Mobile Robot Navigation Using Omnidirectional Vision

**László Mornailla, Tamás Gábor Pekár,
Csaba Gergő Solymosi, Zoltán Vámossy**

John von Neumann Faculty of Informatics, Budapest Tech
Bécsi út 96/B, H-1034 Budapest, Hungary
E-mail: palcommander@googlegroups.com
Web: http://palcom.bmfnik.hu

Abstract: *The topic of this paper is the design and the implementation of a robot, which is capable of autonomous navigation within a homogeneous, weakly-textured environment. The architectural components of the robot are an RC model car, a remote controller extended with some custom electronics, a wireless camera with panoramic annular lens, and a standard PC for controlling the robot.*

Keywords: *PAL (Panoramic Annular Lens), omnidirectional vision, mobile robot, robot sensing systems, obstacle detection, obstacle avoidance*

I INTRODUCTION

The goal of this project is to create a wheeled robot equipped with a PAL-optics, which is capable of autonomous navigation: collision-avoided line-, and object following within a weakly textured environment. The long-term goal of this project is the ability of autonomous mapping of the environment; finding, and navigating through user-specified path; and searching for a predefined object within an unknown environment. In the following, the article will summarize the components of the system.

II THE CONSTRUCTION

A remote controller Model RC is used as the base of the robot. There is a PAL-optics equipped wireless camera fixed upon it. The result of the camera is relayed using a TV tuner to the main program, which controls the robot automatically using the images gained from the camera as its only input.

III HARDWARE

The Model RC is capable of precision controlling: both the direction and the speed can be set. The range of the remote control is about 30-40 meters; maximal speed is about 20 km/hours.
The PC is connected to the remote controller with an extra electronics, and then controls both controller transistors with three impulses. The impulses are generated by an 18F1320 PIC micro controller. An interrupt is generated in the PIC program with a timer every 15ms. On every interrupt, the PC is sending three signals: the first signal sets the beginning of the periodical time. During this, the PIC program starts a second timer, with a

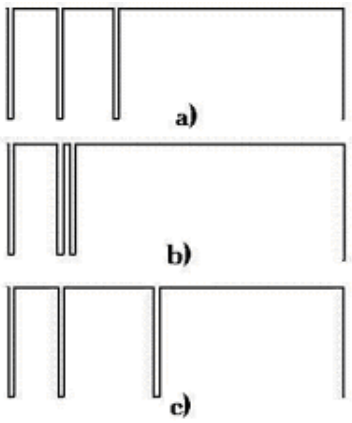

Figure 1
The controlling impulses: a, center direction; b, zero speed; c, maximal speed

timeout set by the computer. After this timeout, a second impulse is send, which controls the direction of the car; then PIC program restarts the counter with the third control-data, which sets the speed of the car.

IV PANORAMIC VIEW

PAL optic [4] is a device that transforms the surrounding environment to an annulus. Huge advantage is that from the image provided by this optic the direction and angle of the object can be directly computed. The PAL-optic with its reflecting and refracting surfaces consists of one single glass block. An auxiliary optic, however, is needed to project the virtual panoramic annular image that is formed inside the optical block to the detector surface, preferably, a CCD chip. Using this lens as sensor may reduce the necessary number of sensors and thus energy use as well.

Other advantage of using PAL optics, instead of a perspective one, is that there is no need of moving components; and there is no need to focus to gather information from differing distances. Therefore, one optical sensor gathers sufficient amount of information for the navigation.

The PAL-optic projects the 360 degrees of the environment to 2 dimensions, thus creating a circular image. The cylindrical projection of the image can be described using a polar coordinate-system. Due to the structure of the PAL-optic, there is a blind spot in the middle of the image, which must be taken in account while processing it. The main properties of the optics are: center of the PAL, inner radius, outer radius, and the angle of the field of view. The PAL-center is usually shifted from the middle of the image; therefore, software correction is needed.

The camera is fixed on the top of the mobile car, looking at the floor; therefore, it is capable of observing the immediate environment only.

