Apply 2D Barcode Scanner for Mobile Robot Navigation in Checkerboard Mapping

Chun-Chi Lai
Bachelor Program in Interdisciplinary Studies, National Yunlin University of Science and Technology, 123 University Road, Section 3,
Douliou, Yunlin 64002, Taiwan, R.O.C.

Kuo-Lan SU
Department of Electrical Engineering, National Yunlin University of Science and Technology, 123 University Road, Section 3,
Douliou, Yunlin 64002, Taiwan, R.O.C.

Chia-Jen Lin
TECO Electric & Machinery Co., Ltd., 4F, No.3, Park St.,
Nan-Kang, Taipei 11503, Taiwan R.O.C.
E-mail: cclai@yuntech.edu.tw, jeromy@teco.com.tw

Evgeni Magid
Laboratory of Intelligent Robotic Systems (LIRS), Intelligent Robotics Department, Higher Institute for Information Technology and Intelligent Systems, Kazan Federal University, Russia

Abstract

In this work, a mobile robot is equipped with an industrial barcode scanner which can provide the pose information respect to the barcode tag in the field of view (FOV). For real multiple automated guided vehicle (AGV) transportation applications, the mobile robot navigation flow is considered to get the global checkerboard type path planning from a remote master server as an input. For the local planner, each robot is applied with a simple path controller to track the global path. The simulation and experimental results show that this implementation has good feasibility for multi robot co-working in a factory area.

Keywords: Automatic Guided Vehicle, 2D Barcode Code Navigation.

1. Introduction

For a traditional automated guided vehicle (AGV) control system that usually follows a magnetic tape for routine tasks. At a later time, two-dimensional barcode tags were researched as a popular AGV positioning solution after Amazon acquired Kiva Systems [1] which was the most notable commercial success of AGV fleet management system. The Kiva robots are operated within in a zone where the 2D barcode tags were affixed to the ground as checkerboard map. Without driving over the 2D barcode tags, the proximity of the 2D barcode tag can be detected and identified via the image processing system that allow Kiva system to get high quality positioning information.

B. Dzodzo et al. [2] had applied the 2D bar codes mounted on ceiling for requirements in facilities where floors and walls are in constant damaging contact with equipment and personnel. Also, there were many researches and commercial approaches which placed the 2D codes on the walls, ceilings as well as floors [3], [4], [5], [6].

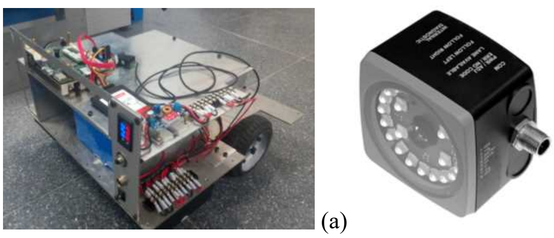
In this work, a mobile robot is equipped with an industrial barcode scanner which can directly provide the pose information respect to the tag in the field of view (FOV). For multiple AGV transportation applications, the navigation flow is considered to get the global checkerboard type path planning from a remote master server as an input. For the local planner, each robot is applied with a path controller to track the global path. The simulation and experimental results show that this implementation has good feasibility for multi robot co-working.

2. System Description

Fig. 1 (a) shows a differential wheeled robot platform that was built in our laboratory. The movement of differential wheeled robot is based on two separately servo driven wheels placed on either side of the robot. The differential wheeled robot platform is the most popular type in AGV applied. Due to it can thus change its direction by varying the relative rate of rotation of its wheels and hence does not require an additional steering motion.

Fig. 1 (b) shows an industrial optical scanner [7]. This device will detect 2D barcode tags which are typically glued onto the floor in a grid as shown in the left of Fig. 2. The individual 2D barcode tags are numbered consecutively and include position information. It can be used together with a colored tap affixed to the floor and code tags printed with 2D barcode.

In the right of Fig. 2 shows the 2D barcode tags of AGV positioning system. A 2D barcode tag contains position information in addition to a specific number. A cross in the center of the 2D barcode tag marks the zero point. The X and the Y axes are marked starting from the zero point. The black arrow indicates the positive axis and the white arrow indicates the negative axis. So the read head reports the position of the AGV in relation to the zero point of the 2D tag to the motion controller.



(a) (b)

Fig. 1. (a) a differential wheeled robot platform equipped with an industrial PC (b) the optical reading head (PGV 100 serial) for barcode tag identification

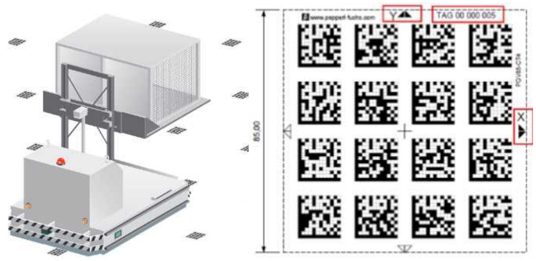


Fig. 2. Automated guided vehicle with 2D barcode tag

3. Frame Transformation in ROS

In this section we take an overview of Robot Operating System (ROS). Base on ROS framework and utilities, we can integrate the control and positioning system quickly. Besides, it is necessary to induce the coordinate transform between tag in map (world coordinate) and the odometry coordinate. So the transform tree in ROS will be completed for AGV navigation.

