EMARO/ARIA - ARPRO
Final Project

A Hospital Robot Assistant

Medical Robotics[1] is an organization which provides automation and robotic based solutions to health care centers. The company decided to hire you to participate in the design of a hospital assistant robot in charge of the distribution and collection of clinical materials.

I.1 Context

The hospital is an institution that cannot cease to serve for prolonged periods of time, so the project imposes as a first constraint that the solution must require of minimal adaptations in the environment.
Other aspects to consider are the security regulations of the institution. One of the most relevant is the mobility convention. According to this, the mobility must be done by the right side of the corridors and as close as possible to the walls. In order to prevent accidents during emergency displacements. Another security requirement is that any automatic system operating in the institution must prove to rely on a robust error diagnosis strategy. The system must count on a simple auditing mechanism (for legal purpose in case of failures or accidents); and be capable of real-time error report to the responsible technicians.
In relation to the mobility constraint, Medical Robotics is considering to design a mobile robot by integrating a serial manipulator to a mobile base, as depicted in Fig. 1.1.
The initial proposal for sensory hardware (it will be detailed later) is a scanning rangefinder and a camera both pointing towards the heading direction of the robot.

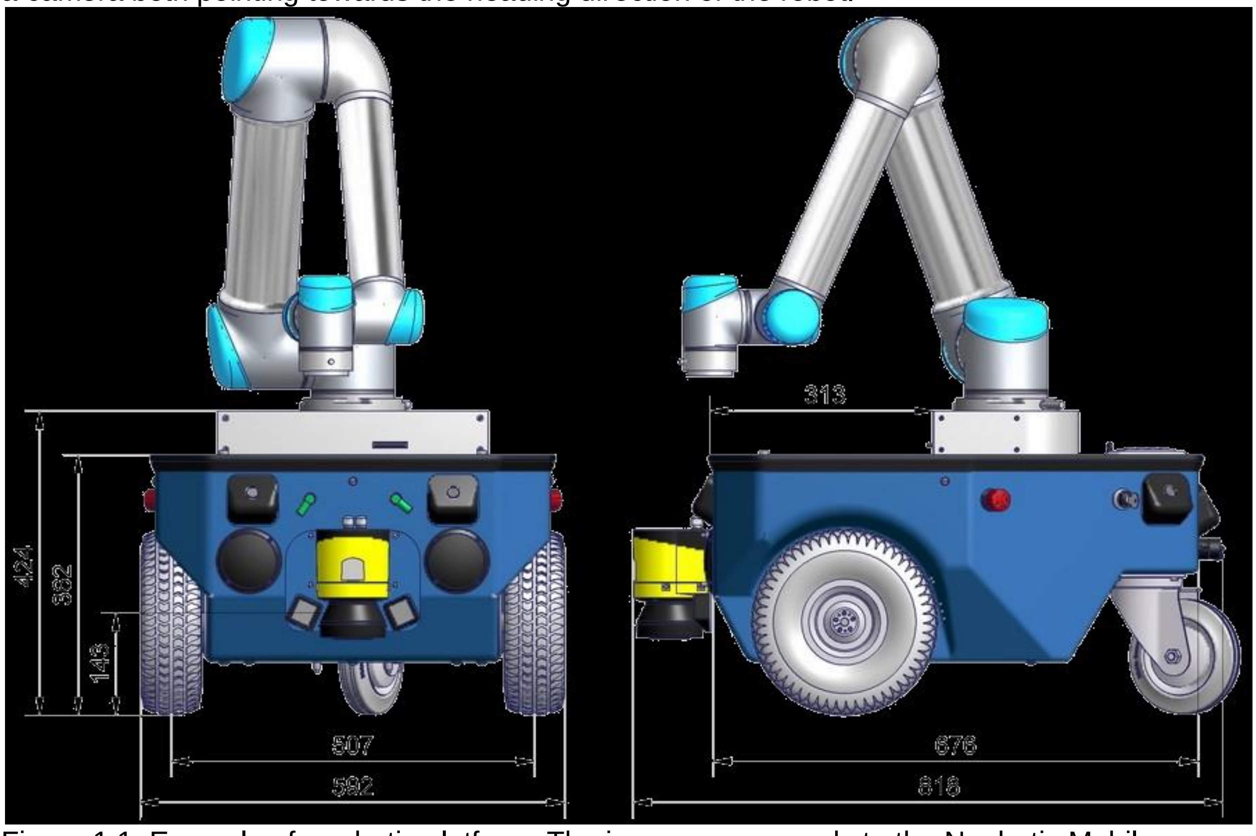

Figure 1.1: Example of a robotic platform. The image corresponds to the Neobotix Mobile

Manipulator MM-500 [1]. The dimensions are in mm.
After a meeting with the direction board of the hospital, it has been decided to adopt the strategy of adding a navigation line to the skirting board in the walls of the corridors, to help the robot to self-localize and simplify the detection of errors. All corridors must be traversed in two directions since there are doors on both sides. The robot goes through one side of the corridor and comes back in the opposite side. Figure 1.2 shows a representation of the navigation requirements.

