Experimental results on position and path control of an automated guided vehicle using fixed camera at ceiling and color markers

Ramesh Pungle[1], Atul Andhare[2], Durva Pungle[3]
[1]Department of Mechanical and Automation Engineering, P.E.S. College of Engineering, Chhatrapati Sambhajinagr, India
[2]Department of Mechanical Engineering, Visvesvaraya National Institute of Technology, Nagpur, India
[3]Department of Computer Engineering, SCTR's Pune Institute of Computer Technology, Pune, India

Article Info

Article history:

Received Feb 24, 2024
Revised Jul 30, 2024
Accepted Aug 12, 2024

Keywords:

Automated guided vehicle
Ceiling camera
Control
Mapping
Path planning

ABSTRACT

This article presents the results of experiments on path planning and control of automated guided vehicles (AGV) using single, fixed ceiling mounted, monocular cameras and colored markers. The camera employed in the system serves as both a sensor and controller. Initially, the working environment is structured using colored markers for given applications. For every new setup, structuring the environment is essential. The image processing algorithm identifies the colored markers and their positions, which are then utilized for path planning and segmentation. The actuation time required to transverse each segment is calculated and then AGV is actuated accordingly. A transformation or inverse mapping matrix (M), predetermined, is employed for calculating world coordinates from given image coordinates. Path planning and AGV control are across various paths, both with and without static obstacles, in real-time applications. The colored marker detection and recognition accuracy for the given setup have been found cent percentage while the AGV reaches the goal point with an error margin of around 3.9% on straight paths, both with and without obstacles.

Corresponding Author:

Dr. Ramesh Pungle
Department of Mechanical and Automation Engineering, P.E.S. College of Engineering
Chhatrapati Sambhajinagr, India
Email: rameshgpungle@gmail.com

1. INTRODUCTION

Traditionally, in automated guided vehicle (AGV) navigation, it has been common practice to utilize a variety of sensors, both onboard (mounted directly on the AGV) and off-board (positioned on the ceiling or at suitable locations), for tasks such as path detection, obstacle detection, and navigation towards a destination while avoiding obstacles. In AGV navigation, the general tradition has been to use different sensors; on board (mounted on the AGV) and off-board (fixed at the ceiling or suitable location), to detect or generate a path, detect obstacles, and also to reach the destination by avoiding the obstacles. Onboard sensors are typically favored for their ability to detect and circumvent both static and dynamic obstacles along a given path. These sensors enable local path planning and motion planning by providing real-time feedback directly to the AGV.

Off-board sensors, on the other hand, are often employed to plan the overall path, taking into account the positions of the AGV and static obstacles within the environment. However, employing multiple sensors onboard the AGV can introduce synchronization challenges among their outputs, potentially leading to complex control systems and increased response times. Alternatively, if a single off board sensor is capable of performing all of these tasks, the problem becomes significantly simpler to solve, and the complexity of the controller can be minimized. By centralizing these functions in a single sensor, the need for synchronization among multiple sensors is eliminated, streamlining the control process and enhancing overall efficiency.

Vision-based navigation is one such innovation. Recent advancements in image processing hardware and software have enabled fast and feasible real-time visual applications. Consequently, visual guidance and obstacle avoidance have become the focus of extensive research [1]. In this context, the current study addresses the challenge of achieving accurate, precise, and robust position estimation in indoor environments. The objective of this research is to utilize a single camera as a sensor and controller to execute planned activities using the AGVs. Image processing is an important issue for many computer vision applications, such as the structure of the environment, object recognition, and pose estimation. The camera model used in computer vision has been modeled by perspective projection. This has been an ideal and accurate model for a wide range of existing cameras. However, the resulting equations from perspective projection are complicated and often nonlinear due to the unknown scaling factor. In vision-based navigations; generally structured or non-structured environments are employed.

In AGV navigation, three methods are mostly adopted which include map-based navigation, map-building navigation, and map-less navigation. In map-based navigation system consists of providing the AGV with the model of the environment. These models contain different degrees of detail, varying from a complete CAD model of the environment to a simple graph of interconnections or interrelationships between the elements in the environment. Further details about such navigation systems are found in [2]–[6]. In map-building navigation, AGVs can explore the environment and build their map by themselves. The navigation process starts once the robot has explored the environment and stored its representation. The navigation is simultaneous localization and mapping (SLAM). This technique is found suitable for local path and motion planning. The applications of this technique in real-time can be found in [7]–[11]. The researchers have used onboard sensors and controllers to achieve SLAM. In map-less navigation, the AGV does not need knowledge of the environment to run. The movements of the robot depend on the elements observed in the environment (walls, features, doors, and desks). Optical-flow-based and appearance-based techniques are generally used in this technique. Further details are given in [10], [12]. This technique is good for local path planning with an onboard vision sensor. Some researchers [4], [6], [8] have utilized various onboard and off-board sensors in an attempt to develop robust navigation systems. However, the controllers implemented in these systems were often overly complex, and the response from the sensors was inadequate. The applications involving semantic maps are outlined in [13], with classifications and scalability. In [14], how colored object is recognized by histogram matching for the various color models is analyzed and compared.

