An Algorithm for Vision-Based Navigation of an Indoor Mobile Robot

Eng. Doaa Abdelfatah, Dr. Ashraf M. Hafez, Ass. Prof. M. Lotfy Rabeh
Shoubra Faculty of Engineering, Cairo, Department of Electrical Engineering

Abstract

In this paper an efficient, robust, mapless navigation algorithm is developed. This navigation algorithm enables the robot to avoid collision with obstacles existing in an indoor environment (e.g. chairs, desks, file cabinets, persons, etc.) and determining which direction (Right, Left, Forward, or Stop) the robot should take. The algorithm has been developed and verified using MATLAB image processing toolbox.

1. Introduction

Research on Mobile Robots is one of the most popular topics in the field of artificial intelligence[1],[2],[3],[4]. Computer vision is the most sophisticated way of perception of the environment for robotics applications. Robots with computer vision systems obtain images of an unknown 3D environment by means of one or several optical cameras, interpret these raw images by image processing and analysis techniques, and then make decisions and plan their actions accordingly. The main aim of this paper is to develop a simple and robust navigation algorithm based on computer-vision principles. In general, five approaches are used for vision-based indoor navigation[5], these are:

i. **Stereo Vision**: These are systems that use two or more cameras to determine the depth of every point in the environment[6],[7]

ii. **Map-Based Navigation**: These are systems that depend on user-created geometric models or topological maps of the environment[8],[9]

iii. **Map-Building-Based Navigation**: These are systems that use sensors to construct their own geometric or topological models of the environment and then use these models for navigation[10]

iv. **Mapless Navigation**: These are systems that use no explicit representation at all about the space in which navigation is to take place, but rather resort to recognizing objects found in the environment or to tracking those objects by generating motions based on visual observations[11],12]. This approach is simple and computationally not expensive.

v. **Range Detection**: These are systems that use a laser coupled with a camera to determine the distance of every point in its field of view[13].

2. Development of the Navigation Algorithm

Some assumptions were made:

1-The camera is black and white, mounted above the robot at a fixed height, and angled downward as shown in figure 1

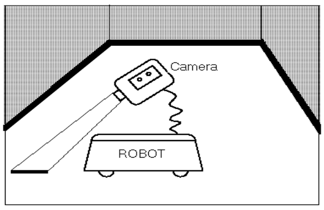

Figure 1
The system configuration

2-The floor is flat and bright .

3-The lower portion of the wall is dark (so the robot can discriminate it from the floor)

4-The image is captured with a size of 340 columns and 180 rows.

5-the algorithm should be robust and simple.

Now we will explain how the navigation algorithm has been developed. The flowchart of the Navigation algorithm is shown in figure 2. First, the image is captured with a size of 340 pixels wide and 180 pixels tall (340 columns x 180 rows) as shown in figure 3. The color depth is specified to be 256 grayscale. The image is then smoothed with a 3x3 median filter to remove noise. A median filter is used because it achieves less blurring than mean filters [14]. The image is then sliced into a binary image using thresholding. We have tried the adaptive threshloding technique and the iterative thresholding technique. The iterative thresholding technique is chosen as the final thresholding algorithm because it successes in the segmentation process. It is implemented as follows[14]:

1-Select an initial estimate of the threshold (T). A good initial value of (T)is the average intensity of the image[15].

2-Segment the image using the threshold (T). This will produce two groups of pixels:

G1: consisting of all pixels with gray level values <(T)

G2: consisting of all pixels with gray level values)>=T)

3-Compute the average gray level values $\mu 1$ and $\mu 2$ for the pixels in regions G1 and G2

4-Compute a new threshold value, $T=0.5(\mu 1+ \mu 2)$

5-Repeat steps 2 through 4 until the difference in T in successive iterations is zero.

0-7803-9521-2/06/$20.00 ©2006 IEEE.

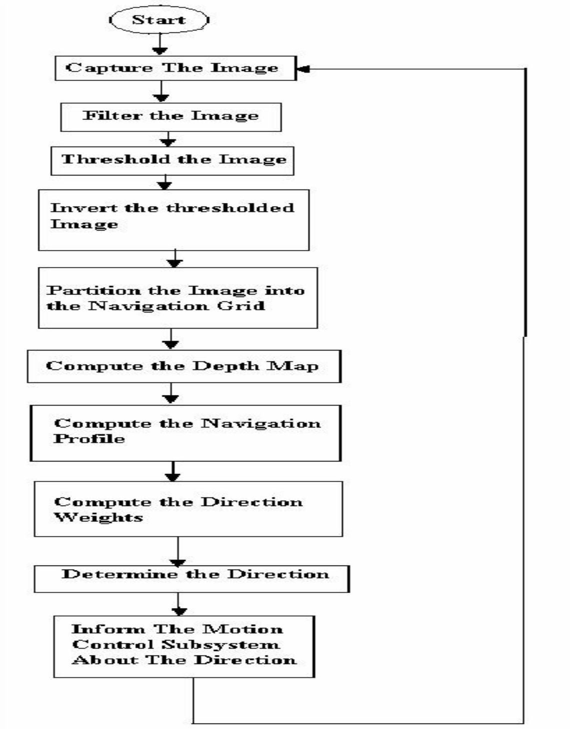

Figure 2
Flowchart of the Navigation Algorithm

Figure 3
Typical Office Image

Figure 4
Inverted thresholded image

To reduce the amount of data to be processed , the inverted thresholded image is divided into a grid of 9 rows by 17 columns as shown in figure 5