[1]A fictional enterprise.

1

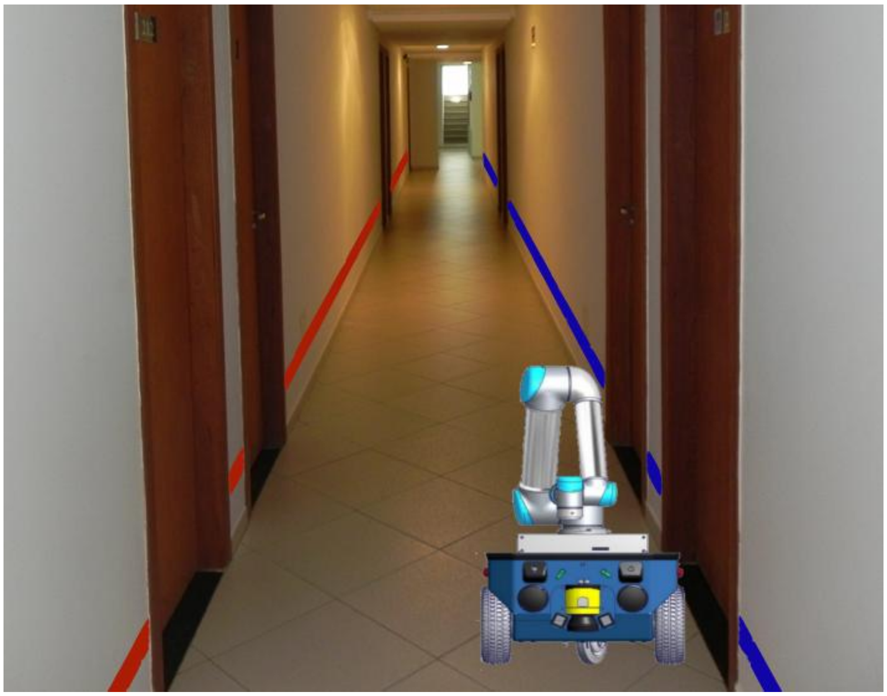

Figure 1.2: Corridor navigation. The color blue is being tracked. The robot will move until the end of the corridor keeping the right side, as close as possible to the wall.

When the robot reaches the end of the corridor, it must invert the tracked color. That is, if it is following a color and suddenly another color is perceived, it must react promptly by correcting its position. However, the color of interest must be reversed when reached the end of the corridor. The change in the system's internal state is achieved by attaching a third (neutral) color delimiting the traveling direction. Thus, the perception of a neutral color constitutes a switch that marks the beginning of the return path. Figure 1.3 illustrates the situation described. This simple strategy helps the system to identify navigation errors given to disturbances (i.e., someone hit the robot and changed its orientation). The navigation decision space is detailed in Table 1.1.

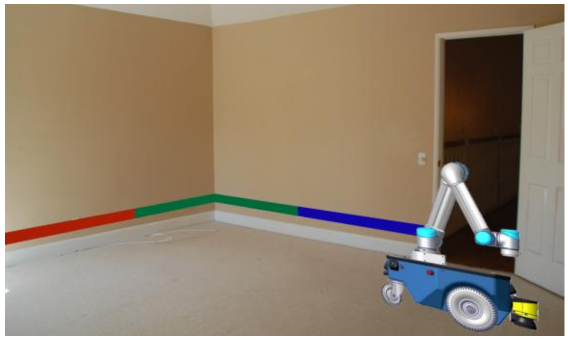

Figure 1.3: Navigation direction change. The color blue is being tracked. The color green signals a change in the direction. The robot will start to track the red color to return from when it came from; but tacking the oposite direction to keep the right side while it moves as close as possible to the wall.

Id	**Side**	**Matching**	**Change**	**Interpretation**
A	Right	Equal	-	Correct direction and side.
B	Right	Different	Yes	Correct side, direction transition.
C	Right	Different	No	Wrong navigation plan, returning before finishing.
D	Left	Equal	-	Correct side, wrong direction.
E	Left	Different	No	Correct direction, wrong side.
F	Center	-	-	Wrong orientation with respect to the wall.
G	-	-	-	No color detection, stop the system immediately.

Table 1.1: The navigation decision space. *Id* provides an identification for the case. *Side* indicates the side of the visual space where the line is detected. *Matching* indicates whether the detected color corresponds to the tracked color. *Change* indicates the detection of the neutral color. *Interpretation* presents a description of the state.

I.2 Your contribution

The project is in its initial state and you have been invited to join the development team. It has been entrusted to you, as a professional in the programming of robots, to explore a set of technologies available in the market in order to present a report on the feasibility of developing the perceptual module of the system. To perform this evaluation, you are asked to build a prototype concept testing, to assess the following elements:

- The feasibility of using computer vision in the described environment.
- The convenience of using a scanning range sensor so estimate the scene depth.
- The relevance of a set of open source libraries under the C++ programming language.