An AGV navigation using fiduciary marks design and experiments are presented in [15]. In [16], [17], single omnidirectional camera is used, and an area of interest is extracted, and by using image processing the vehicle is navigated on a set path. Color and shape feature extraction using a vision model for accurate recognition of traffic signs located at a reasonable distance is outlined in [18]. Ranganathan *et al.* [19] describes a navigation system for a mobile robot that must execute motions in a building. A qualitative analysis of a USB camera for an AGV control is given in [20]. Colored marker recognition and an AGV position and path control implementation are explained in [21]–[24].

The main objective of this study is to utilize a single off board camera both as a sensor and a controller without employing any other types of internal and external sensors. The camera is utilized to perceive the environment, formulate path planning, and control AGV movements by avoiding static obstacles. Experiments were conducted on several paths using the camera sensor.

2. PROPOSED SYSTEM OVERVIEW

The scientific rationale behind developing this system is to eliminate the need for various sensors fixed on the AGV to collect different environmental information. The single web camera is utilized to map the environment, recognize the obstacles, plan the path, and control the AGV along that path. The AGV navigation is planned and executed in a structured environment by considering the limitations of a single camera working both as a sensor and controller. In such a navigation system, environment mapping is essential for accurate positioning of the AGV, determining the end position, and landmarks. The positions of the AGV or obstacles, as determined by the camera, are initially in image coordinates (in pixels) and need to be mapped into world coordinates (in millimeters or meters) to ascertain the actual path traveled by the AGV in the world coordinate system. Therefore, as the first step, the inverse mapping matrix (M_{inv}) for 2D points in the world plane and their corresponding points in the image plane is determined using a checkerboard sheet. For estimating the mapping matrix (M), a checkerboard sheet with 100 mm × 100 mm square is placed in the environment, and the camera is activated to capture images of the environment. Using a camera’s pinhole model, the projection from the 3D space to the image plane can be described with (1).

$$\lambda\begin{bmatrix}u\\v\\1\end{bmatrix}=\mathrm{K[RT]}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix} \tag{1}$$

where λ is an arbitrary scale factor, p = [u v 1] ^{T} are the coordinates of the 2D point projected onto the image plane, P = [X, Y, Z, 1] ^{T} are the coordinates of the point in the reference or world plane of the 3D space. R represents the rotation of the coordinate system of the 3D points with respect to the origin; T is a vector representing the translation of the origin of the camera coordinate system, [R T] represents the transformation between the camera frame and the world coordinate system.

Equation (1) can be simplified as (2).

$$\lambda|\bar{p}|=\mathrm{K[RT]}|\bar{\mathrm{P}}|,\text{further can be simplified as }\lambda|\overline{p_\imath}|=\mathrm{M}|\bar{\mathrm{P}}_\mathrm{i}| \tag{2}$$

The projection matrix M can be estimated by detecting point correspondence with p_i = [u, v, 1] ^{T} being the pixel positions and P_i = [X, Y, 1] ^{T} being the world coordinates of the corresponding feature point of index, 'i' on the planar pattern or checkerboard. In order to eliminate the scaling factor, cross product of each term of (2) is taken as (3).

$$\bar{p}_i\times|\lambda\overline{p_\imath}|=\bar{\mathrm{p}}_\mathrm{i}\times|\mathrm{M}\bar{\mathrm{P}}_\mathrm{i}|,\text{since cross product of }\lambda|\bar{p}_i\times\bar{p}_i|=0 \tag{3}$$

Equation (3) can be written as (4).

$$\bar{\mathrm{p}}_\mathrm{i}\times|\mathrm{M}\bar{\mathrm{P}}_\mathrm{i}|=\begin{bmatrix}\mathrm{u}_1\\\mathrm{v}_1\\1\end{bmatrix}\begin{bmatrix}m^{1T} & \bar{P}_i\\m^{2T} & \bar{P}_i\\m^{3T} & \bar{P}^i\end{bmatrix} \tag{4}$$

Equation (4) is formulated as (5).

$$\begin{bmatrix}0_\mathrm{i}^\mathrm{T} & -\mathrm{P}_\mathrm{i}^{-\mathrm{T}} & \mathrm{v_iP_i^{-T}}\\\mathrm{P}_\mathrm{i}^{-\mathrm{T}} & 0_\mathrm{i}^\mathrm{T} & -\mathrm{u_iP_i^{-T}}\\-\mathrm{vP_i^{-T}} & \mathrm{u_iP_i^{-T}} & 0_\mathrm{i}^\mathrm{T}\end{bmatrix}\begin{bmatrix}\mathrm{m}^1\\\mathrm{m}^2\\\mathrm{m}^3\end{bmatrix}=0 \tag{5}$$

Equation (5) is in the form of AM = 0, where A is a 3×9 matrix and M is a vector with row-wise nine elements of M. The AM is linear in M and has two linearly independent equations i.e., one row can be removed. Removing the third row of (5), the following equation is obtained.