Figure 2
The mobile robot

V PRE-PROCESSING

The image flow arrives from the Input module, which is responsible for either capturing images from a camera, or play back a test video file. It forwards the images to both the decision maker, and the mapper module. The decision maker analyses the image, and sends a direction/speed signal to the navigation module, which, in turn, forwards it to the controller PIC.

In order to make a valid control decision, the image is preprocessed. Using a HSL [5] filter, the program segments the image to Hue, Saturation, and Luminance components. The Hue component is between 0, and 360, the Saturation, and the Luminance will fall between 0, and 100.

The HSL filter is used in two cases: in line following mode, when the predefined track is homogeneous, or – in object-following mode – when the object to be followed is significantly differing in color from the rest of the environment.

The RGB filter is almost as efficient as the HSL, but the algorithm is significantly faster. Using this filter, the three color channels is analyzed, using a minimum, and a maximum values; if the color of the pixel is within these values, then it remains intact on the resulting image; otherwise the filter will make it black.

Using the threshold filter, image binarization can achieved; the result image will contain only the pixels needed for navigation.

VI NAVIGATION

One complex algorithm is used for all three of the navigation types (line following, object following, and free fall navigation).

First, the properties of the PAL-optics needs to determined: image center point, inner blind spot radius (*r1*), and effective radius (*r2*).

Then, some user-defined values are gathered:

- *Distance* (1): determines the maximal processing distance, in percent, relative to r2;
- *Scanning degree* (2): the maximal degree of scanning;
- *Turning degree* (3): the maximal turning degree of the robot;
- *Threshold minimum*: the minimal acceptable sum of pixel intensities.

For line following, the algorithm determines a direction line, which converges to the center of the image, and begins *r1* distance away from the center point. The optimal length of the line of sight can be set using the distance value.

The algorithm determines the pixel intensity under the line of sight starting from 90 degrees (forward), scanning at each iteration first left, then right, until it reaches 90± scanning degree value. For every line of sight, it determines the sum of the pixel intensities, whether it exceeds the value set by threshold minimum; the maximum of these values will be used as a navigation direction.

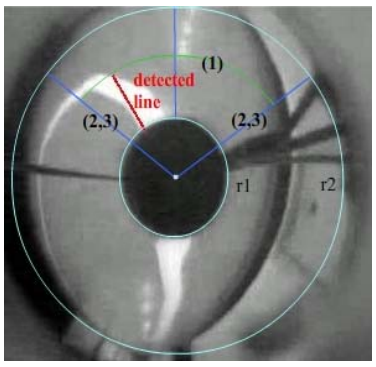

Figure 3
The PAL image, and the line of sight

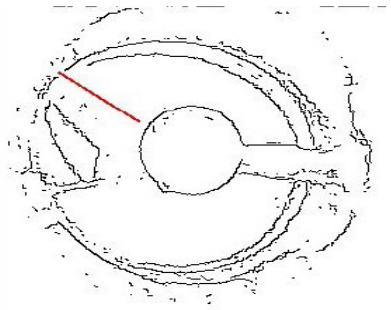

Figure 4
Track navigation with Canny edge-detection

This value is later verified, whether the robot fits on the given path, or not.

If there is no appropriate path on the top part of the image, the bottom part will be analyzed by the same method, and the robot will either reverse, or stop.

For free fall navigation, the Canny edge-detection [6] algorithm is used: the contour on the image will be interpreted as obstacle to avoid. The algorithm will potentially avoid these obstacles by selecting the longest clear direction.

For object following, the algorithm uses the RGB, and HSL filter described above, to distinct the object from the rest of the environment. After applying binarization, the controlling line will head towards the object.

Figure 5
Object following

VII MAPPING

Although the mapper module is being constructed at the time of writing this article, some early results are already available.

The mapper module will be used to implement a user-defined navigation: after the robot builds up the map of the environment, the user sets some checkpoints, and the robot tries to find the shortest path, and navigate through them, while avoiding any obstacles. The specific purposes of this module are localization of the robot (relative to the starting point), and storing, and maintaining a virtual top-view image of the environment. The hardware circumstances are somewhat similar to Caboto [2], though a different approach is used.

