JOURNAL OF INTELLIGENT & ROBOTIC SYSTEMS

Theory & Applications

CODEN JIRSES
ISSN 0921 – 0296

Vol. 45, No. 1 January 2006

Journal of Intelligent and Robotic Systems (2006) 45: 15–30
DOI: 10.1007/s10846-005-9014-4

RISCBOT: A WWW-Enabled Mobile Surveillance and Identification Robot

SAROSH PATEL, RAJEEV SANYAL and TAREK SOBH
RISC[*] lab, University of Bridgeport, Bridgeport, CT 06601, USA

(Received: 8 October 2004; in final form: 22 July 2005)

Abstract. This article describes RISCBOT (RISCBOT name has been derived from RISC lab and 'bot' of robot), a modular 802.11 b-enabled mobile autonomous robot built at the RISC lab of the University of Bridgeport. RISCBOT localizes itself and successfully fulfills www – enabled online user requests and navigates to various rooms, employing a visual recognition algorithm. This article describes the mechanical design, hardware and software algorithms of the robot, and the web-based interface for communicating with the robot.

Key words: autonomous robot, character recognition, image processing, telerobotics, wireless.

1. Introduction

Telerobotics (controlling robotic devices from a distance) has enjoyed a rich history. It has led to many practical applications and to a broad vision of interacting with environments far removed from the user. With the advent of the Internet, telerobotics has received a major boost.

A number of mobile robots exist in the world today catering to online requests to move to a desired location. Xavier [3] can accept commands to travel to different offices within a CMU building, broadcasting camera images as it travels. Minerva [1] is an interactive autonomous robot that moves daily through crowds at the Smithsonian's National Museum of American History. Rhino [2] has been deployed as a tour guide robot at Deutsches Museum in Bonn, Germany.

The online robot we built, RISCBOT, utilizes room visual identification for localization. RISCBOT is equipped with an onboard PC (personal computer), WLAN (Wireless Local Area Network) card, NM6403-based DSP (Digital Signal Processing) board, batteries, cameras and ultrasonic sensors. The robot has been designed modularly in order for researchers in the future to be able to add new features to the present system. Online users receive real time video feedback from the robot and can also view the robot position. Navigation is

[*] RISC lab: Interdisciplinary Robotics, Intelligent Sensing and Control laboratory at the University of Bridgeport.

performed with the help of the cameras and ultrasonic sensors. The robot processes images from the camera to differentiate between doors, walls and obstacles. The robot can navigate the University of Bridgeport (UB) Engineering Technology building and successfully fulfills online users' requests from the Internet. It can run uninterrupted for about an hour, but has to be recharged hourly.

RISCBOT navigates the second floor of the UB TECH building and can identify 11 rooms. A network camera (AXIS) is installed at the ceiling of the second floor of the TECH building, giving the user a view of the robot and the different rooms. The installed wireless access point has a range of 50 ft. Figure 1 depicts a screen shot of the robot web server.

2. Mechanical Construction

The initial design of the robot was done using Pro Engineer (ProE). The robot frame is built with T-slotted aluminum extrusion rods to allow for a modular structure. A differential drive mechanism has been implemented with two 4-in. wheels and a caster wheel for support. Two 12-V dc servo-motors (Pittman) drive the wheels. An ATM 103 MCU, inverter (purchased off the shelf) and a 12V Panasonic SLA battery are mounted on a 1/4-in. acrylic sheet. The PC cabinet housing the WLAN card and an NM6403 [5] DSP board is mounted on top of the base. Three ultrasonic sensors, two Logitech cameras and an NTSC camera are mounted on the PC cabinet. Figure 1a–c show different views of the mobile base.

The ATM103 MCU [6] controls the ultrasonic sensors and the two motors. The PC sends commands to the MCU through a serial port at 9,600 bps baud rate. A MATLAB program that checks for doors runs on the PC continuously. The NM6403 DSP board performs a visual recognition algorithm when signaled by the PC. Figure 2a,b show different views of the mobile robot platform. Figure 2c shows a sample task performed by RISCBOT.

3. Navigation Module

The navigation module serves as the central controlling unit commanding the motors though the MCU. The navigation module instructs the MCU when to move, execute a turn or stop. This module takes decisions based on the inputs from the server, the onboard ultrasonic sensors and the image-processing module. Figure 3 shows the control flow diagram of the navigation module.