$$\begin{bmatrix}0_\mathrm{i}^\mathrm{T} & -\bar{\mathrm{P}}_\mathrm{i}^\mathrm{T} & \mathrm{v_i}\bar{\mathrm{P}}_\mathrm{i}^\mathrm{T}\\\bar{\mathrm{p}}_\mathrm{i}^\mathrm{T} & 0_\mathrm{i}^\mathrm{T} & -\mathrm{u_i}\bar{\mathrm{P}}_\mathrm{i}^\mathrm{T}\end{bmatrix}\begin{bmatrix}\mathrm{m}^1\\\mathrm{m}^2\\\mathrm{m}^3\end{bmatrix}=0,\text{where M}=\begin{bmatrix}\mathrm{m}_1 & \mathrm{m}_2 & \mathrm{m}_3\\\mathrm{m}_4 & \mathrm{m}_5 & \mathrm{m}_6\\\mathrm{m}_7 & \mathrm{m}_8 & \mathrm{m}_9\end{bmatrix} \tag{6}$$

Equation (6) is valid for all homogeneous representations. The matrix M has 8 parameters along with a 9th homogeneous scaling factor. To solve for the parameters of M, a sufficient number of equations are required. Since each point correspondence gives rise to two equations (one for the x-coordinate and one for the y-coordinate), you need at least four correspondences (8 equations) to solve for the 8 parameters in M. However, in practice, more correspondences are often used to create an over-determined linear system. This redundancy helps improve the robustness and accuracy of the solution by minimizing errors and handling outliers more effectively.

The over-determined equations are solved using the linear least square technique to determine matrix (M). The complete procedure is outlined in [25]. The inverse mapping matrix (M_{inv}) can be calculated by simply taking the inverse of the mapping matrix (M) i.e., ($M^{-1)}$. Equation (7) is utilized to determine any world point from its corresponding image point.

$$P_{(world)}=M_{inv}\mathrm{P}_{(\mathrm{image})} \tag{7}$$

The system consists of a personal computer (PC), camera, radio frequency (RF) module, colored markers, and an AGV as shown in Figure 1. The colored markers are used for structuring the environment as per the requirement. The actuation time is determined by recognizing the positions of the colored markers in both the image as well as world coordinate system. The AGV is then actuated for a specified time and then feedback is taken through visual information obtained from the camera.

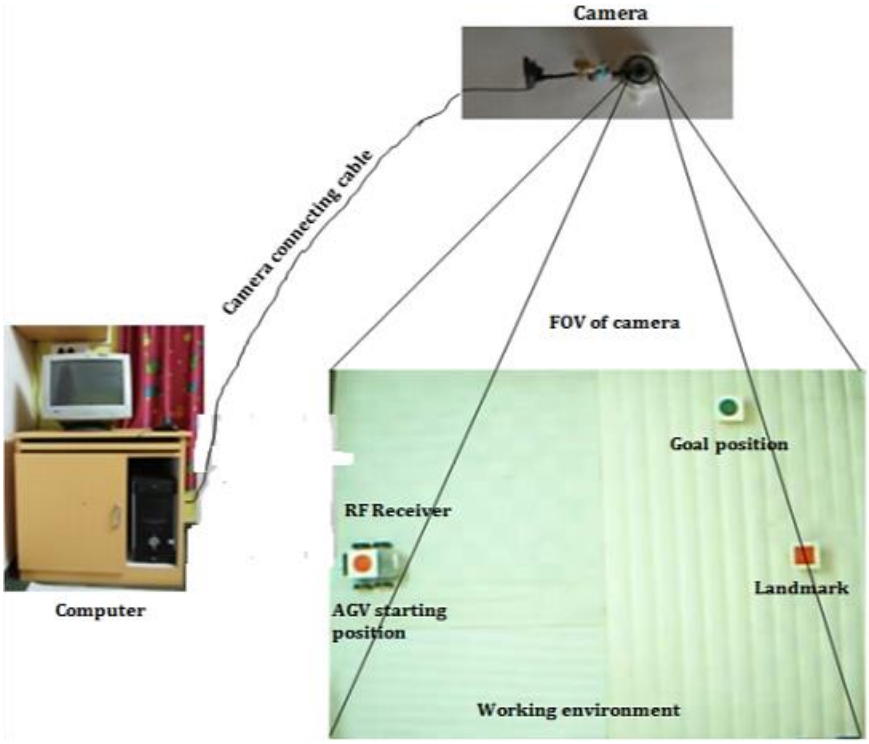


Figure 1. Experimental setup layout

3. COLOURED MARKER RECOGNITION

This section presents the results of acquiring and recognizing circular markers of colors such as red, green, and blue. Additionally, colored markers of shapes like square, triangular, and circular may also be used in experimentations. These markers are placed in a testing environment where the camera is fitted to the ceiling. Initially, a red colored marker of a circular shape was placed in the mapped environment. High-quality images captured by the camera are selected for further processing in MATLAB using an image processing algorithm. The original-colored markers and their processed images are shown in Figures 2 and 3. The blob analysis technique is used to process RGB images to detect and recognize colored markers in the input image. During blob analysis, a blob is a region that has a value of 1 (or true). There is no way for the different color blobs to overlap in the binary image and therefore, only one blob is considered even though blobs have different colors. The complete process of object recognition is grouped into four operations: labeling, grouping, extracting, and matching. Once a set of regions has been identified, the properties of the regions become the input to the higher-level procedure that performs decision-making tasks such as recognition or inspection.