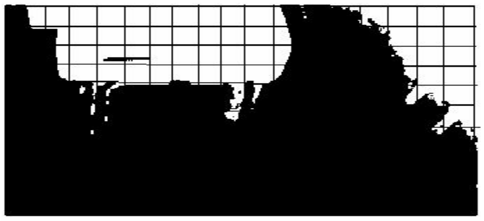

Figure 5
Inverted thresholded image with the 9x17 grid shown

The elements of the grid are called the navigation elements (navels) .We will have 9x17=153 navels . Based on the 180x340 measurement of the image , each navel is a 20 by 20 pixels square. For each navel , the total number of white pixels is determined. If the total number of white pixels for a given navel exceeds 25% [16] of the total number of pixels in this navel (i.e. 100 out of 400 pixels) , this navel is marked as an obstacle element (obsel) and given the binary value 1 or white otherwise it is marked as an ordinary navel and given the binary value of 0 or black as shown in figure 6.

Figure 6
The Navigation Grid

Notice that the algorithm accurately ignored the navels between the floor tiles by not marking them as obstacle elements (1's). Notice also , along the boundary between the floor and the chair , there are several navels that correctly labeled as obsels with the 25% threshold selection used .

After computing the navigation grid , the Depth Map (DM) that describes the amount of free space has to be computed . This is done by starting at the bottom of the navigation grid and tracing up each column until an obsel (1or white) is found then start filling this column from this obsel and up to the first row with 1's as shown in figure 7.

Every obstacle rests on the ground plane , so we can use the DM columns heights to estimate for depth . This results in calculating the navigation profile (NP).This is done by starting at the bottom of the depth map and counting the number of black elements (0's) in each of the 17 columns then record this number as shown in figure 7 .

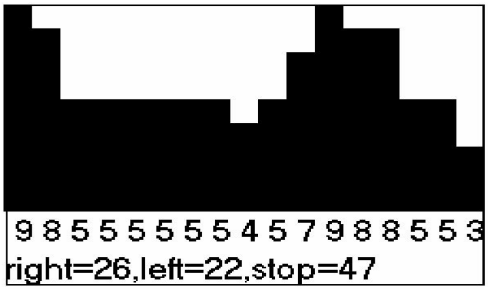

Figure 7
The Depth Map (DM) ,The Navigation Profile (NP), and The Direction Weights

Each digit of the navigation profile (NP) corresponds to the nearest obsel in each column .We have now reduced an image containing 180x340=61200 pixels to a single 17-digit number . All the data we need to produce accurate navigation is presented in this NP. It is clear from the NP shown in figure 7 that the robot should stop to avoid collision with the chair. The final task will be to convert this navigation profile into basic directional message like "Turn Right", "Turn Left", "Stop", "Move Forward". To do this three direction weights (right , left , and stop) are calculated as follows[16]:

i-A perimeter profile PP of 1 1 2 3 4 5 6 7 7 7 6 5 4 3 2 1 1 which emphasizes the criticality of obsels in the center of the image (i.e. directly in front of robot) is defined .

ii-Also 3 sets R , L , S are defined:

R={10,10,9,8,7,6,5,4,3,2,1,0,0,0,0,0,0}

L={0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10,10}

S={0,0,1,2,3,4,5,6,6,6,5,4,3,2,1,0,0}

iii-Subtract the digits of the NP from the digits of the PP , replacing negative results with zero (This calculation gives the amount each obsel intrudes into the zone of interest , that is the area directly in front of the robot , or the center of the image)

iv-Multiply the difference by the corresponding element from each of the sets R , L , and S and sum the products . This gives the 3directon weights:

Right=26, Left=22, Stop=47

To determine which direction , if any , the robot should take , the following tests are performed :

-If stop>=40 , Move forward

-If Stop<40 , Left < Right , and Left<80 ,Turn Left

-If Stop<40 , Right <Left , and Right<80 , Turn Right

-If Stop<40 , Right =Left , and Right<80 , Turn Right or Left

-If Stop<40 , Left < Right , and Left>=80 ,Stop

-If Stop<40 , Right <Left , and Right>=80 , Stop

-If Stop<40 , Right =Left , and Right>=80 , Stop

So the robot should stop

The R , L , and S sets , the PP , and the minimum weight values of 40 , 80 are determined experimentally by examining a number of images . If the camera orientation or the building is changed , theses parameters have to be readjusted .

3. Experimental Work

Extensive number of experiments is carried to test the validity of the developed algorithm. A typical "Turn Right" , "Turn Left" , "Stop" , and "Move Forward" images are shown in figures 8, 9, 10, 11 respectively. The depth maps, navigation profiles, and direction weights of these images are shown in figures 12, 13, 14, 15 respectively. The images have been taken by a digital camera in a real office building and processed by the suggested navigation algorithm.