To achieve this goal, first the image received from the PAL-optics is mapped to a virtual top-view. To apply this transformation the algorithm assumes that the PAL-image is a regular annulus. Thus, transforming the distance of the pixels from the PAL-center will result in a top-view, map-like image. To determine the curve of this transformation, the relation is measured between the real distance, and the PAL-distance on several points from the center of the image; and a cubic spline interpolation is applied on the measured data. To increase the performance of the algorithm a transformation matrix is generated, which determines the source for every pixel on the resulting image. Once this matrix is generated, it can be used for every image, with real-time speed.

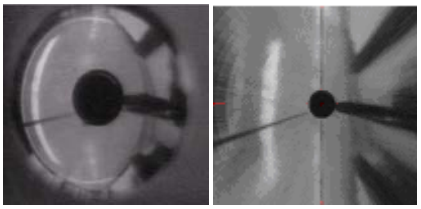

Figure 6
Original PAL-image and the corresponding virtual top-view

After the top-view transformation, the module will use a static mask to cut out segments from the image, which has no information-content [for example, the central blind-spot]. Then it will search, and track characteristic features on the resulting image. This feature points is used to determine the location, and angle of the robot, utilizing a Kaman-filter [3].
After localizing the robot, the module will rotate the image to reflect the initial direction of the robot; the resulting image will be melted into the global map.
In order to dynamically extend the map as the robot gathers more, and more information, a static bitmap would not be sufficient; instead, the map is divided it into several, small images, and the module stores the 2 dimensional ordering between these segments.
The "melting" of one image into the global map will use a statistical approach: the value of each pixel will be averaged by the value of every corresponding pixel of the separate images.

VIII RESULTS

The robot was tested firstly in an indoor environment using a pre-build test field. Acceptable results was achieved in following a line, track-navigation, and object following.

There were three disadvantages of the indoor environment: the narrow corridor, the reflection of the neon lamp, and sunlight on the floor (see Figure 3). In order to eliminate these problems, the equipment was tested at an outdoor environment, on concrete floor; this resulted in a more stable navigation.
The duration, and processor usage of the used algorithms was measured; the results are shown in Table 1.

	Duration	***Processor usage***
Avg. time with a test video	< 1 ms	3%
Line following	< 1ms	5%
Canny	6 ms	35%
HSL	43 ms	70%
RGB filter	2 ms	10%
Threshold	5 ms	24%

Table 1
Measuring results of the filters, and algorithms used

IX IMPROVEMENTS

The current system has several drawbacks, which might be eliminated in the future.
On the hardware part, the range of vision and the quality of the signal has to be increased, by using a more advanced camera. The number of peripherals used in the robot also has to be decreased.
To fulfill these goals, a new robot is under design. This robot communicates with the processing unit using a WiFi connection, thus increasing the manipulation range of

the robot. It would be equipped with a laptop computer, which is connected physically with the camera, and the motor-controlling PIC. The laptop would stream the video signal to the controlling unit, and receive the controlling values, both via a wireless local area network.

Conclusion

The developed system, a low-budget mobile robot, is easy to reproduce. The system has met its current specified criteria: it is capable of collision-avoided line-, and object following, within a weakly textured environment.

Acknowledgement

The research has been supported by the OM TDK grant under the terms of grant OM FPO 245540/2005.

References

[1] The Page of Omnidirectional Vision, http://www.cis.upenn.edu/ ~kostas/ omni.html

[2] The Caboto Project, www.dei.unipd.it/~emg/papers/ caboto.pdf

[3] An Introduction to the Kalman Filter, http://www.cs.unc.edu/~welch/ media/pdf/kalman_intro.pdf

[4] Veress, M., Greguss, P.: Centric Minded Imaging in Space Research, In: Proc. of 7th International Workshop on RAAD'98, (K. Dobrovodsky (Ed.)), 1998, pp. 121-126, Bratislava, Slovakia

[5] Adrian Ford, Alan Roberts: Colour Space Conversions, http://www.poynton.com/PDFs/coloureq.pdf

[6] Canny, J: A Computational Approach for Edge Detection, IEEE Trans. Pattern Analysis Machine Intelligence, Vol. 8, No. 6, 1986, pp. 679-698