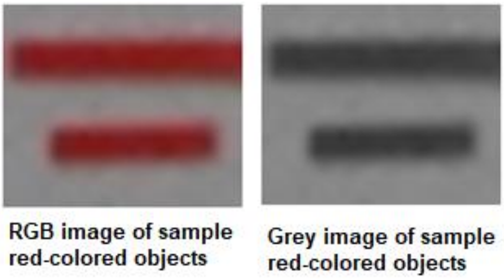
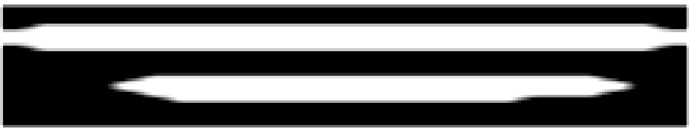

RGB image of sample red-colored objects
Grey image of sample red-colored objects

Figure 2. Expanded view of the RGB and grey image of sample red-colored objects



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0
0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0
0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0
0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0
0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Pixel distributions and Connected component Labeling

Figure 3. The connected component labeling in binary image

The blob analysis was performed on an RGB image that has a red color circular marker with a white background. MATLAB code was developed to perform blob analysis on the given image. The experimental results of blob analysis are presented step by step in Figure 4. Markers of the same color with different shapes or the same shape with different colors can also be used for experimentation. The major steps of the image processing algorithm employed in experimentation are shown in Figure 4. In [17], colored marker recognition techniques using the Hough transform were utilized, claiming an average recognition accuracy of around 98.8%.

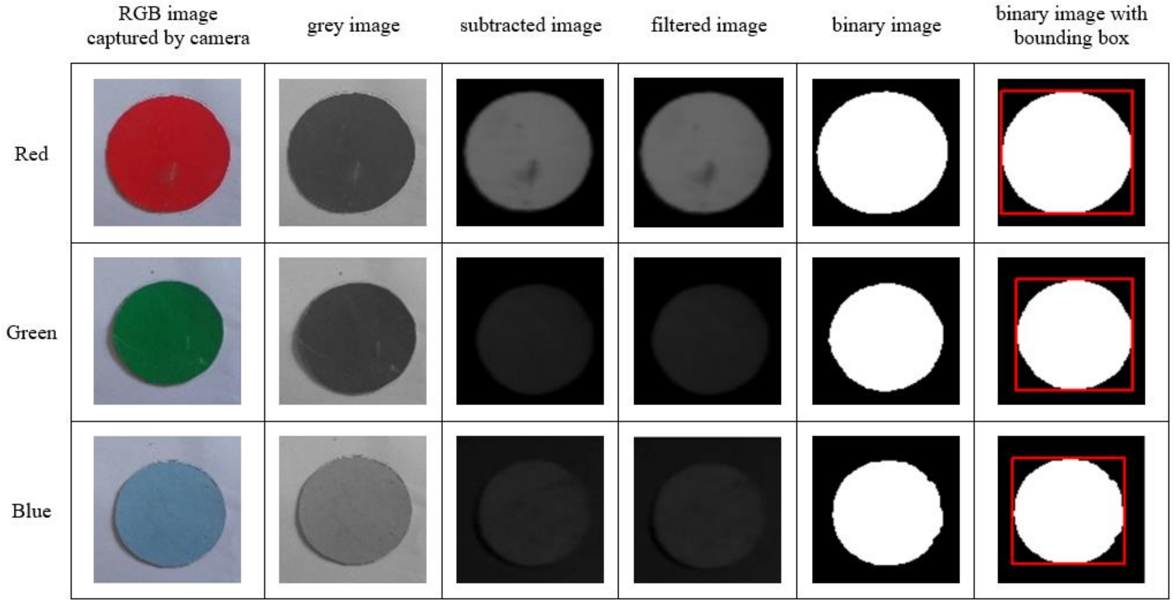


Figure 4. Display of results of image processing algorithm

4. EXPERIMENTAL RESULTS AND DISCUSSION

The navigation and path planning of an AGV gives a reliable estimation of the vehicle's position and an accurate workspace or navigation map [14]. The approach adopted here involves decomposing the given path into straight segments and circular curves or simple 90-degree turns. However, such paths cannot be precisely tracked due to discontinuities in curvature at transition points between segments, which necessitate instantaneous accelerations. Taking into account the constraints of image processing, position control of the AGV, and the field of view (FOV) of the environment using a single camera, a real-time position and path control system has been developed and utilized in the experimentations and for corrections.

The experimentation of AGV presented in this paper considers the position and path control of AGV on straight paths with and without static obstacles using a single camera as a sensor and controller. In a straight path with no obstacles, path planning is simple, and the path is converted into a single segment. In the case of a straight path with a static obstacle in a given path, the total path needs to be segmented into straight paths and 90-degree turns or circular arcs. Then, the time required to cover the given segment is determined. As the speed of the AGV is predetermined, only the segment distance needs to be calculated, either in pixels or in millimeters.

