A Versatile Mobile Robot Platform Based on ROS

Chongben Tao, Gang Li, Guodong Liu

*1,3School of Internet of Things, Jiangnan University, PRC, tom1tao@163.com
2School of Electrical and Computer Engineering, Oklahoma State University, USA, Gangli@okstate.edu

Abstract

According to limitations of existing mobile robot platforms, this paper proposes a low-cost, modular, universal multi-purpose mobile robot experimental platform. The hardware parts of the platform are made up of iRobot Create mobile chassis, microcomputer FitPC3, Laser Range Finder (LRF), and four types of cameras; Robot Operating System (ROS) is used in software portion to complete distributed computing, wireless communications and motion control. Experiment results show that the proposed mobile robot experimental platform is a kind of qualified and promising tool for mobile robot researches.

Keywords: Mobile Robot Platforms; Multi-Purpose; ROS; iRobot Create

1. Introduction

Mobile robotics as a branch of robotics is originated from traditional manipulator robotics, artificial intelligence and machine vision in cognizing large-scale space related issues [1]. In order to cognize large-scale unknown environments and intelligently act in such environments, which means that mobile robots need capabilities of feature recognition, model checking, learning from experience, location, mapping, navigation and other functions. More importantly, accomplish these tasks in real time [2].

With the continuous progress of computer hardware technology, computers have been able to handle algorithms which have relatively high computational complexity. Especially with the advent of high performance sensors, like Laser Range Finder (LRF) and high resolution cameras, which make mobile robots more and more powerful. Many companies have already developed a variety of different functions of mobile robot experimental platforms. One of the famous mobile robots is PR2 humanoid robot, which has seven degrees of freedom, two mechanical arms each loading 1.8kg, a 5 million pixels camera, a tilting laser range finder and an inertial measurement unit [3]. Series of Pioneer mobile robots are another kind of well known mobile robots, which equipped with laser range finder and cameras. But they are custom-build robots and relatively expensive [4].

2. Mobile robots design philosophy

Through in-depth study on a large numbers types of mobile robot platform, this paper proposes seven design standards about creating a universal mobile robot platform:

• Availability: It should have sufficient computing resources to complete tasks, such as image processing, wireless communications, autonomous navigation, and multi-robot collaboration et al.
• Reliability: it should be built on a firm mobile platform to ensure repetitive experiments;
• Compatibility: It should be compatible. Hardware of the platform can be applied in platforms using the same software. Hardware can be universal among such a variety of platforms, which means types of platforms with different functions can be created.
• Versatility: it should be multi-purpose, so that it can adapt the needs of a variety of research subjects. Hence its hardware structure should be open to the public;
• Cheap: it should be a relatively low-cost experimental platform, so most of researchers are affordable. Therefore, researchers are able to have a sufficient number of robots for their multi-robot experiments.

Compared with existing mobile robot platforms, the proposed platform is a kind of low-cost, modular and universal multi-functional mobile robot experimental platform. It meets most of the demands mentioned above, meanwhile equipped certain capabilities. At present, hardware of the platform is mostly made up of iRobot Create mobile chassis, microcomputer FitPC3, laser
range finder (LRF) and four types of replaceable cameras. The software system is built by Robot Operating System (ROS) [5], whose role is like a real operating system. In the proposed system, algorithms are executed in the form of ROS packages to achieve three basic functions including target detection and tracking, simultaneous localization and map building and autonomous navigation.

3. Hardware and software design

3.1 Hardware setup

In this paper, the proposed mobile robot experimental platform is comprised of iRobot Create mobile chassis, microcomputer FitPC3, URG-04LX laser range finder and four types of replaceable cameras. The mobile chassis of the platform is based on vacuum cleaning Robot Roomba. Hardware part of the Roomba vacuum cleaner is taken away, only keeps mobile chassis, which called iRobot Create. Motor control commands are issued through the iRobot Roomba Open Interface Protocol. Sensor data are read through its serial port. FitPC3 is a kind of fanless micro-computer. LRF is a USB-powered device which uses laser beams to determine distances between robot and objects. LRF model used here is Hokuyo URG-04LX, whose measuring range is from 20mm to 4049mm, scanning angle range of 240°, scan rate 100ms/scan, ranging accuracy of ±3% and angular resolution of 0.36°.