3.1 *Robot Operating System*

Robot Operating System (ROS) is a software frame work for robotics research and development. It applies a peer-to-peer topology for communication between robot processes and it provides tools for robot software development. For more information about ROS, the online wiki [8] is available. For the sake of completeness, some common concepts are listed as below:

- Nodes are ROS processes that perform computation. Nodes can communicate with each other using messages.
- Topics are named medium over which nodes exchange messages. Multiple nodes can publish / subscribe to a predefined topic.
- Subscriber is wrapped in a node which listens to the messages that are published to a topic.
- Publisher is wrapped in node which sends to a topic for other nodes can subscribe.
- roscore is a collection of nodes that are pre-requisites of a ROS-based system. roscore starts a ROS master node. So we must have a roscore running in order for nodes to communicate.
- TF in ROS is a package that lets the user keep track of multiple coordinate frames over time. TF maintains the relationship between coordinate frames in a tree structure buffered in time, and lets

the user transform points, vectors, etc., between any two coordinate frames at any desired point in time.

3.2 *Coordinate Transform from Barcode Scanner*

Fig. 3 shows the TF tree relationship in this work. The barcode scanner is attached in the robot with a pose transformation respect to the robot center (base_link). In general, the odometry will accumulate the encoder pulses from wheels and transfer it as displacement and rotation from the robot center respect the initial zero odometry coordinate frame. If there is no wheel drift error, the odometry coordinate frame will be the same as the world map coordinate frame. However in real situation, the wheel drift will accumulate to increase the positioning error. So we need the barcode scanner to get the real world's position, and re-align the drift between the odometry and world map.

In Fig. 4, the cross m' presents the world's coordinate center and the cross o' presents the odometry's coordinate center. The blue arrow presents the AGV's pose {xo, yo, Φo} in odometry coordinate frame. When the barcode scanner attached in the AGV that identifies the barcode tag on the floor, we will convert it as global position information such as {xm, ym, Φm}. So we can calculate the coordinate drift {xd, yd, α} between the world map and odometry frames.

$$R = \sqrt{xo^2 + yo^2} \quad (1)$$
$$\beta = \tan^{-1}\frac{yo}{xo} \quad (2)$$
$$\alpha = \Phi m\text{-}\Phi o \quad (3)$$
$$xa = R\cos(\beta + \alpha) \quad (4)$$
$$ya = R\sin(\beta + \alpha) \quad (5)$$
$$xd = xm - xa \quad (6)$$
$$yd = yd - ym \quad (7)$$

From the above equation (3), (6) and (7) the coordinate drift {xd, yd, α} will be calculated if we have the input {xo, yo, Φo} and {xm, ym, Φm}. In ROS framework, we need to create a transfer node to publish the /tf topic between odomety and world map for the TF tree's completion such as shown in Fig. 3.

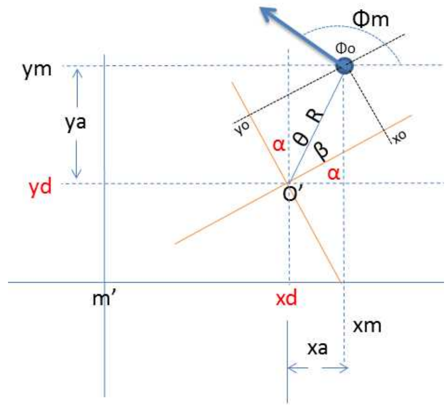


Fig. 4. Coordinate Transform between world (map) and odometry.

4. Action Server Mechanism for Navigation

In ROS based system, if we would like to send a request to a node to execute a task, and also receive a reply to the request. This can currently be achieved via ROS actionlib package. Fig. 5 shows the Action Server node that is designed in this work. The global path receiver is a template function to subscribe the goal messages from the Action Client. The goal would be a pose stamped array message that contains the path information for robot sequential moving in the world. When Action Server has received the goal, the local path follower will calculate the motion command and publish the cmd_vel topic. And another servo control node will subscribe the topic for handling AGV to follow the global path. Also the Action Server will give the feedback as the state reply to tell the Action Client about the incremental progress of the goal. For moving the platform, this will be the robot's current pose along the path. Finally, a result is sent from the Action Server to the Action Client upon completion of the goal. This is different than feedback, since it is sent exactly once.