The time required to cover a particular segment is calculated by using (8), and the time required to execute a 90-degree turn is calculated by using (10).

$$t_1 = \frac{d_1}{v_1}, t_2 = \frac{d_2}{v_2}, t_3 = \frac{d_3}{v_3}, \dots\dots\dots\dots.., t_n = \frac{d_n}{v_n} \tag{8}$$

Equation (8) can be summarized as (9).

$$T_{seg} = \frac{1}{v}\sum_{i=1}^{n} d_i \tag{9}$$

As the AGV speed is calibrated, the time required to take a 90-degree turn is calculated using (10).

$$T_{turn} = \sum_{j=1}^{n} T_{rij} \tag{10}$$

The total time to cover a given path can be calculated by adding the times obtained from (9) and (10). The AGV with differential drive configuration was developed and employed in the experimentation. It has four wheels, out of which two rear wheels are connected to two geared DC motors of the same speed, driven independently, and the front two wheels are used only for balancing purposes. A 12-volt battery is used to supply the required voltage and current.

4.1. AGV on straight path

To test the performance of an AGV on the straight path, two distinct path conditions were tested: a straight path with no static or dynamic obstacle and a straight path with a static obstacle. In case 1, the AGV was tested on a clear, unobstructed straight path. This condition served as a baseline to assess the AGV’s basic performance, including its ability to maintain a straight trajectory and its responsiveness to control inputs when no external factors are present. In case 2, static obstacles were placed along the straight path. These obstacles do not move but require the system to detect the obstacles, an AGV, and navigate around them.

4.1.1. Straight path with no obstacles

In straight path planning with no obstacles in the path, an AGV (with a red colored circular marker at the top) is placed at the starting position in switched-off condition. Another marker, colored red, green, or blue is placed at the goal (end) position as in Figure 5(a). When the system starts, the camera captures the first image and subsequently, the image is processed in MATLAB and the initial positions of the AGV and end position are determined. The positions detected by image processing algorithms and segmented paths are depicted in Figure 5(b). These positions are subsequently transformed into world coordinates using an inverse mapping matrix (M_{inv}) to compare the actual path traveled by the AGV on the ground floor. Further, these positions are also utilized to generate the total path and to calculate the distance between the starting point and the goal point, both in the image as well as in the world coordinate system.

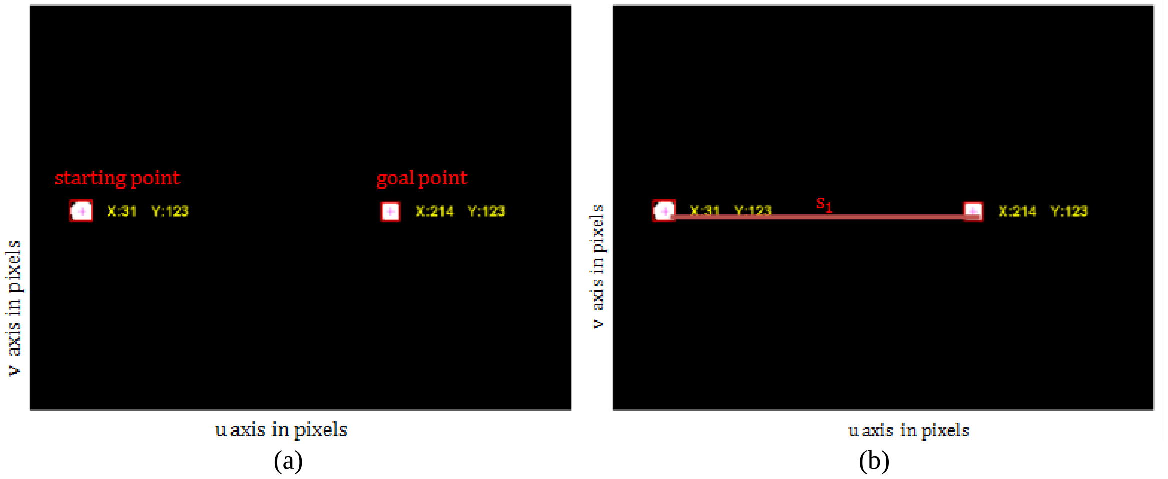


Figure 5. Binary image showing (a) the positions of starting and goal point and (b) the segmented path

Using the distance and the calibrated AGV speed, the time required for the AGV to reach the goal point is calculated. Following this calculation, the AGV is then actuated for that duration through an RF remote control. The distance between two positions is determined by using (11) in image coordinates d_{image}(pixels) and in world coordinates d_{world} (mm).

$$d_{image} = \sqrt{(u_g - u_s)^2 + (v_g - v_s)^2}, \qquad d_{world} = \sqrt{(X_g - X_s)^2 + (Y_g - Y_s)^2} \tag{11}$$

The actual path traveled by the AGV during testing and the predicted path utilizing calibrated speed during a given time duration is depicted in Figure 6. On the straight path, the distance covered by the AGV in the world coordinates system and the actual distance measured on the floor are given in Table 1. The percentage error was found to be 2.41%.