The robot waits until it receives a command from the server to navigate to a particular room. Once it receives a command from the server it starts searching for the requested room. The robot navigates along the left side of the corridor. With the help of ultrasonic sensors, the robot maintains a distance of 45–50 cm from the left wall. If the robot gets closer, it turns right, if the robot gets farther

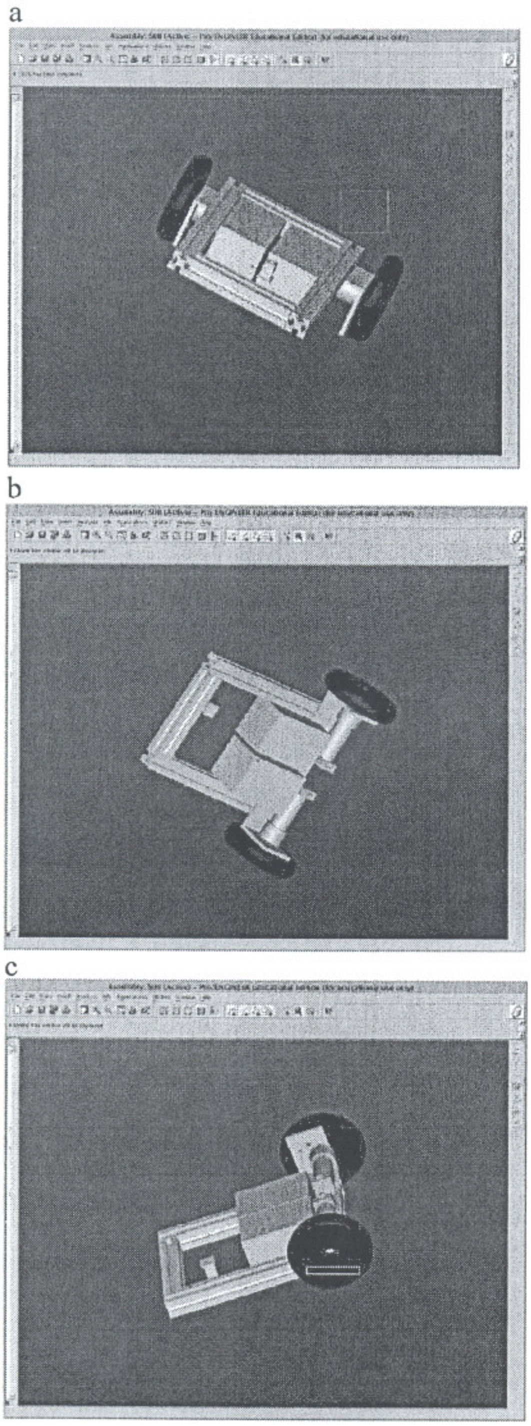


Figure 1. (a) *Top view* of RISCBOT. (b) *Bottom view* of RISCBOT. (c) *Side view* of RISCBOT.

a b c

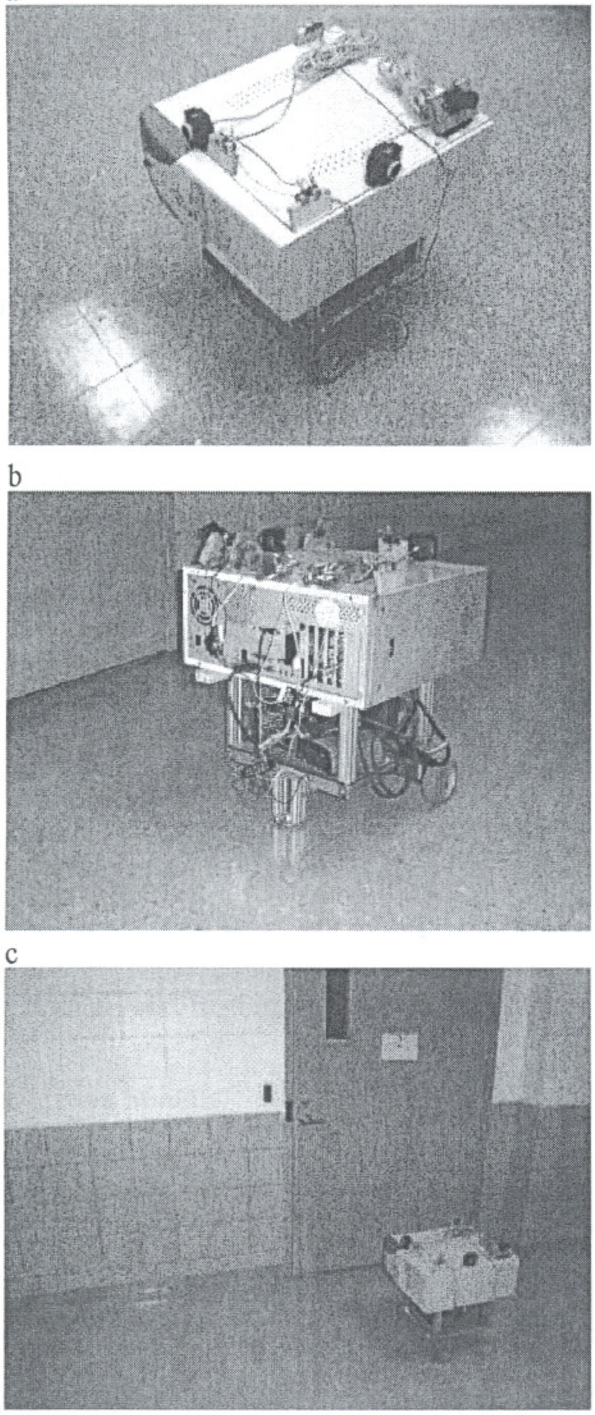


Figure 2. (a) RISCBOT top view (*front*). (b) RISCBOT top view (*back*). (c) RISCBOT accomplishes its task.

