for mobile robot based on single-axis gyroscope and monocular camera. *Int. J. Automat. Comput.*, 9: 292-298. DOI: 10.1007/s11633-012-0647-z
- Ibraheem, N.A., M.M. Hasan, R.Z. Khan and P.K. Mishra, 2012. Understanding color models: A review. *ARNP J. Sci. Technol.*, 2: 265-275.
- Mercimek, M., K. Gulez and T.V. Mumcu, 2005. Real object recognition using moment invariants. *Sadhana*, 30: 765-775. DOI: 10.1007/BF02716709
- Ortega-Garcia, J.L., J.L. Gordillo and R. Soto, 2014. A new method to follow a path on indoor environments applied for mobile robotics. *Proceedings of the 11th IEEE International Conference on Control and Automation*, Jun. 18-20, IEEE Xplore Press, Taichung, pp: 631-636. DOI: 10.1109/ICCA.2014.6870993
- Rivera-Rubio, J., I. Alexiou and A.A. Bharath, 2015. Appearance-based indoor localization: A comparison of patch descriptor performance. *Patt. Recogn. Lett.*, 66: 109-117. DOI: 10.1016/j.patrec.2015.03.003

- Rojas Castro, D.M., A. Revel and M. Menard, 2015. Document image analysis by a mobile robot for autonomous indoor navigation. Proceedings of the 13th International Conference on Document Analysis and Recognition, Aug. 23-26, IEEE Xplore Press, Tunis, pp: 156-160.
DOI: 10.1109/ICDAR.2015.7333743
- Smith, R., 2007. An overview of the tesseract OCR engine. Proceedings of the 9th International Conference on Document Analysis and Recognition, Sept. 23-26, IEEE Xplore Press, Parana, pp: 629-633.
DOI: 10.1109/ICDAR.2007.4376991
- Ye, A., H. Zhu, Z. Xu, C. Sun and K. Yuan, 2012. A vision-based guidance method for autonomous guided vehicles. Proceedings of the IEEE International Conference on Mechatronics and Automation, Aug. 5-8, IEEE Xplore Press, Chengdu, pp: 2025-2030.
DOI: 10.1109/ICMA.2012.6285133
- Zhang, J., S. Singh and G. Kantor, 2014. Robust Monocular Visual Odometry for a Ground Vehicle in Undulating Terrain. In: Field and Service Robotics, Yoshida, K. and S. Tadokoro (Eds.), Springer Berlin Heidelberg, pp: 311-326.
- Zhao, Y., Y. Liu and L.M. Ni, 2007. VIRE: Active RFID-based localization using virtual reference elimination. Proceedings of the International Conference on Parallel Processing, Sept. 10-14, IEEE Xplore Press, Xi'an, pp: 56-56.
DOI: 10.1109/ICPP.2007.84
- Zhuang, Y., Z. Syed, Y. Li and N. El-Sheimy, 2015. Evaluation of two WiFi positioning systems based on autonomous crowd sourcing on handheld devices for indoor navigation. IEEE Trans. Mobile Comput. DOI: 10.1109/TMC.2015.2451641