Submitted on May 10, 2018 to the Notre Dame Journal of Formal Logic
Volume ??, Number ??,

Augmented Reality Application on Robot Trajectory with ROS system

Abstract There is a growing need for research on human-robot collaboration. Augmented Reality (AR) is a technology for overlaying three-dimensional virtual graphics onto the users view of the real world. It also allows for real time interaction with these virtual graphics, enabling a user to reach into the augmented world and manipulate it directly. Augmented Reality could be used to overlay virtual imagery on a real robot and so display the internal state and intentions of the robot. Thus AR can bridge the divide between human and robot systems and could enable effective human-robot collaboration.
In this project, we propose using augmented reality to visualize the TurtleBot's planned paths and real time trajectory. The architecture consists three components. First, the TurtleBot creates planed paths by user defined navigation goals prior to execution of the navigation tasks. As the TurtleBot navigates towards the goals, the planed paths and real-time locations against the TurtleBot's map are generated and stored in standard files. Second, a real-time communication system between TurtleBot and server is responsible for transferring the paths and locations information from the TurtleBot to the server side. The last, an AR application is running at server side, displaying the planned paths in green dash line and real-time trajectory in red dash line. We succeeded in visualizing the navigation goals and localization tasks by this augmented scene using multiple markers. Our project demonstrates the great potential of AR in facilitating human-robot interaction.

1 Introduction

The robotic technologies has proceeded with progress for years, from single robot to multi-robot design, from basic mechanical arm to humanoid robots, from basic dialogue to human-robot movement reaction, etc. In the concept of the future world, the city corners will gradually filled with human, blend with more AI products. How to collaborate human and AI products turns out to be an important issue called: *Human-Robot-Collaboration* research.

Establishing a human-robot interaction need sharing of visionary in surrounding. Try to think a robot want to share a door with human in a tiny corridor, The robot

2010 Mathematics Subject Classification: Primary X001, Y002; Secondary Z003
Keywords: typesetting, guide for authors

will yield human to avoid collision. planning makes robots follow the rules, why not acknowledging the robots intention before the robot passing by this place?

The *Augmented Reality* is a method of projection that enable human to know, even operate the routine planning in the real world. AR technique provide 3D graphics interaction on the screen to show the occupied routine regarding to a certain place.

In this project, we built an application using Augmented Reality toolkit, with which we showed the navigation trajectory of the robot towards goals. We succeeded in visualizing the navigation goals and localization tasks by the augmented scene.

2 System Overview

2.1 Augmented Reality AR is a growing and promising area. As AR system generates a composite view for the user by combining the real scene viewed by the user and a virtual scene generated by a computer. Virtual information is embedded into the real world, thereby augmenting the real scene with additional information.

One of the goals in developing AR application on human-robotic application is to integrate the navigation system with an interaction mechanism that permits the human user to supervise the robot operation. This project is only the initial stage, which is to visualize the robot's navigation intention prior to the navigation tasks and visualize its movement in real-time. The problem at core is to align the real and virtual images properly . To enhance the visualization of entire process, multiple markers are in use.

After receiving the planned trajectory from ROS operating system, AR application is going to visualize the virtual image of this trajectory on the screen of Ubuntu 14.04.

2.1.1 System Configuration This platform is based on Ubuntu 14.04, the external picture was filmed by Logitech webcam, this program need to be made on the terminal. Video Configuration should be set default value as *LinuxV4L2* on mode selection of the following script:

```
$./Configure
```

2.1.2 Marker Detection AR-toolkit provide a coordinate set up system on the certain marker during the virtual object establishing work. The camera will capture each frame to identify if there are any markers on the screen, when there is one, the coordinate system will be automatically set up to the marker so that could display the route and the direction patterns, quoted parameters are put on the screen. Size-known square markers are used as a base of the coordinates frame in which Virtual Monitors are represented(Figure 1). The multi-marker detection system is provided by duplicating multi-pointer on single marker at the inside of the functions in simpleLite.c file, such as *gARTImage, which receive the graphic systems and *gARHandle, which handling the markers that we frame our cameras to, etc. These pointers a sets to realize different marker detection system.