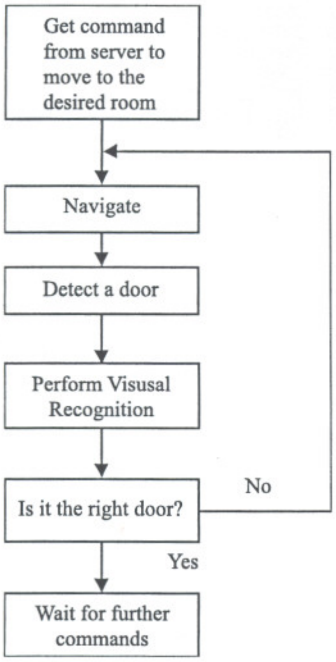


Figure 3. Navigation module.

away, it turns to its left and if its within the range of 45–50 cm, the robot moves straight. If there is a wall in front (i.e., near corners), the robot turns right. The turns are executed by a differential drive mechanism; the MCU controls the speed to the DC motors accordingly.

The image processing program checks for doors continuously. Once the program detects a door, it signals the MCU to stop the motors. The PC signals the NM6403 DSP board to check for the room ID. If the room ID matches the request, the robot stops. If not, the robot continues moving till it finds the desired room.

4. Image Processing

The door recognition algorithm is independent of varying light intensity and color. It is computationally fast, so that the doors can be recognized in real time and appropriate commands can be sent to the Navigation module to stop the robot in front of the desired door.

Our algorithm makes use of edge detection to differentiate between the wall and the door. As the walls are rougher, the edges can be easily detected by selecting an appropriate order for the filter.

We tried various filtering techniques for edge detection such as Sobel, Canny, Laplacian of Gaussian, Prewitt and Roberts. These filtering techniques where

Figures 4–8. Images captured by the Robot.

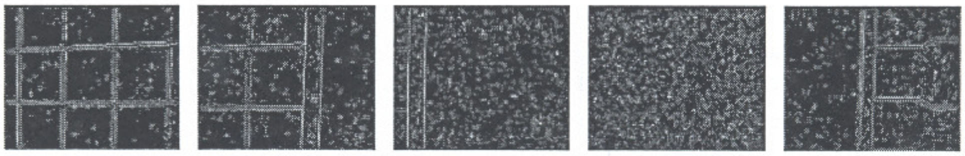
tried on a set of different images with varying filter orders and in both, the horizontal and vertical directions. The ideal filter for real-time door recognition is one which demonstrates good accuracy in detecting edges and can compute the threshold level dynamically depending on the intensity of light. Best results were obtained using a Laplacian of Gaussian filter [9] also commonly known as the LoG filter, with an order of 1.7. The order of the filter should be carefully selected, since with the increase in order of the filter, undesired and very minute edges show up.

This module is programmed in MATLAB. Images from the camera are captured on the run using the *vcapg2* utility [7], since MATLAB does not have built-in support for the USB port. Images are captured at a resolution of 352 × 288 pixels. The auto gain for the camera is turned off so that all the images are captured with a constant gain.

Figures 4–13 show a set of images captured by the camera and some results of the edge based door recognition algorithm. These images are converted to gray scale and then filtered to realize the edges in them.

As seen from Figures 4–13, the edges are represented by zeros (white). As the robot encounters a door, there is usually a sharp drop in the number of zero-valued pixels in the image, since the door has no fewer edges than the wall. This can be clearly seen in the graph in Figure 14. The graph depicted in Figure 14 was plotted as the robot moved across the corridor encountering doors. The graph shows the amount of white pixels (or edges in each frame). Sharp drops indicate that the robot has encountered a door. This technique is not independent of the varying light intensity. In regions where there is high light intensity, clearer edges can be seen when compared to low intensity regions.

A clear example of how the intensity of light varies along the corridor can be seen from the stem plot depicted in Figure 14. The smooth curves show the varying light intensity and the sharp changes are the doors (Figure 15).



Figures 9–13. Images after edge detection.