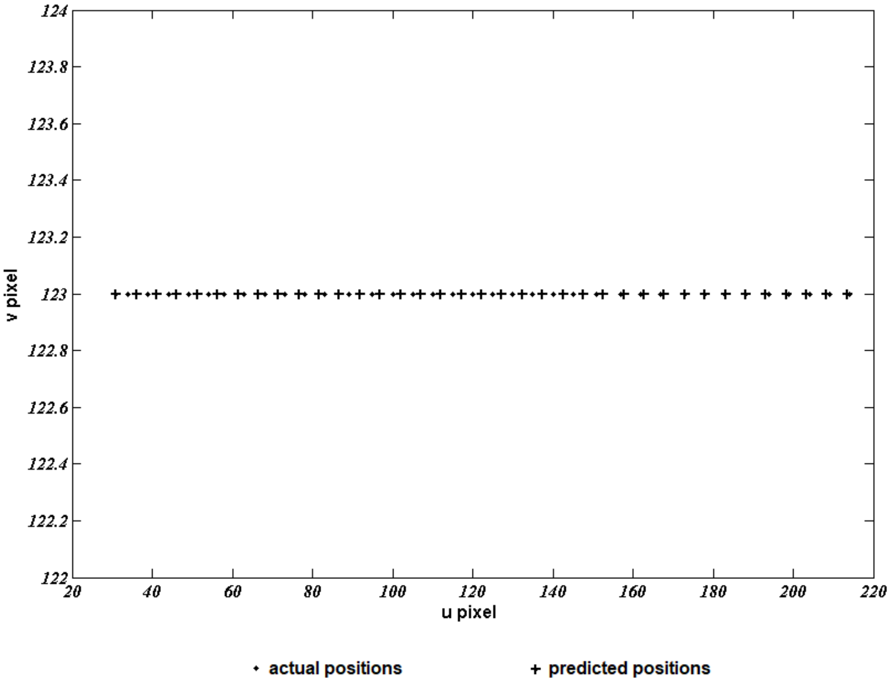


Figure 6. The predicted and actual path

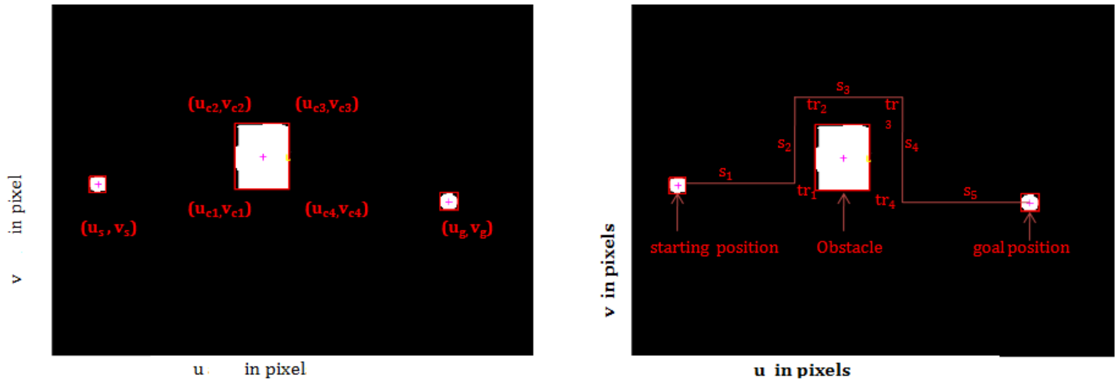
Table 1. Error in distance measurement for straight path

Distance in pixels	Distance in m calculated using inverse mapping (M_{inv})	Distance actual measured in meters	Error in calculated and measured distance meters	Percentage error
183	1.2687	1,300	0.0313	2.41

4.1.2. Straight path with static obstacle

To test the performance of an AGV on a straight path with a static obstacle, a red-colored rectangular object is positioned between the starting position and the end position Upon system initialization, the first image captured by the camera undergoes processing using an image processing algorithm to estimate the positions of the starting point, end point, and obstacle in the image coordinate system. In addition to the centroids of these positions, the four corner positions of the rectangular color marker (obstacle) are also necessary, as illustrated in Figures 7(a) and 6 (b).

All these positions are initially in pixels and are then transformed into world coordinates using the inverse mapping matrix (M_{inv}). The distance between each segment and the actuation time required for the AGV to cover that segment is calculated. The total path is divided into different segments as depicted in Figure 7(b), comprising straight segments and arcs of radius ‘r’. The total actuation time is the sum of the times required for the AGV to cover the horizontal distance, navigate the curve, and traverse the vertical distance. The total actuation time required for AGV to cover all segments and reach the goal point while avoiding the obstacle is calculated using (10) and (11). Subsequently, the AGV is actuated for the calculated time, and the resulting positions and paths achieved in real-time are presented in Figure 8.