2.1.3 Coordinate Setting ARtoolkit provides a coordinate system that enable the marker to establish a virtual platform for 3D object plotting. The 3D coordinate transplanted from the marker to the front of the caliber on camera with lossless

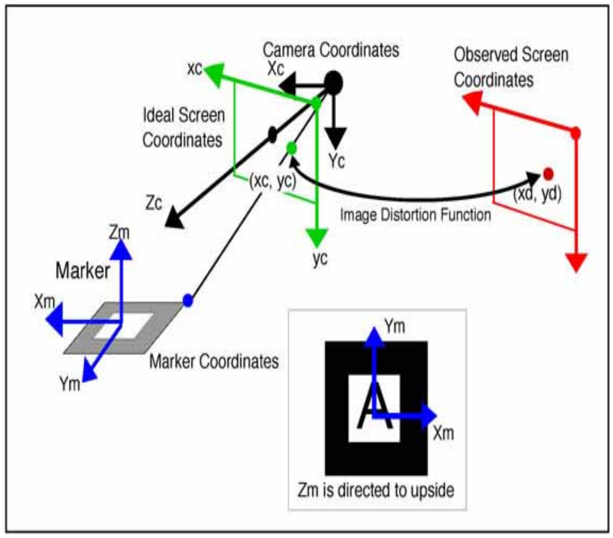

Figure 1 Coordinate transformation[1]

transformation, then the coordinate of caliber passing through a distortion function to show the 3D object in front of the screen, Figure 1 is shown as the following picture in the article.

3 Communication System

The communication between the machine which hosts the ROS system and the server is solved by setting up a password-less SSH configuration between them.

4 TurtleBot

We use TurtleBot in our project, as shown in the Figure 2. TurtleBot is an open source hardware platform and mobile base. When powered by ROS software, TurtleBot can handle vision, localization, communication and mobility. It can autonomously move anything on top of it to wherever that item needs to go, avoiding obstacles along the way.

4.1 RVIZ Mapping is the process of creating a spatial model of the environment surrounding the robot using its sensors. The map is then used for localization and navigation. With building the map, Node rviz is used for visualization of the robot localization. Figure 3 is showing the lab environment captured by rviz. During the experiment, we place the markers as shown in the green dots and set the navigation goals as shown in the red crosses.

4.2 move_base package We use *move_base* package for navigation. The *move_base* package lets you move a robot to desired positions using the navigation stack. The *move_base* node links together a global and local planner to accomplish its global navigation task. The *move_base* node may optionally perform recovery behaviors when the robot perceives itself as stuck.

Figure 2 The TurtleBot used in our project

Figure 3 Markers layout and the navigation goals design on RVIZ map: the turtlebot navigates towards the first goal and the second goal.

4.3 amcl package We use published topic *amcl_pose* to get robot current position. *amcl_pose* contains Robot's estimated pose in the map, with covariance. Localization is the problem of estimating the pose of the robot relative to a map. ROS uses the amcl package for localization. amcl is a probabilistic localization system for a robot moving in 2D, As shown in Figure 4, amcl localization information are generated and transfered to AR.

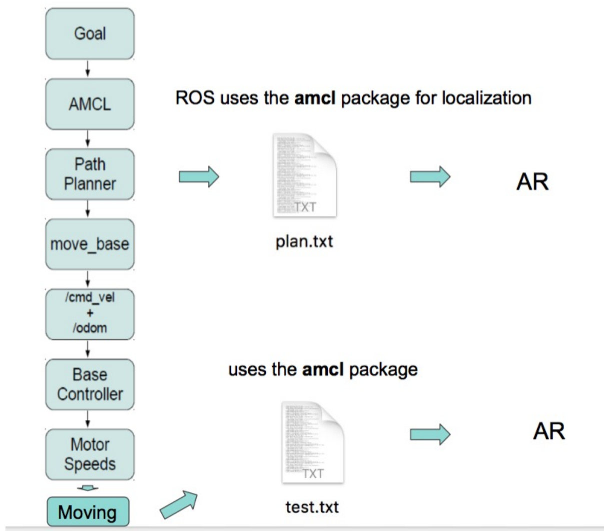

Figure 4 amcl package[5]