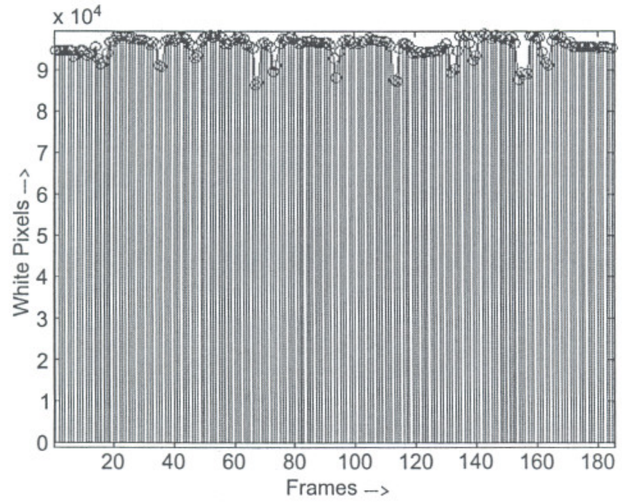


Figure 14. White pixels (*edges*) vs. frames.

A better strategy to recognize a door is to monitor the relative percentage change in the edges. When there is a drop in the relative percentage, below a particular threshold, the robot is assumed to have encountered a door. Figure 16 shows a plot for the relative percentage verses the frames. Sharp negative peaks below a threshold of -0.03 indicate doors. Figure 17 shows a plot of recognized doors.

There can be instances where only a partial image of the door is captured by the robot. In such a case there might not be a significant relative percentage drop as expected, even though a door has been encountered. The relative percentage threshold has to be set bearing in mind what percentage of the door if captured partially should be considered as a door.

The program maintains an internal count for the doors encountered, if the door recognition algorithm fails. In addition, adequate measures have been incorporated so that when more than one image of the same door is captured; the robot does not treat them as two different doors. Once a door is recognized, the control is passed to the Recognition Module in order to recognize the door.

5. Character Recognition

The door ID character recognition algorithm runs on the NM6403 DSP board.

5.1. IDENTIFYING THE DOOR PLATE

The room numbers have been printed on a plate, stuck on the doors. The image-processing algorithm identifies the plate on the door. We have implemented this

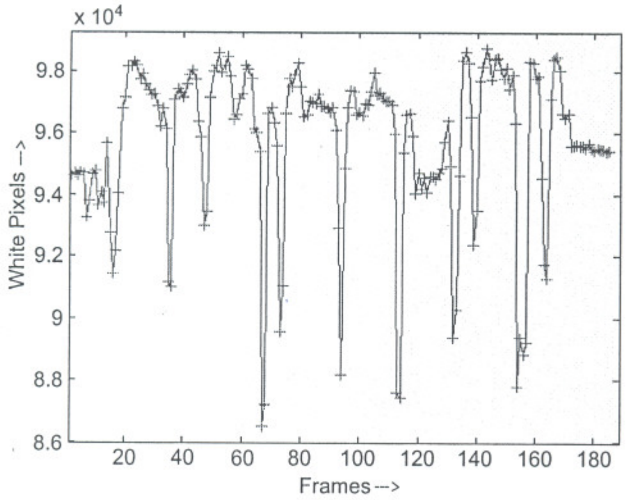


Figure 15. Varying light intensity.

task by using the Hough transform for detecting lines first, and then checked the relative dimensions of the lines for detecting the plate. We implemented the ideas described in [8].

Figure 18 shows the steps involved in detecting the plate [8]. Thresholding and edge detection is performed to reduce the number of processed points. The resulting image is then fed to the Hough transform, which then returns a list of straight lines. The algorithm for detecting a straight line is as follows:

1. Select start pixel S (*x*/*y*).
2. Select end pixel E (*x*/*y*).
3. Follow line from S to E pixel-by-pixel and count the number of pixels on that path that are set in the binary image.
4. If the counted number of pixel is greater than the threshold value, a line SE is present in the picture and hence is labelled.

Go back to step 1 and select two different pixels until the entire image is done. Finally, the list of straight lines is scanned to find pairs of straight lines that have the following attributes:

1. Both lines start and end at approximately the same position on the *x*-axis.
2. The relation between the length and distance of both lines should be equal to that of a door ID plate.