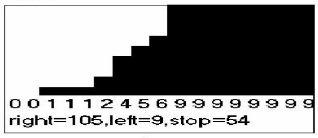

Figure 8 Figure 12


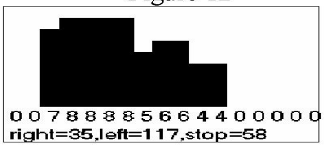

Figure 9 Figure 13

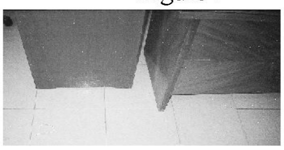
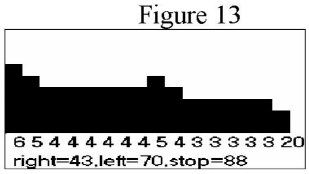

Figure 10 Figure 14


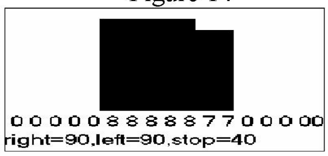

Figure 11 Figure 15

The results have shown that the algorithm is efficient in determining the direction the robot should take.

4. Conclusion

The problem of mobile robot navigation based on vision sensing has been manipulated in this work. A simple, robust, and efficient navigation algorithm that enabled the mobile robot to navigate without colliding with obstacles has been developed. The algorithm partitions the images into 20 by 20 pixel navigation elements, or navels. The number of white pixels in each navel is determined . Navels whose total number of white pixels exceeds a given threshold are labeled as obstacle elements, or obsels. The position of the nearest obsel in each column of the navel matrix is noted. These obsel positions form the navigation profile. This navigation profile is then used to determine what course the robot should take. The experimental work has verified the validity of the developed algorithm.

An extension to this work would be to implement the algorithm on a real mobile robot using a low cost microcontroller. Improving the direction resolution is another issue. The use of advanced soft computing techniques such as fuzzy logic, neural networks, and genetic algorithm should also be considered.

5. References

[1] Kendi C., Erden A., "Development of a Prototype Mobile Robots As Technology Demonstrators", 6th Int. Machine Design and Production Conference, pp. 81-92, 1994.

[2] Nielsen L., "Automated Guidance of Vehicles using Vision and Projective Invariant Marking", Automatica 24:2, pp. 135-148, 1988.

[3] Schraft R.D., "Mechatronics and Robotics for Service Applications", IEEE Robotics & Automation Magazine, pp. 31-37, December 1994.

[4] Harris G., "Robotics Decontamination Keeps Operators Clear of Danger", Industrial Robot, 20:3, pp. 30-33, 1993.

[5] Guilherme N. DeSouza and Avinash C. Kak ,"Vision for Mobile Robot Navigation: A Survey", IEEE transactions on pattern analysis and machine intelligence, Vol. 24, No. 2, February 2002

[6] W Eric and L. Grimson , "Computational Experiments with a Feature Based Stereo Algorithm," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-7, no. 1, pp. 17-34, January 1985.

[7] Cho Y. C., Cho Hyung S., "A stereo vision based obstacle detecting method for mobile robot navigation ", Robotica, 12, pp. 203-216, 1994.

[8] T. Tsubouchi and S. Yuta, "Map Assisted Vision System of Mobile Robots for Reckoning in a Building Environment," Proc. 1987 IEEE Intl. Conf. on Robotics and Automation, pp. 1978-1984, 1987.

[9] K. Sugihara, "Some Location Problems for Robot Navigation Using a Single Camera," Compt. Vision Graphics Image Process., 42, pp. 112-129, 1988.

[10] Z. Zhang and O. Faugeras," A 3D World Model Builder with a Mobile Robot", Int'l J. Robotics Research, vol. 11, no. 4, pp. 269-285, 1992.

[11] Bauzil, G., Briot, M. and Ribes, P., "A Navigation Sub-System Using Ultrasonic Sensors for the Mobile Robot HILARE." 1st In.. Conf. on Robot Vision and Sensory Controls, Stratford-upon-Avon, UK., pp. 47-58 and pp. 681-698,1981.

[12] Giralt, G., "Mobile Robots." NATO ASI Series, Vol. F11, Robotics and Artificial Intelligence, Springer-Verlag, pp. 365-393, 1984.

[13]Giralt, G., R. and Vaisset, M.," An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots", in M. Brady and R. Paul (eds), Robotics research- The First international Symposium, The MIT Press, New Hampshire, pp. 191-214,1984.

[14]Ramesh jain, Rangachar Kasturi, Brian G. Schunck, "Machine Vision," McGRAW-HILL Int. editions, Computer science series,1995

[15]Rafael C. Gonzalez, Richard E. Woods," Digital Image Processing," second edition, Prentice Hall, Upper Saddle River, New Jersy, 2001

[16]Joshua J. Potts, "Real-Time Vision-Based Autonomous Robot Navigation", master of science, University of Illinois at Urbana-Champaign, 1999