RoboSantral

An autonomous mobile guide robot

A.Denker* , A. U. Dilek, B. Sarıoğlu, J. Savaş, Y.D. Gökdel
Department of Electrical and Electronics Engineering
Istanbul Bilgi University
Istanbul, Turkey
*ahmet.denker@bilgi.edu.tr

RoboSantral, An autonomous mobile robot which has been designed and realized in order to guide the visitors through a university campus, is presented in this paper. This robot accompanies guests through the campus and gives presentations on predefined locations. Location data is obtained from GPS sensors. Targets such as faculty buildings, museums etc... are recognized by the image processing of pre-defined tags. As microprocessor and microcontroller, Raspberry Pi and Arduino are used respectively.

I. INTRODUCTION

Mobile robots which came as an alternative to the stationary constraints of the industrial robots first had missions such as exploring other planets or inhospitable areas on Earth, collecting geological samples... As the costs of mobile robots come down, the fields of use are extending rapidly, including the daily use of mobile robots.

In this paper an autonomous mobile robot that is designed and developed to greet and guide the visitors of a university campus is presented. Besides halls, theatres, concert areas, libraries and museums, university campuses are preferred locations of artistic exhibitions and cultural activities. This makes university campuses a center of attraction not only for the students but also for the native and international visitors. When the required manpower for introduction and guidance of crowded visitors is considered, a mobile robot with navigation capabilities turns out to be a desirable choice.

The autonomous mobile guide robot RoboSantral was designed to fulfill this need. RoboSantral has several different components such as Arduino, Raspberry-Pi, LCD screen, sensors, speakers, motion kits… In the design, components were integrated in order to implement a self-navigating, introductive guide robot. Guidance tasks are achieved with the data from surroundings processing on RoboSantral's software. Following sections of this paper presents information on the design, implementation and the test phases of RoboSantral.

II. MECHANICAL DESIGN

There is a fundamental approach behind every successful mobile robot application. The integration of the different parts of a mobile robot has an extreme importance in design.

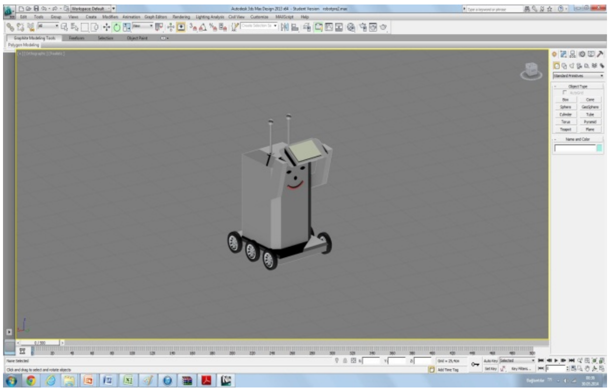

Fig.1 : A screenshot of the 3D drawing of RoboSantral

Therefore how to solve the movement problem, how to comply mechanical design with kinematics, dynamics and control considerations had to be decided during the design phase. As the initial step of the mechanical design possible appearances of the robot were considered and a humanoid look with a torso, face and arms was preferred. The LCD screen which would be used in presentations was placed upon raised arms, at eye-level.

Fig.2: A screenshot of the ultimate design of RoboSantral

978-1-4799-7800-7/15/$31.00 ©2015 IEEE

Software tools that facilitate 3D modeling and animation, such as 3ds Max and AutoCAD, were used at the stage of mechanical design of RoboSantral (Fig.1). Mechanical design comprises 3 pieces which are base, body and arms. Firstly, the base was designed and six wheels were symmetrically placed as three wheels on each side. Subsequently, the body and arms that hold the LCD screen were added. With the integration of these 3 pieces, final design was completed (Fig.2). Furthermore, 3D model was implanted in an animation in order to simulate guidance tasks of the robot on campus. For animation, campus paths, buildings, location tags, green areas were incorporated into the model. In the animation scenario, robot moves along the campus paths, when a building of interest is detected robot moves towards it and reads the information tag in front of the building. This calls for the robot to turn around, so as to facie the visitors and afterwards it starts a brief presentation about the building.

Base platform was designed considering the field circumstances. Thus, in the production of the base, steel and off-road tires are used considering rough and uneven paths (Fig.3).

After the hardware implementations and the test drive, measurements for body design were refined, body and joints are drawn taking into account of easy manufacturing and transportation then sent to manufacturing. Measurements and cuts on CNC table formed the ultimate design which is a robot in sizes 51cm x 65 cm x 85 cm. In addition to this, two antennas that are 25 cm long and 11 Inches long LCD Screen were placed on the robot.