This algorithm is carried out on both horizontal and vertical lines. The resulting pairs from the list of horizontal and vertical lines are then compared and the end points not contained in both regions are discarded. The remaining regions

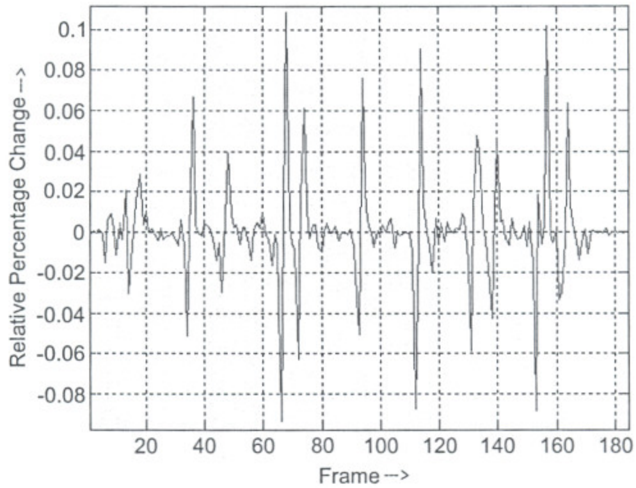


Figure 16. Relative percentage change.

are the final candidate regions, which contain the plate. A detailed explanation can be found in [8].

5.2. CHARACTER EXTRACTION

Although there are many methods for character recognition, the region-growing method has been adopted for the following reasons:

1. Fast Algorithm – Each pixel is examined no more than once for each neighbor.
2. Invariant to the distance between camera and door – The method extracts candidates with correct shape; it does not depend on size of regions.
3. Resistant to noise – The region is expanded to the largest possible rectangle based on maximum and minimum values.

Characters are extracted using a region-growing method [8]. The extracted space (from the top) is searched for a black pixel. When a black pixel is found it is assumed that it is part of a character. An iterative process is then carried out using the following equation described in [8]:

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3 \ldots$$

Where Xk represents the extracted component, A is the source image and B is a structuring element of size 3 × 3 indicating 8-connectivity neighboring. $X0$ is the first black pixel from the location where the iteration starts.

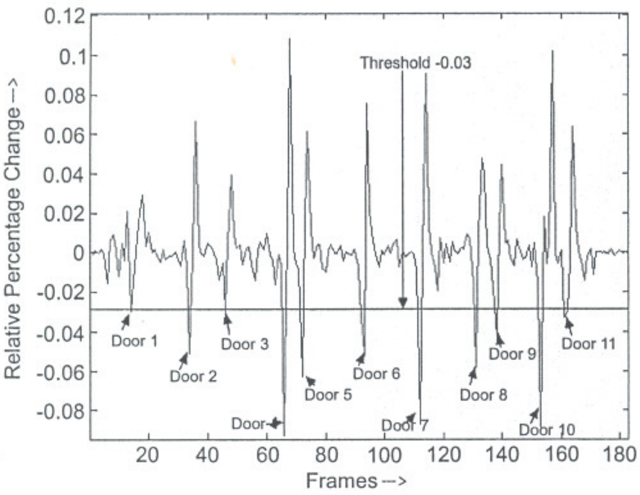


Figure 17. Plot showing the doors recognized.

This iterative algorithm creates a new image X_0, which only contains the first black pixel. X_0 is dilated (i.e., expanded) such that it contains all its neighboring pixels. Any of the neighboring pixels that are also black are considered to be a part of the component and are put into a new image X_1 along with the original black pixel to complete the first iteration of the algorithm. X_1 is dilated in the next iteration. This process continues until $X_k = X_{k-1}$.

When a connected component is found it is then tested to see if it meets the requirements of a character (e.g., size). This whole process is carried out iteratively until all the characters in the page have been extracted. Each character is labeled. Figure 19 illustrated the first two iterations in the extraction process [8].

5.3. SYMBOL RECOGNITION TECHNIQUE

We have implemented the symbol recognition technique described in [9]. In [4], Sridhar et al. describe a collection of topological features that can be used to classify numerals. Most of these features are properties of the outline or profile, of the numeral.

After the digit is isolated and thresholded, the number of background pixels between the left side of the character's bounding box and the first black pixel is counted and saved for each row in the bounding box. This provides a sampled version of the left profile (LP), which is then scaled to a standard size (52 pixels). A similar process produces the right profile (RP); the difference between the processes is that the last black pixel on each row is saved.