(a) (b)

Figure 7. Binary image of the AGV (a) the initial, goal, and corner positions of obstacle and (b) the total path and segmented paths with turning points

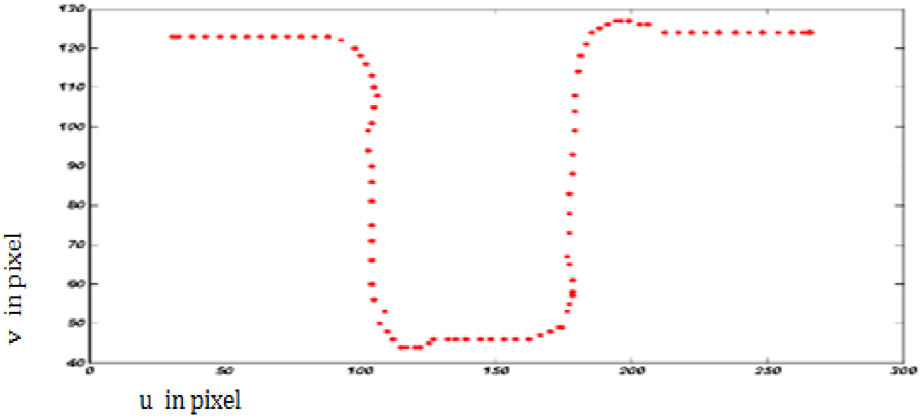


Figure 8. Positions and path achieved by the AGV in real time application for straight path with static obstacle

5. CONCLUSION

The primary objective of performing the experimentation was to utilize a single off-board fixed ceiling camera to determine the position of the AGV, end position, and obstacles in the mapped environment and then to control the position and path of the AGV by avoiding obstacles along a straight path. To achieve the objectives, the work was divided into different areas including hardware, software, environment mapping using the camera, object recognition, and position and path control of AGV on the paths that are planned by the off-board camera.

The experiment was successfully performed with the given constraints and the final conclusions are as follows. First, the efficiency of colored marker detection and recognition was recorded at 100% for defined thresholds and lighting conditions. The constant-speed AGV was tested on paths such as straight paths with and without static obstacles. Second, the developed algorithm generated vehicle paths according to the node positions detected by the camera using image processing, both in the image plane and the world plane. The total path was segmented, and segment-wise actuation times were calculated in seconds. The AGV was then navigated along the given path for the calculated actuation time. During real-time testing, the AGV successfully reached the end position while avoiding obstacles for each planned path. The average error in reaching the end position was found to be 3.94% in the world coordinate system and 3.82% in the image coordinate system, for a length of 1269 mm or 183 pixels. Third, the recorded position error is attributed to factors such as camera resolution, uncertainties in image capturing time, variations in lighting conditions, and pixels in the blob area. Fourth, in the repeatability test, the AGV could reach a given destination with an accuracy of 99%. Finally, structuring and environment mapping are indeed crucial steps before automated guided vehicle (AGV) navigation. Therefore, it is essential to invest time and resources in accurately mapping the environment before deploying the AGV solutions.

REFERENCES

[1] T.-J. Lee, D.-H. Yi, and D.-I. Cho, “A monocular vision sensor-based obstacle retention algorithm for autonomous robots,” *Sensors*, vol. 16, no. 3, Mar. 2016, doi: 10.3390/s16030311.

[2] G. N. Desouza and A. C. Kak, “Vision for mobile robot navigation: a survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002, doi: 10.1109/34.982903.

[3] J. Geun Kim, D. Hwan Kim, S. Kwun Jeong, H. Kyeong Kim, and S. Bong Kim, “Development of navigation control algorithm for AGV using D* search algorithm,” *International Journal of Science and Engineering*, vol. 4, no. 2, Jan. 2013, doi: 10.12777/ijse.4.2.34-38.

[4] I. Ohya, A. Kosaka, and A. Kak, “Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 969–978, 1998, doi: 10.1109/70.736780.

[5] G. Parekh, M. Skubic, O. Sjahputera, and J. M. Keller, “Scene matching between a map and a hand drawn sketch using spatial relations,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 4007–4012. doi: 10.1109/ROBOT.2007.364094.

[6] H. Hu and D. Gu, “Landmark-based navigation of industrial mobile robots,” *Industrial Robot: An International Journal*, vol. 27, no. 6, pp. 458–467, Dec. 2000, doi: 10.1108/01439910010378879.

[7] S. Panzieri, F. Pascucci, and R. Setola, “Simultaneous localization and map building algorithm for real-time applications,” *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 457–462, 2005, doi: 10.3182/20050703-6-CZ-1902.01346.

[8] V. Bonato, M. M. Fernandes, and E. Marques, “A smart camera with gesture recognition and SLAM capabilities for mobile robots,” *International Journal of Electronics*, vol. 93, no. 6, pp. 385–401, Jun. 2006, doi: 10.1080/00207210600565465.

[9] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: real-time single camera SLAM,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007, doi: 10.1109/TPAMI.2007.1049.

[10] J.-H. Shim and Y.-I. Cho, “A mobile robot localization using external surveillance cameras at indoor,” *Procedia Computer Science*, vol. 56, pp. 502–507, 2015, doi: 10.1016/j.procs.2015.07.242.

[11] T. J. Chong, X. J. Tang, C. H. Leng, M. Yogeswaran, O. E. Ng, and Y. Z. Chong, “Sensor technologies and simultaneous localization and mapping (SLAM),” *Procedia Computer Science*, vol. 76, pp. 174–179, 2015, doi: 10.1016/j.procs.2015.12.336.