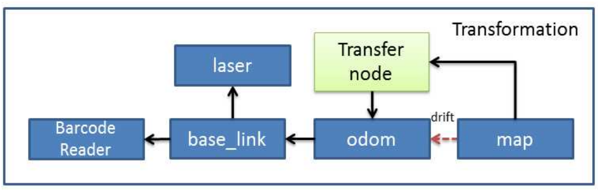


Fig. 3. Coordinate transformation (TF) in ROS.

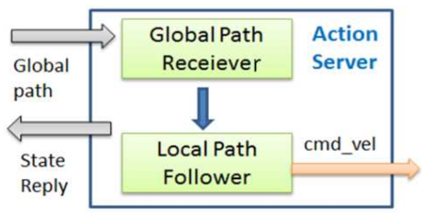


Fig. 5. An Action Server framework for AGV navigation

5. Fleet Management Framework

A fleet management system [9] primarily concerns itself with managing a group of vehicles to meet the goals and objectives obtained from an enterprise computer system. In this paper, we primarily implement a simplified fleet management system using ROS multi master system. To implement a multi master system, a package called multimaster_fkie [10] is needed and can be easily installed. Fig. 6 shows the simple fleet management framework in this study. The major task of the path planning server is to handle all the AGV's path planning and send the path message via wireless network to each AGV.

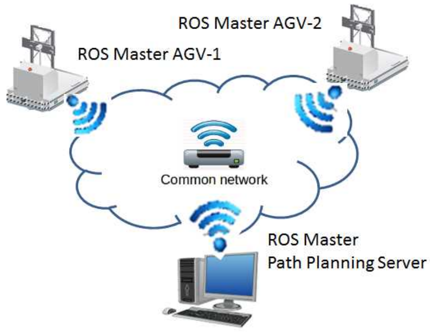


Fig. 6. Fleet management framework

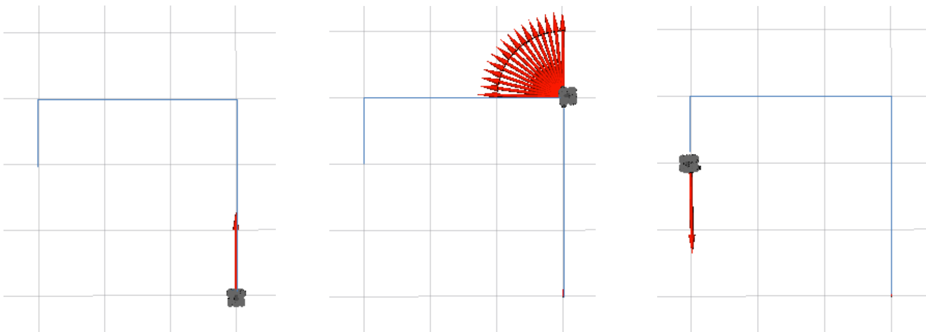
6. Experiment Results and Conclusions

The experiment is firstly simulated and demonstrated in ROS rviz user interface for verifying the system integration. For convenient, a robot platform model named (turtlebot-waffle) is pre-loaded as shown in Fig. 7 (a) and the blue line presents the global path which was published from the remote ROS master. Fig. 7 (b) shows the Action Server was handling robot's orientation for global path following. Fig. 7 (c) shows that robot had reached the goal.

In this work, we have demonstrated the methodology and integration the mobile robot navigation in checkerboard mapping based on the 2D barcode scanner. The direction for future work therefore includes optimal management strategy to improve the efficient when numerous robots are co-working in the same time.

7. References

1. Kiva system overview. http://www.kivasystems.com/solutions/system-overview
2. B. Dzodzo et al., Realtime 2D code based localization for indoor robot navigation, in *IEEE International Conference on Robotics and Biomimetics* (ROBIO), 2013. p. 486-492.
3. J. Huh et al., Mobile robot exploration in indoor environment using topological structure with invisible barcodes, in *ETRI journal* , 2007, 29(2), p.189–200.
4. S. Zhao et al., Robot autonomous localization based on single artificial landmark, in *Journal of Shandong University (Engineering Science)*, 2007, 37.4, p.39-44.
5. G. Lin, X. Chen, A Robot Indoor Position and Orientation Method based on 2D Barcode Landmark. *JCP*, 2011, 6.6, p.1191-1197.
6. H. Kobayashi, A new proposal for self-localization of mobile robot by self-contained 2d barcode landmark, in Proceedings of *IEEE SICE Annual Conference,* 2012, p. x2080-2083.
7. Position Guided Vision (PGV): https://www.pepperl-fuchs.com/global/en/
8. Robot Operation System (ROS): http://wiki.ros.org/
9. A. Singhal et al., Managing a fleet of autonomous mobile robots (AMR) using cloud robotics platformm. In *IEEE European Conference on Mobile Robots (ECMR)*, 2017, p. 1-6.
10. Juan et al., Multi-master ros systems, *Institut de Robotics and Industrial Informatics*, 2015.



(a) robot received the global path (b) robot turned in the check point (c) robot reached the goal

Fig. 7. The robot executed the path following task which was published from remote server.