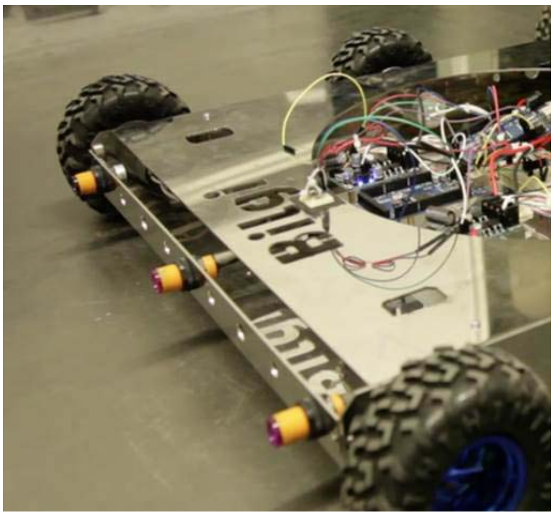

Fig.3: Base platform of RoboSantral

The variables that needs to be considered in kinematic modeling and also that take part in RoboSantral's arrival to the target are presented in Fig.4.

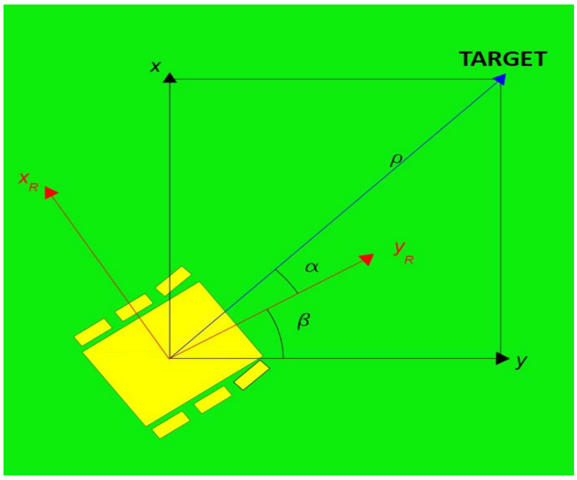

Fig.4 : Variables that take part in arrival to the target

As seen in Fig.4 x-y stands for the reference system. $\mathbf{x_R}$ and $\mathbf{y_R}$ stands for the coordinate axis that are fixed on the robot. $\boldsymbol{\rho}$ stands for the distance to the target and $\boldsymbol{\alpha}$ stands for the angle of how much robot should change its orientation. Lastly $\boldsymbol{\beta}$ stands for the orientation of the robot on x-y reference system.

As the angle $\boldsymbol{\alpha}$ goes to zero, robot reaches the direction that will lead it to the destination. This is achieved by using the differential driving technique [2]. RoboSantral achieves rotation depending on the speed difference on each side of the base. As soon as the angle $\boldsymbol{\alpha}$ is zero speed becomes equal on each side of the base and RoboSantral moves straight towards the direction.

Kinematic and dynamic models of a robot which has such movement system are obtained following the methods given in [2] and [3].

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ -\frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} k_\rho \rho \\ k_\alpha \alpha + k_\beta \beta \end{bmatrix}$$

Where k_ρ, k_α and k_β are the control parameters. k_ρ, k_α are implemented to drive the robot towards the target linearly. Whereas k_β is employed to drive

$$\beta \to 0$$

IV. SYSTEM DESIGN AND OPERATION

The block diagram that represents the system design is shown

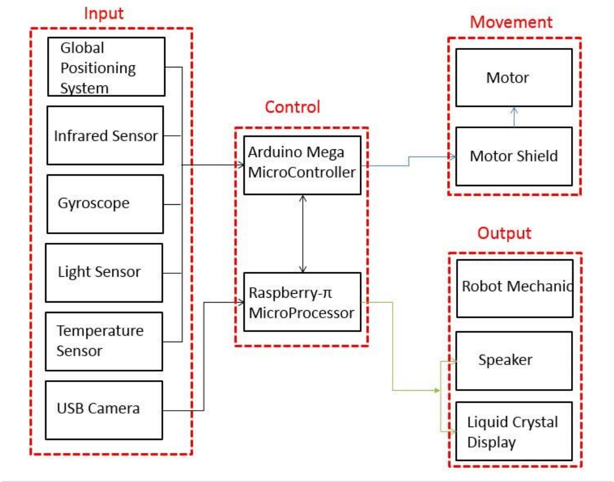

Fig.5: Block diagram of the system

in Fig.5. Input block that collects the environmental data lies on the left column. This block makes RoboSantral able to understand different types of environmental data using various sensors inside the robot. Moreover RoboSantral often encounters new environments and images. Image processing software process them. For this purpose software called "OpenCV" is used and various add-ons are added using C++ language.