Four types of cameras are used in our platform, which are commonly used in computer vision. Three of the four are monocular vision cameras including Point Grey Flycapture fish eye camera, Mobotix Q24 360° panoramic camera and QuickCam webcam. Another one is a trinocular camera called Kinect depth camera. Model of the Point Grey Flycapture fish-eye camera used here is Firefly MV FMVU-03MTC, whose resolution is 640×480, frame frequency is 63fps. This kind of camera provides color images up to 1 million pixels; Q24 360° panoramic camera can offer several kinds of view, including a panoramic image. So that it can cover areas around the mobile robot. This camera can provide up to 3 million pixels color image, whose size can be scaled down from 2048×1536 to 160×120. Features of the camera (resolution, frame rate etc.) can be modified by sending a network request; Model of the QuickCam webcam is Logitech QuickCam Pro 5000, which uses a wide-angle lens and needs to manually focusing. This type of camera can provide up to 300,000 pixels, maximum resolution of color image is 640×480, maximum frame rate of 20fps; Kinect depth camera is an external device of Microsoft Xbox 360 console, which is a kind of amusement device developed by Microsoft Company. This kind of camera incorporates several technologies, including RGB imaging, 3D imaging, audio processing and motor control. Our mobile platform only uses camera part function, which developed by Israel PrimeSense Company based on RGB camera and depth sensor. This device connects to a computer via USB port. The effective sensing range is from 0.4 meters to 4 meters, vertical viewing angle range is ±43°, horizontal range of ±57°, frame rate of 30fps. In addition to major components mentioned above, external battery supply power for FitPC3 computer and camera. USB-powered mini-fan is used to cooling for FitPC3.

A plexiglass support frame has been built on the upper part of iRobot Create mobile chassis, which is used to place control components, wireless communication components, sensors and types of cameras. The support set has four layers. Its ground layer directly connected to mobile chassis, another three layers fixed together by screws from the top to the bottom. FitPC3, LRF, cameras and mini cooling fan are fastening on the support set. According to different types of camera, there are four versions of design, as Figure.1 shown. The bottom and second layer are prepared for FitPC3. Sometimes we need to use two FitPC3s. For the third and top layer, we usually put LRF on the third layer, and put camera on the top layer. In addition, equipments on these two layers can be exchanged as needed. Plexiglass support frame and mobile chassis are combined firm enough to support all components placed on it. The hardware design mentioned above makes the system structure compact and firm, and as plexiglasses are lightweight, which will not affect moving of the robot. Moreover, more plexiglass baffles can be fixed on the frame to put more needed equipments. And multiple mobile robots network can be set up based on the platform, as Figure.2 shown.
3.2 Software setup

Robot Operating System (ROS) is an open source meta-operating system, including low-level device control, wireless communications and software package management. According to the form of software packages, ROS can effectively create and manage programs. Executable files are executed by Nodes [5]. Because of the distributed computing characteristics of ROS, multiple mobile robots can be easily connected to the network system.

A program can be divided into different nodes (Nodes) in ROS. These nodes can be running on different computers in the same network. A node is an independent process, which can receive and publish Topics from other nodes. A part of the hardware can be regarded as a Node, and a data processing algorithm also can be used as a node, so long as all the Nodes working in the same “ROS MASTER”. The whole algorithm can be successfully achieved in such a distributed mode.

There are two kinds of nodes in our ROS. One kind is hardware drivers including iRobot driver node and LRF driver node. Another kind is functional procedures including Object Detection node, Object Follower node, Integrated Exploration node and FastSLAM GMmapping node, as shown in Figure.3. iRobot Create mobile chassis driver subroutine called irobot_create_2_1 package, developed by Brown University; LRF driver subroutine called Hokuyo_node which comes from ROS community. Target detection and tracking subroutine are written by the author; Integrated Exploration subroutine are written by the author expect FastSLAM GMmapping package which is used for robot localization in this part; FastSLAM GMmapping package and navigation package are supplied ROS community which are used for autonomous navigation.
4. Functionality realization

With the help of cameras, Ad-hoc wireless network and joystick control, the proposed platform is equipped with visual control interface, as shown in Figure 4. In order to provide convenience for research community, this platform is equipped with three novel and useful functions, which are target detection and tracking, integrated exploration and autonomous navigation are achieved in the platform.