[12] D. C. de Andrade, L. G. Trabasso, C. C. A. Eguti, and R. Suterio, “A robust methodology for outdoor optical mark recognition,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 39, no. 8, pp. 3103–3120, Aug. 2017, doi: 10.1007/s40430-017-0807-8.

[13] I. Kostavelis and A. Gasteratos, “Semantic mapping for mobile robotics tasks: a survey,” *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, Apr. 2015, doi: 10.1016/j.robot.2014.12.006.

[14] T. Gevers and A. W. M. Smeulders, “Color-based object recognition,” *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, Mar. 1999, doi: 10.1016/S0031-3203(98)00036-3.

[15] J. Lee, C.-H. Hyun, and M. Park, “A vision-based automated guided vehicle system with marker recognition for indoor use,” *Sensors*, vol. 13, no. 8, pp. 10052–10073, Aug. 2013, doi: 10.3390/s130810052.

[16] B. Kotze and G. Jordaan, “Investigation of Matlab® as platform in navigation and control of an automatic guided vehicle utilising an omnivision sensor,” *Sensors*, vol. 14, no. 9, pp. 15669–15686, Aug. 2014, doi: 10.3390/s140915669.

[17] C.-H. Hsieh, M.-L. Wang, L.-W. Kao, and H.-Y. Lin, “Mobile robot localization and path planning using an omnidirectional camera and infrared sensors,” in *2009 IEEE International Conference on Systems, Man and Cybernetics*, Oct. 2009, pp. 1947–1952. doi: 10.1109/ICSMC.2009.5345993.

[18] X. W. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, “Recognition of traffic signs based on their colour and shape features extracted using human vision models,” *Journal of Visual Communication and Image Representation*, vol. 17, no. 4, pp. 675–685, Aug. 2006, doi: 10.1016/j.jvcir.2005.10.003.

[19] P. Ranganathan, J. B. Hayet, M. Devy, S. Hutchinson, and F. Lerasle, “Topological navigation and qualitative localization for indoor environment using multi-sensory perception,” *Robotics and Autonomous Systems*, vol. 41, no. 2–3, pp. 137–144, Nov. 2002, doi: 10.1016/S0921-8890(02)00276-2.

[20] D. Puppim de Oliveira, W. Pereira Neves dos Reis, and O. Morandin Junior, “A qualitative analysis of a USB camera for AGV control,” *Sensors*, vol. 19, no. 19, Sep. 2019, doi: 10.3390/s19194111.

[21] P. Ong, W. K. S. Tan, and E. S. Low, “Vision-based path detection of an automated guided vehicle using flower pollination algorithm,” *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 2263–2274, Jun. 2021, doi: 10.1016/j.asej.2020.09.018.

[22] Q. Zhu, Z. Zheng, C. Wang, and Y. Lu, “Research on AGV path tracking method based on global vision and reinforcement learning,” *Science Progress*, vol. 106, no. 3, Jul. 2023, doi: 10.1177/00368504231188854.

[23] I. Kubasakova, J. Kubanova, D. Benco, and D. Kadlecová, “Implementation of automated guided vehicles for the automation of selected processes and elimination of collisions between handling equipment and humans in the warehouse,” *Sensors*, vol. 24, no. 3, Feb. 2024, doi: 10.3390/s24031029.

[24] K. Wilson, “Real- time tracking for multiple objects based on implementation of RGB color space in video,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 4, pp. 331–338, Apr. 2016, doi: 10.14257/ijsip.2016.9.4.29.

[25] L. H. Wurm, G. E. Legge, L. M. Isenberg, and A. Luebker, “Color improves object recognition in normal and low vision.,” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 19, no. 4, pp. 899–911, 1993, doi: 10.1037/0096-1523.19.4.899.

BIOGRAPHIES OF AUTHORS

Ramesh Pungle is working as an associate professor and head of the Department of Mechanical and Automation Engineering at P. E. S. College of Engineering, Chhatrapati Sambhajinagar (M.S.)-431002, India. He received a Ph.D. in 2018 from Visvesvaraya National Institute of Technology, Nagpur. His area of research is robotics and automation. He has 31 years of experience in industry and teaching. He can be contacted at rameshgpungle@gmail.com.

Atul Andhare is working as a professor in Mechanical Engineering and also head of the Department of Mechanical Engineering at Visvesvaraya National Institute of Technology, Nagpur. He has received a Ph.D. from the Indian Institute of Technology, Bombay, Mumbai. He has more than 35 years of industrial and academic experience in reputable institutes. His areas of interest are machine design, machinery vibrations, and condition monitoring and machining. He has published numbers of articles in various journals, guided 9 Ph.D. students and more than 30 dissertations. He can be contacted at abandhare@gmail.com.

Durva Pungle is pursuing an M.Tech. (second year) in Computer Engineering at SCTR’s Pune Institute of Computer Technology (PICT), Pune, Maharashtra, India. Her research area is hand-written text recognition and data analysis. She completed her graduation in computer engineering from Progressive Education Society’s Modern College of Engineering, Pune in 2022. She can be contacted at durvapungle@gmail.com.