5 Experiment and Results

The experiment is sticking the markers on the top of floor, as shown in Figure 3. The TurtleBot has to create a planed path using amcl package, navigates from the starting point to the first goal. After reaching the first goal, it loads the second navigation goal and generates another path, and moves towards the second navigation goal. The AR tool successfully visualize the real-time movement of the TurtleBot and its internal states.

The Figure 5 displays the starting moment of navigation process: the TurtleBot is leaving its starting point. Prior to it, a planed path has been generated by the amcl package and transfered to the AR tool. On the screen of the server monitor, the dash-green line shows the planed path, and the red line shows the real-time trajectory of the TurtleBot.

The Figure 6 shows the TurtleBot reaches the first goal. We see the TurtleBot rotates a few circles before localizing itself at the goal position. The virtual image on AR tool is shown via the help of second marker.

The Figure 7 shows after TurtleBot reaching the first goal, it starts loading the second navigation goal. AR tool is visualizing the TurtleBot's intention ahead its navigation task.

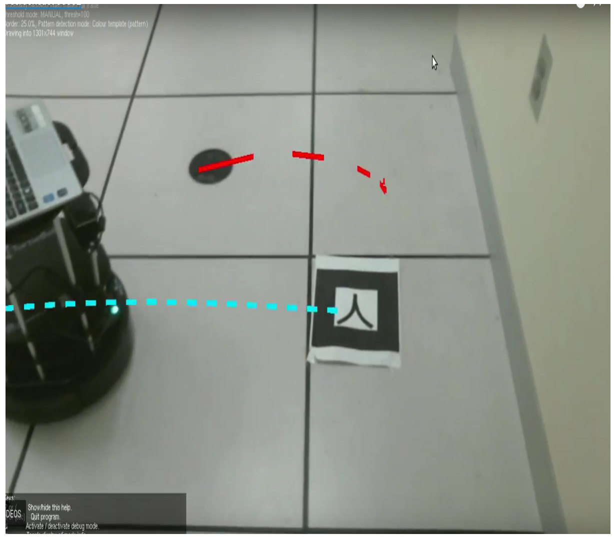

Figure 5 Starting navigation: green dash-line shows planed path and red dash-line shows real-time trajectory

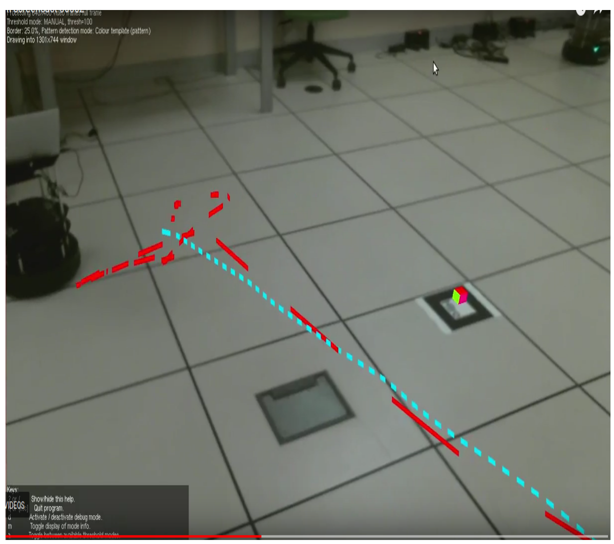

Figure 6 Reached the first goal: green dash-line shows planed path and red dash-line shows real-time trajectory

The Figure 8 shows the TurtleBot reaches the final destination goal via the help of the third marker.