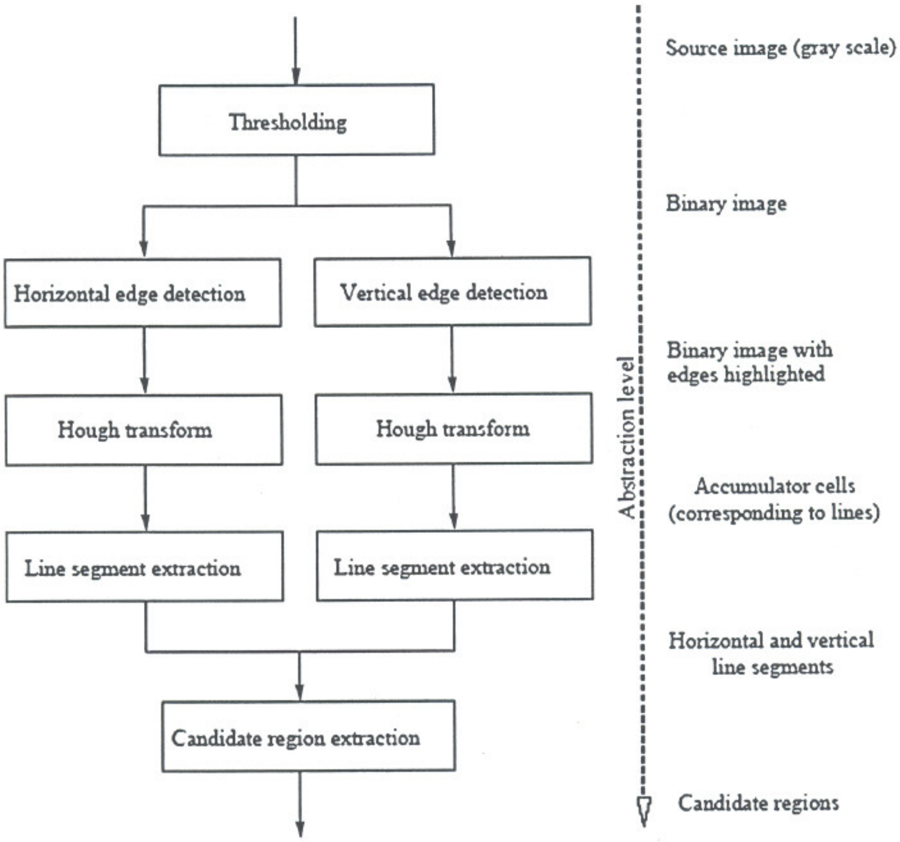


Figure 18. Steps for detecting the plate.

Having computed the profiles, the properties of the profiles are measured, as described in [9], as follows:

1. Location of extremas:
   a) Lmin: location of minimum value on the left profile.
   b) Lmax: location of maximum value of the left profile.
   c) Rmin: location of minimum value on the right profile.
   d) Rmax: location of maximum value of the right profile.
2. $W(k)$ = Width at position $k = \mathrm{RP}(k) - \mathrm{LP}(k)$.
3. Wmax, the maximum width of the digit; this is $W(k)$ at some point k where $\mathrm{RP}(k) - \mathrm{LP}(k)$ is a maximum.
4. R, the ratio of height to maximum width.
5. Based on first differences:
   a) $\mathrm{LDIF}(k) = \mathrm{LP}(k) - \mathrm{LP}(k-1)$.
   b) $\mathrm{RDIF}(k) = \mathrm{RP}(k) - \mathrm{RP}(k-1)$.

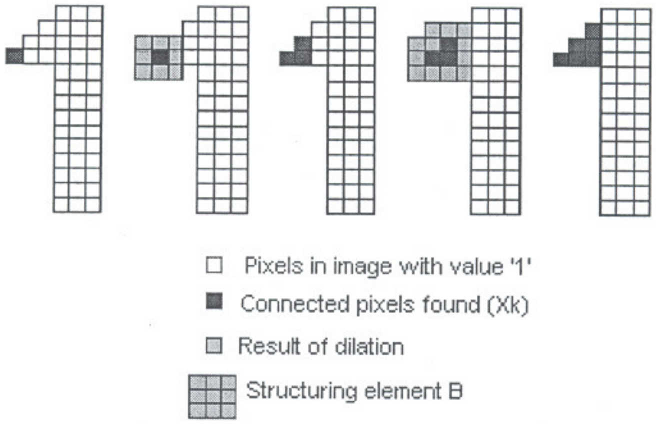


Figure 19. The first two iterations in the process [8].

Based on the above properties, 48 features are computed for each sample numeral. Please refer to [9] for the complete list of features.

In the training phase, all 48 features are computed for each sample numeral and a feature vector is created in each case. The features are binary. Then, all of the resulting bit strings for each digit are searched for common elements, and the features in common for each digit class are stored in a library. Matching is performed by extracting the profiles of the input image and measuring the bit string (feature vector). This string is matched against the common elements of the templates. This is computed very fast since only bit operations are involved. A match of a library bit string against an input string results in the corresponding digit class being assigned to the input digit. If a match is not found then the internal door count of the MATLAB program is used to make a decision.

6. The Web Interface

The web interface is an integral part of the mobile navigation and identification process. The mobile robot is connected to the Internet through an onboard WLAN 802.11b card which connects to the nearest wireless service providing access point.

The robot can be controlled and viewed from the internet, through its website: http://www.bridgeport.edu/sed/risc/html/proj/RISCBOT/index.htm. Updates on the web services and server availability information will be posted on the website. Users can also download videos and pictures of sample navigation and recognition tasks performed by the robot.

The RISCBOT web interface is simple, consisting of three windows. The interface consists of three different windows: the control window, top view window and the camera window. Figure 20 shows a view of the web interface while the robot is navigating.