The design team has prepared some data using an analogous situation, with which you can evaluate your prototype. The data corresponds to a scene where colored cardboards are presented simultaneously to the laser scanner and the camera. Your task is to propose a prototype to identify the color of the cards, the side of the camera in which they appear, as well as their orientation. Depending on the results of this experience, the team will meet to decide on the interest of keeping the proposed configuration to develop more sophisticated tests.

I.3 Requirements

Your solution must consider the following aspects:

a) The detection algorithms (the perceptive module) should be coded as a dynamic library.

b) The principle of software engineering known as *reusability* must be considered. You are encouraged to code the minimum as possible by using existing routines to do your tests. In this way, your development time will be reduced while the possibility of bugs

in your program stay reasonably small.
c) Since this is a project developed under the methodology of incremental prototyping; the total set of events that integrate the system has not been specified yet. So your contribution should be extensible from the point of view of software engineering, with a minimum cost in terms of code modifications to consider new events in the perceptive module.
d) You must present: a) a technical report on the evolution of the time cost of the perceptual algorithm of your program; b) a log file showing the current state of the system at each iteration during the experiment; c) the documentation of the project including the description of the API (Application program interface) that you have defined; and finally, d) the source code of your program.
e) Your program must follow the testing protocol as explained next.

I.4 Data capture

The data that has been given to you was recorded by placing the camera above the scanning range finder. Every five seconds the scene was registered simultaneously in both sensors. From the measurements of the scanning range finder, you must ignore the values beyond a radius of 0.6*m*; since they correspond to the structure of the room where the experiment was recorded, and not to the colored cards presented to the system (See Fig. 1.4).

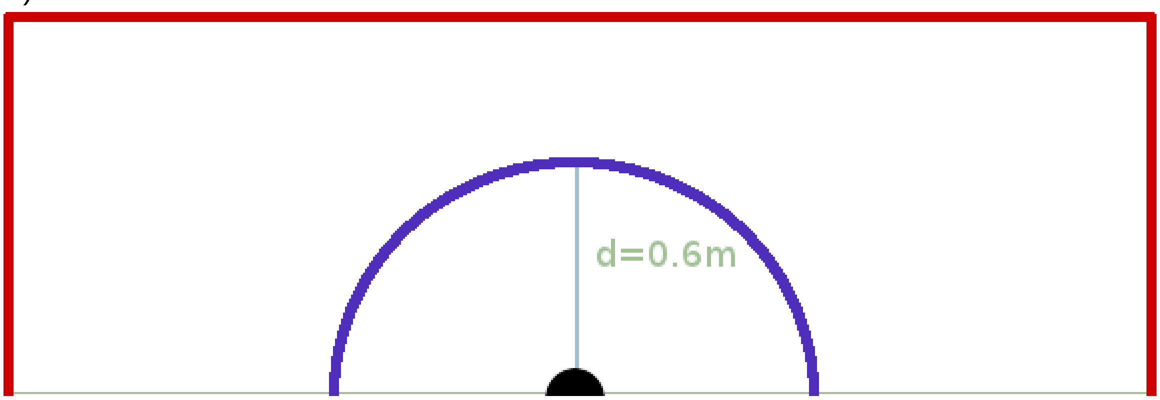

Figure 1.4: Scene capture setup seen from above. The red lines correspond to the walls of the room. The region of interest for your algorithm is located between the black region (which represents the scanning range finder and the camera location), and the blue semi-circunference (which represents the maximum distance of interest. The data vector contains 720 measurements within the range of −90_ _ _ _ 90_, with a resolution of 0.25_.

I.6 Hardware configuration

Hokuyo UTM 30 LX. The Hokuyo UTM 30 LX[2] sensor (see Fig. 1.5) is a scanning range finder suitable for robots with higher moving speed. The sensor presents a long range detection and fast response time. Though for the experiment, the detection range was reduced. Table 1.2 enumerates its technical characteristics.

Feature	Description
Long Detection range	30*m*.
Wide Angle	270_.
Environment	Outdoor.
Dimensions	*W*60 × *D*60 × *H*87*mm*.
Weight	370*g*.

Table 1.2: The Hokuyo UTM 30 LX technical specifications.

Figure 1.5: The Hokuyo UTM 30 LX

Logitech Webcam C90. The images where captured with a Logitech Webcam C90 (see Fig. 1.6) at the low resolution of 320 × 240. The camera's technical specifications are given in Table 1.3

Figure 1.6: The Logitech Webcam C90

Feature Description
Optics Carl Zeiss R with autofocus.
HD Sensor Native 2MP.
HD video capture up to 1600 × 1200.
Widescreen 720p.
Frames per second video 30.
Hi-Speed USB 2.0 certified.

Table 1.3: The Logitech Webcam C90 technical specifications.

Bibliography

[1] Neobotix. Mobile manipulator mm-500. http://www.neobotix-robots.com/mobile-manipulator-mm-500.html. Online; accessed the 1st December 2013.

[2] Hokuyo. Scanning range finder (sokuiki sensor). http://www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html. Online; accessed the 1st December 2013.

[3] opencv.org. The open source computer vision library. http://opencv.org/. Online; accessed the 1st December 2013.