4.1 Target detection and tracking

The specific task of target detection and tracking is that the mobile robot platform automatically navigates in an unknown environment, identify and locate a moving object, then tracking it. A fast and less calculation algorithm is proposed to detect moving orange T-shirts using color segmentation. The goal of color segmentation is in order to find out continuous regions, where each pixel has common characteristics. T-shirt has a unique color, so this is a relatively simple and rapid target detection method. As RGB color space is very sensitive to changes of lights, captured images firstly need to transform from RGB color space to HSV color space [6]. Let (R, G, B) is red, green, and blue coordinates of a color, respectively, whose value is a real number among 0-1. Set MAX and MIN is equivalent to the maximum value and minimum value among R, G and B, respectively. In order to find out (H, S, V) values in the HSV color space, where $H \in [0, 360)$ is the angle of hue, S, V $\in [0, 1]$ is the saturation and brightness, respectively. Calculated as:

$$
H = \begin{cases} 
0^\circ, & \text{if } MAX = MIN \\
60^\circ \times \frac{G - B}{MAX - MIN} + 0^\circ, & \text{if } MAX = R \text{ and } G \geq B \\
60^\circ \times \frac{G - B}{MAX - MIN} + 360^\circ, & \text{if } MAX = R \text{ and } G < B \\
60^\circ \times \frac{G - B}{MAX - MIN} + 120^\circ, & \text{if } MAX = G \\
60^\circ \times \frac{G - B}{MAX - MIN} + 240^\circ, & \text{if } MAX = B 
\end{cases}
$$

(1)

$$
S = \begin{cases} 
0, & \text{if } MAX = 0 \\
\frac{MAX - MIN}{MAX}, & \text{otherwise} 
\end{cases}
$$

(2)

$$
V = MAX
$$

(3)

Then brightness part information was decoupled from color information in HSV color space. Finally set thresholds for each band in the HSV color space, respectively. Thus color segmentation has been fulfilled. The left binary image of Figure 5 was obtained by filtering each HSV band. After Gaussian
filters and topology methods (Dilation and erosion) were used to reduce image noises, the detected T-shirt was shown in right part of Figure 5. Then, based on the target detection algorithm mentioned above, FastSLAM GMapping package and target tracking algorithm proposed by Pushpa D. [7], the platform keeps tracking moving target detected by the proposed color segment algorithm in the unknown environment.

![Figure 5: T-shirts detected by Color segmentation](image)

### 4.2 Integrated exploration

In the field of robot autonomous navigation, integrated exploration is a synthesis of three parts, including localization, map building and motion control, whose essentially meaning is that effectively guide the mobile robot creating map and moving in one way or another in an unknown environment. In this paper, integrated exploration is divided into two steps: candidate target points creating and utility function design.

1) Candidate target points creating. In order to create candidate target points in a local map, boundary based method is used here [8]. A 2D occupied grid map created by FastSLAM GMapping packages only uses three types of grids. They are -1, 0 and 100, which respectively means unexplored space, no obstacle space and obstacles. On the basis of the map information format, the boundary based candidate points generation algorithm is divided into three steps. The first step is preliminary creating. The frontier candidate point values are 0, and a certain number of neighboring grids will be abstracted. But a number of candidate points will be created in the process. The second step is filtering. A kind of filter designed in accordance with certain standards is used to filter out obviously impossible candidate points. These standards come from some common sense of indoor environments, such as for offices or corridors. In the filtering step, the ratios of unexplored grid numbers, free grid numbers and occupied grid numbers to the overall neighborhood grid numbers are three kind of useful values that indicate the pattern of the neighborhood which decide whether the grid is a valid frontier candidate. In addition, the shape of the occupied grids can also be used to delete some error results generated in preliminary creating step due to sensor noises. Equation (4) is used to detect the correct candidate points. The final step is classification, which is important for classifying clusters of these points. Then completely classified candidate reference points are obtained. Outputs of the candidate generation process should be well-separated candidate points.

\[
\begin{align*}
a & \leq \frac{N_{\text{unexplored}}}{N_{\text{total}}} \leq b \\
c & \leq \frac{N_{\text{free}}}{N_{\text{total}}} \leq d \\
e & \leq \frac{N_{\text{occupied}}}{N_{\text{total}}} \leq f
\end{align*}
\]

In the above equations, \(a, b, c, d, e, f\) are threshold values, which can be adjusted. In addition, \(N_{\text{total}}\) also can be adjusted according to changes of the neighborhood size.