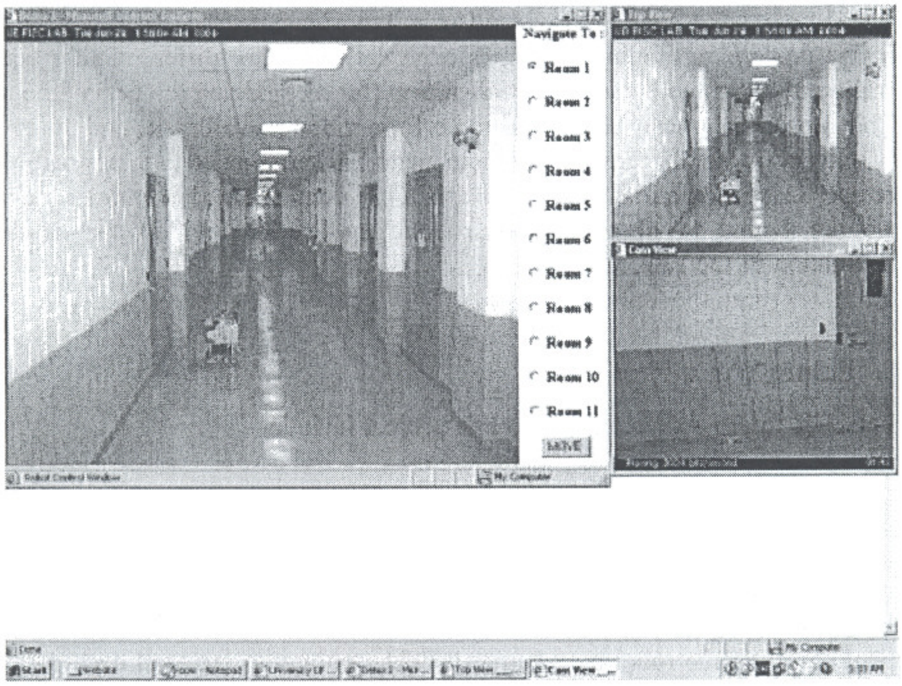


Figure 20. The RISCBOT website.

The control window shows the instantaneous position of the RISCBOT. Once logged on, any remote user can command RISCBOT to move to a desired door by selecting one of the 11 listed doors on the right of the control window. When the user presses the MOVE button the respective door number is sent to the RISCBOT server which then sends the command to the RISCBOT computer via the wireless link. The RISCBOT website is hosted on a Microsoft IIS Server supporting ASP for user interaction with the robot.

The main purpose of the other two windows; the top view window and the cam window is to provide visual feedback. The top view window shows the position of RISCBOT in the corridor. This view is the same as that on the command window. This video is provided by a network camera mounted on ceiling of the corridor.

The cam view window provides a head-on real time video feedback from the robot as it moves through the corridor. The onboard camera continuously grabs image frames from the USB camera and wirelessly transmits them to the server. This feedback has been implemented using Microsoft Media Encoder.

7. Conclusion

In this project we built a remote surveillance robot, RISCBOT, which can be monitored and controlled through the World Wide Web. The robot utilizes an image-processing module to identify and navigate to the desired room. The image-processing module combines edge detection (for differentiating between walls and doors) with character recognition (for recognizing the desired door ID). This mobile robot can be utilized to perform video surveillance, and many other functions within the areas of remote material handling, online tour guidance, remote inspection and security. With 802.11b connections becoming ubiquitous, we are bound to observe more robotic applications within web-controlled navigation in the future.

8. Future Work

Potential future enhancements to this project include

1. Introducing map building and tracking capabilities.
2. Designing and building pan – tilt units for the cameras.
3. Implementing a dc – dc (ATX power supply) converter circuit that will increase the power conversion efficiency and thereby the operational time for the robot.
4. Permitting the robot to recharge itself by plugging into wall outlets.
5. Mounting a manipulator on the mobile platform for implementing mobile manipulation tasks.