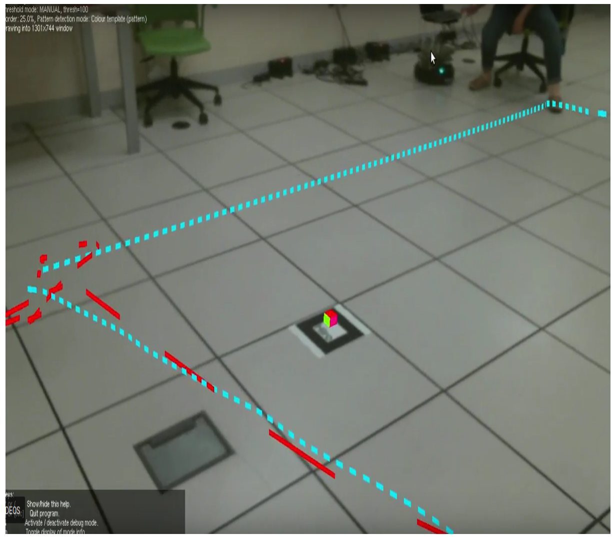

Figure 7 Load and plan the second navigation goal: green dash-line shows planed path and red dash-line shows real-time trajectory

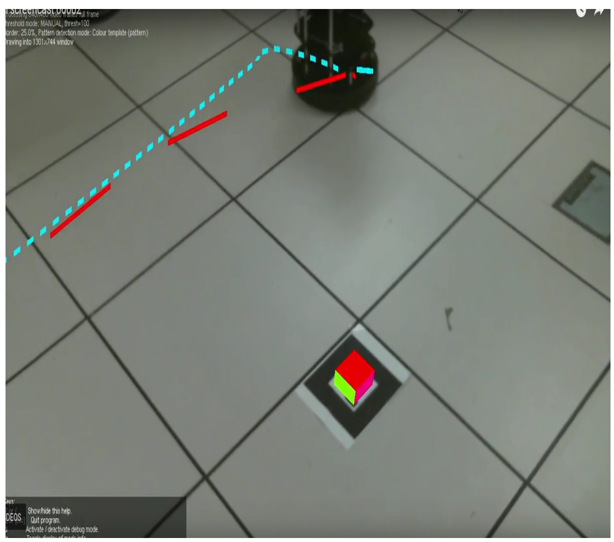

Figure 8 Reached the second navigation goal: green dash-line shows planed path and red dash-line shows real-time trajectory

6 Conclusions

In this project we have described an Augmented Reality application and the computer vision techniques used in the visualizing robot's trajectory. Our computer vision methods give good results in showing TurtleBot's planed path prior to executing the tasks and real-time trajectory. In future, we will improve this AR prototype to add more human-robot interactions components, such as user interrupts the TurtleBot's current navigation task and alters its navigation goal via AR tool.

References

[1] I. Poupyrev, H. Kato, M. Billinghurst. "ARToolkit User Manual, Version 2.33," *Human Interface Technology Lab, University of Washington, 2000*

[2] H.C. Fang, S.K. Ong n , A.Y.C. Nee., "Interactive robot trajectory planning and simulation using Augmented Reality," *Robotics and Computer-Integrated Manufacturing*, vol. 28 (2012), pp. 227–237. Zbl 1205.03055. S0736584511001116.

[3] R. Bischoff,A.Kazi "Perspectives on Augmented Reality Based Human-Robot Interaction with Industrial Robots," *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*. (IROS 2004) 4, 32 26-3231 (2004).

[4] Quigley, M, Brian Gerkey,Ken Conley,Josh Faust,Tully Foote, Jeremy Leibs,Eric Berger,Rob Wheeler,Andrew Ng. "ROS: an open-source Robot Operating System," *In ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5)*

[5] Daniel Hennes, Daniel Claes, Wim Meeussen, Karl Tuyls. "Multi-robot collision avoidance with localization uncertainty," *Proceedings of the 11th International Confer- ence on Autonomous Agents and Multiagent Systems (AA- MAS 2012)*