The process of office candidate point generation is shown in Figure 6. Blue stars and red rectangles respectively indicate the results obtained after filtering and classification.
Figure 6 Frontier creating process of an office: (A) raw map; (B) candidate points after preliminary creating; (C) candidate points after filtering and classification; (D) candidate point selection.

2) Utility function design. Utility function is used to select a final target frontier point in all the candidate points. The target point stands out from a number of candidate points is determined by the position of the candidate point and the robot. From the location of the candidate point, information gain and the localizability can be used. The information gain means how much unexplored grid can be seen from the candidate position, while localizability depends on how much markers can be seen from the candidate point, which will instruct the robot to reposition its angle of view. For the position of the robot platform, Rao-Blackwellized particle filter is used in the FastSLAM GMapping program to estimate the exact location of the robot [9], where the entropy will affect the moving distance estimation between the robot location and the candidate position. In the utility function $U_{total}(x)$, a shorter path is better, and it is subject to the impact of entropy from the robot position.

$$U_{total}(x) = w_I U_I(x) + w_N U_N(x) + w_L U_L(x)$$

In equation (5), $w_I$, $w_N$, and $w_L$ are the relative weights for information gain, navigation cost and localizability respectively. $w_S$ represents the scenario coefficient. For instance, offices and bedrooms have different values of $w_S$, $w_E$ means the effect of entropy for the navigation cost.

$$x^* = \arg \max_x (U_{total})$$

In equation (6), it means that the value of the $U_{total}$ will be maximized by the selected candidate.

4.3 Autonomous navigation

FastSLAM is a kind of algorithm that recursively estimates the full posterior distribution over robot pose and landmark locations, yet scales logarithmically with the number of landmarks in the map [10]. Entropy is given for the distribution of the robot pose. The higher the entropy value, the higher the uncertainty. A* algorithm is used to create a global path and cost map, meanwhile an underneath local planner is issuing speed control commands for mobile robot safely moving toward the goal [11]. The global coordinate system is used for high level planning, while the local coordinate system is used for local planning and obstacle avoidance.

If we hope mobile robots autonomous navigation in an artificial setting environment, first of all we need to manually create an environment map. Then manually set start position, the moving route and the target location in ROS rviz visualization window. Finally, run the ROS move_base startup file for autonomous navigation [5]. Four yaml files are used as parameters of the startup file to run the move_base Node. Trajectory, angle, linear speed limit, different tolerances and deviations have been set in these four yaml files. Figure 7 shows the route of the mobile robot navigation in an indoor environment.
navigation. The result shows that the proposed mobile robot platform can autonomous navigate in indoor environment.

![Figure 7](image)

**Figure 7** The proposed mobile robot platform navigation in indoor environment

### 5. Performance Evaluations

Respectively manual created a 2D occupied grid map in an artificial laboratory environment using the proposed mobile robot platform and a Pioneer 3DX robot. The artificial environment is surrounded by a rectangular wooden frame. Four shapes of obstacles are placed on the middle of the wooden frame. Both of them move at a speed of 0.1m/s. Two created maps have no great difference between each other, as shown in Figure 8. The left top part of the map created by the proposed mobile robot platform is not continuous. The reason is mainly because objects on the border reflected laser beams launched by URG-04LX LRF. While the Pioneer 3DX robot creates a smooth boundary as a more powerful SICK LRF was used.

![Figure 8](image)

**Figure 8** Artificial environment maps comparison
SLAM experiments have been completed on the third floor corridor part of a building using the proposed mobile robot platform and a Pioneer 3DX robot. The corridor is about 120 meters long, 25 meters wide. The two robot platforms were used to manual create a map at a speed of 0.3m/s respectively, as shown in Figure.9, map A was created by the Pioneer 3DX robot and map B was created by the proposed mobile robot platform. On the overall shape, maps created by the two kinds of mobile robot platforms are similar. Corridor length of the short side in map B is longer than map A, and the angle of the corner is slightly different in the two maps. These errors are caused as them use different odometers and laser range finders.

6. Conclusion

Based on the existing functionalities equipped in the proposed mobile robot platform, many research contents, such remote mobile robot control, the human-robot collaboration and multi-robot cooperation can be realized by this platform. In the future, 3D depth Kinect camera on this platform will be used to conduct 3D scanning and 6D of SLAM and intuitive human-robot interaction will be achieved based on the existing software systems. More sensors and mechanical arm will be installed the platform, then this platform can be used in more subjects, such as remote indoor monitoring system, the elderly auxiliary systems et al.

7. References