References

1. Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C. R., Roy, N., Schulte, J., and Schulz, D.: MINERVA: A second-generation museum tour-guide robot, in: *Proceedings: IEEE International Conference on Robotics and Automation (ICRA '99)*, Detroit, Michigan, USA, May 1999.
2. Burgard, W., Cremers, A. B., Fox, D., Hahnel, D., Lakemeyer, G., Schulz, D., Steiner, W., and Thrun, S.: *The Interactive Museum Tour-Guide Robot*, American Association for Artificial Intelligence (AAAI-98), Madison, Wisconsin, July 1998.
3. Goodwin, R., Haigh, K., Koenig, S., and Sullivan, J.: A layered architecture for office delivery robots, in: *First International Conference on Autonomous Agents*, Marina Del Ray, California, February, 1997.
4. Kimura, F. and Shridhar, M.: Handwritten numerical recognition based on multiple algorithms, Pattern Recogn. **24**(10) (1991), 969–983.
5. NeuroMatrix Board Support Kit: Multiprocessor Module Assembly NM42, Programmer's Manual, Rev. 01 95 02A, Module Research Centre, Russia, 2003.
6. Atmel ATM103 Datasheet, Rev. 0945SE – 01/00/xM, Atmel Corporation, USA, 2000.
7. Kobayashi, K.: *MATLAB Utilization Book*, Shuwa System Co, Ltd., 2001, ISBN 4-7980-0141-4.
8. Hansen, H. Kristensen, A. W., Kohler, M. P., Mikkelsen, A. W., Pedersen, J. M., and Trangeled, M.: *Automatic Recognition of License Plates*, Institute of Electronic Systems, Aalborg University, Denmark, May 26, 2002.
9. Parker, J. R.: *Algorithms for Image Processing and Machine Vision*, ISBN 0-471-14056-2, Wiley Computer Publishing, 1997.

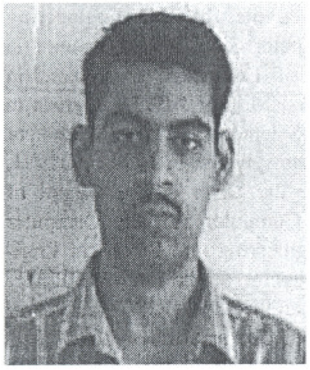
Dr. Tarek M. Sobh received the B.Sc. in Engineering degree with honors in Computer Science and Automatic Control from the Faculty of Engineering, Alexandria University, Egypt in 1988, and M.S. and Ph.D. degrees in Computer and Information Science from the School of Engineering, University of Pennsylvania in 1989 and 1991, respectively. He is currently the Dean of the School of Engineering; a Professor of Computer Science, Computer Engineering, Mechanical Engineering and Electrical Engineering and the Founding Director of the interdisciplinary Robotics, Intelligent Sensing, and Control (RISC) laboratory at the University of Bridgeport, Connecticut.

He was a Research Assistant Professor of Computer Science at the Department of Computer Science, University of Utah from 1992–1995, and a Research Fellow at the General Robotics and Active Sensory Perception (GRASP) Laboratory of the University of Pennsylvania during 1989–1991. Dr. Sobh's current research interests include reverse engineering and industrial inspection, CAD/CAM and active sensing under uncertainty, robots and electromechanical systems prototyping, sensor-based distributed control schemes, unifying tolerances across sensing, design, and manufacturing, hybrid and discrete event control, modeling and applications, and mobile robotic maniputlation. He has published over 120 refereed journal and conference papers, and book chapters in these and other areas. Dr. Sobh is a Licensed Professional Electrical Engineer (P.E.), a Certified Manufacturing Engineer (CMfgE) by the Society of Manufacturing Engineers, and a Certified Reliability Engineer (C.R.E.) by the American Society for Quality, a member of Tau Beta Pi, Sigma Xi, Phi Beta Delta, and Upsilon Pi Epsilon. Dr. Sobh was the recipient of the Best Paper Award at the World Automation Congress Conference (WAC 98). Dr. Sobh is a member or senior member of several professional organizations including; ACM, IEEE, IEEE Computer Society, IEEE Robotics and Automation Society, the National Society of Professional Engineers (NSPE), the American Society of Engineering Education (ASEE), and the Society of Manufacturing Engineers (SME).

Sarosh Patel recieved his B.E. degree in Electrical and Electronics Engineering with Distinction from the Faculty of Engineering Osmania University, India. He is currently pursuing M.S. in Electrical Engineering at the School of Engineering, University of Bridgeport, CT.

Currently working as a Research Assistant at the Interdisciplinary RISC (Robotics and Intelligent Systems Control) Lab, under Dr. Tarek Sobh, Dean School of Engineering, Univeristy of Bridgeport, and receives a full tuition wavier for the same. He had been nominated for inclusion in 2005 edition of Who's Who Among Students in American Universities.

Sarosh has also published Journal and Conference papers in the field of Robot Prototypng, Dynamic Control and Autonomous Mobile Robots. His current research interests include Robotics, Non-Linear Control Systems, Artificial Intelligence and Ethical Hacking.



Rajeev Sanyal completed his B.E. in Electrical Engineering from the Mumbai University, India in 2001 and his M.S in Electrical Engineering from the University of Bridgeport, USA in 2004. He worked as an Electrical Engineer at Hylift Equipment, India and as a Research Assistant at the RISC lab in the Univeristy of Bridgeport. Rajeev Sanyal's major research interests are mobile robotics, machine perception, analog and digital design, FPGAs, speech recognition and artificial intelligence.