Four infra-red sensors on the input block are placed on both sides, on the front and on the back. When these sensors met an obstacle within 50 cm the robot stops. If the obstacle leaves within 10 seconds, robot continues its former direction. Otherwise a new direction is set to avoid the obstacle.

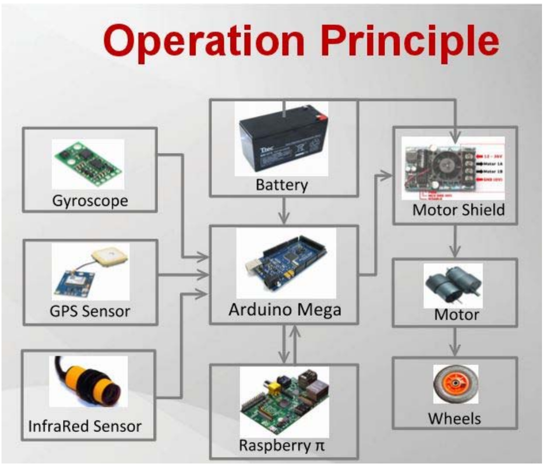

Fig.6: System diagram of Arduino

In addition to that a gyroscope and GPS sensors take place on the input block. Robot receives sufficient information from the gyroscope and the GPS sensor and is able to change the direction accordingly. Furthermore the data comes from GPS sensor is transferred to Raspberry Pi from Arduino so that the audio and video presentations related to the location can be started. The communication between Arduino and Raspberry Pi is achieved through the serial monitor.

Fig.7: System diagram of Raspberry Pi

Arduino and Raspberry Pi that are placed on the control block of the system diagram make the robot work and move. Motor driving is achieved through the motor shield that is on the movement block of the system diagram. Motor driving manages the currents going to the wheels therefore makes turns and a balanced straight movement possible. On the robot six 12 V/200-rpm motors are driven by the motor shield.

OpenCV, Raspberry Pi and a USB camera are responsible for image processing. OpenCV is widely used software that makes real time image processing, movement tracking and face recognition using its free library. C, C++ and Python are available languages for OpenCV and C++ is used in this project. The code consists of several predefined signs. When these signs are observed by the USB camera, they are processed by Raspberry Pi and Arduino and robot moves forward, backward, left or right. At the same time a video that presents the location is started on the LCD screen.

The code for image processing consists of the following 3 modules;

1. Definition of the libraries and source files.
2. Comparison of the camera input and the source files
3. According to the comparison results, audio and video play and communicating with Arduino.

RoboSantral recognizes certain buildings in a university campus using the predefined location tags. These tags give information about the location and also assign the following maneuver. [Fig.8]

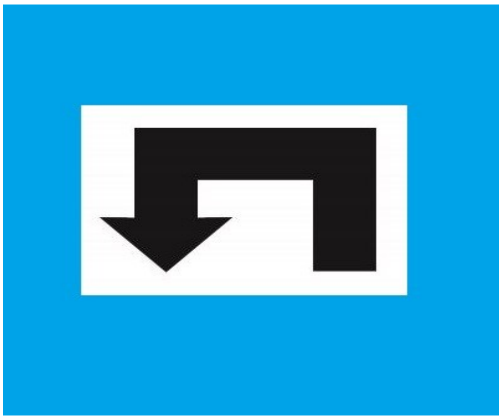

Fig.8: An example of the re-defined location tags on buildings

In Fig.9 there is the screenshot of the animation of RoboSantral, reading the sign and giving the presentation before the next maneuver.

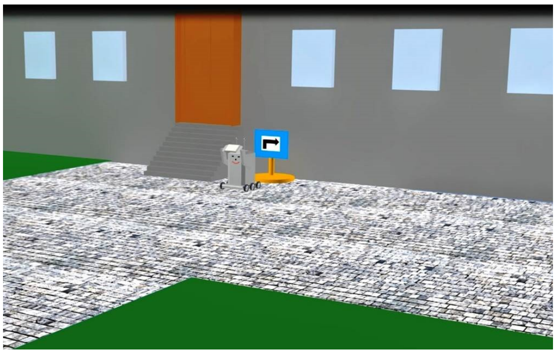

Fig.9: Screenshot of the animation that shows RoboSantral giving a presentation

In the image processing phase techniques such as Canny Edge Detector and Gaussian Blur Filtering are used. The Canny Edge Detector that was found by John F. Carry in 1986 was employed for eliminating the details of an image, and keep the main lines that will help with the recognition so that having faster processing.

Gaussian Blur Filtering function also provides a simpler look to the captured images so that image processing function has less to process. After Gaussian Blur Filtering, Canny Edge Detector is applied to the image. Image that was processed by both functions has a simple and smooth map.

When both functions are applied to a round object only a plain circle (Fig.11) is observed as a result.

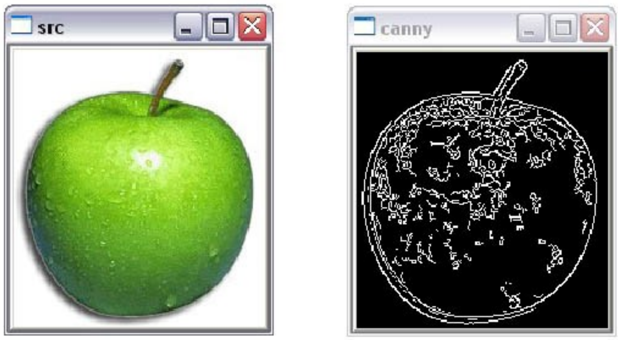

Fig.10: An apple and its Canny Edge Detector result

In order to fix the perspective of the image, a software called

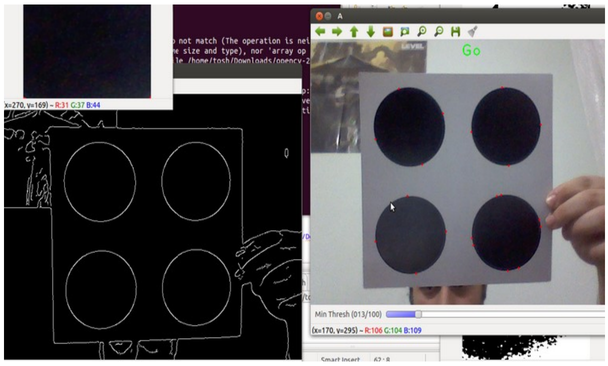

Fig.11:The results of the The Canny Edge Detector function applied on a round object.

“warpPerspective” is developed which has a module that is able to read the images warped until 30 degree. Therefore, RoboSantral can read and process the images even if it’s not right in front.

In Fig.12 there is the observation and processing of a location tag

Conclusions

RoboSantral was tested on Istanbul Bilgi University Santral Campus. Two images that were captured during the test are presented on Fig.13 and Fig.14.

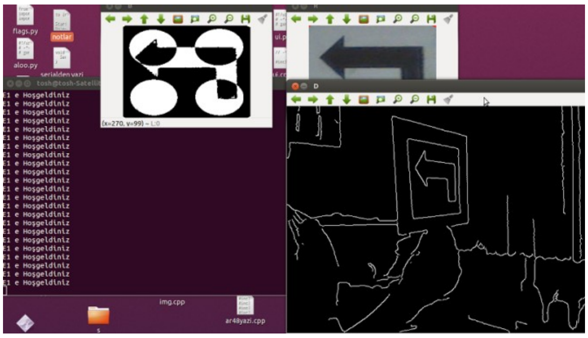

Fig.12: Results of the image processing of a building tag

Fig.13: RoboSantral sets off.

Fig.14: RoboSantral on its route

Even in the tests made during the rush hours of the campus, RoboSantral did a satisfactory job avoiding people and other obstacles.

In the light of these results, it is clear that autonomous mobile robots can achieve guidance tasks in campuses, museums and such.

REFERENCES

[1] B. Sciliano ve O. Khatib, *Springer Handbook of Robotics*, Springer-Verlag, 2008.

[2] R. Siegwart, IR Nourbakhsh ve D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, MIT Press, 2011.

[3] P.Corke,*Robotics, Vision and Control,*Springer tracts in advanced robotics, 2011

[4] J. Canny, " A computational effect to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679-698, Nov. 1986