View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Queensland University of Technology ePrints Archive

Singapore, 1-3 December, 2004

Vision-based Pirouettes using the Radial Obstacle Profile

Trevor Taylor
School of Information Systems
Queensland University of Technology
Brisbane, Australia
t.taylor@qut.edu.au

Shlomo Geva
School of Software Engineering and Data Communications
Queensland University of Technology
Brisbane, Australia
s.geva@qut.edu.au

Wageeh Boles
School of Electrical and Electronic Systems Engineering
Queensland University of Technology
Brisbane, Australia
w.boles@qut.edu.au

***Abstract*—Mapping algorithms commonly use "radial sweeps" of the surrounding environment as input. Producing a sweep is a challenging task for a robot using only vision.**

With no odometers to measure turn angles, a vision-based robot must have another method to verify rotations. In this paper we propose using the Radial Obstacle Profile (ROP) which gives the radial distance to the nearest obstacle in any direction in the robot's field of view. By matching the ROPs before and after a turn, the robot should be able to verify that the expected angle of rotation matches the actual angle. Combining successive ROPs then produces a radial sweep.

***Keywords*—computer and robot vision;wheeled mobile robots; mapping; radial obstacle profile**

I. INTRODUCTION

Computer vision research has been underway for over 30 years. Although there have been significant advances, the current state of the art for mobile robots is still fairly primitive [1]. Part of the reason for this is that other sensors, such as sonar and infra-red, have been used to perform obstacle detection, thereby making it unnecessary to solve some of the difficult problems in vision.

The objective of our research is to perform mapping using only vision, and to do so using cheap off-the-shelf hardware. Therefore, our robot is equipped with a single color camera as its only sensor.

In order to build accurate maps a robot must be able to track its own motions with a high degree of precision. This involves measuring both translations and rotations.

In the classical approach using Occupancy Grids [2], the robot makes radial sweeps of the surrounding environment to detect obstacles. This data is then incorporated into the map by applying Bayes Rule and a sensor model.

Sweeps are easy for a robot equipped with a ring of sonar sensors because the sensors all have known orientations. However, sonar measurements are imprecise due to the fact that the beam expands as it moves away from the robot, and so the resulting sensor model is complex. Sonar also suffers from problems caused by noise and sound-absorbing surfaces.

In contrast to sonar, vision-based sweeps have a very simple sensor model. Another advantage of vision is that color information can be used to distinguish between obstacles. However, making radial sweeps using a single camera is quite difficult because the robot must turn through a full 360º capturing images as it turns. The mapping information from these images has to be "stitched together" accurately.

Tracking the robot's position can be simplified by limiting the robot to two types of motion: forward movements and in-situ rotations. (Rotation on the spot is possible for a robot with two direct-drive wheels. Many robots commonly used in research are of this design.) This approach results in piece-wise linear motion. To produce a map the robot must know how far forward it has traveled with each move, and how far it has turned with each rotation. It should be noted that the robot is limited to this mode of operation by choice. It is perfectly capable of simultaneous translation and rotation by appropriate control of the differential drive signals.

One advantage of this approach is that the robot does not have to make real-time steering decisions – it can take as long as is necessary to decide whether to move or rotate. It also allows the use of complex segmentation methods, e.g. [3], which are very slow to compute. This eliminates the problem of limited CPU power which can be solved in other ways, but they are not the subject of this paper.

Cumulative errors are known to present problems for robots using odometers [4]. This is why localization and mapping are normally performed together, i.e. Simultaneous Localization and Mapping (SLAM).

Errors in the estimates of the distance traveled in forward motions will result in a scaling of the map, assuming the errors are consistent. For instance, if the robot determines that it has moved 5cm but it has only moved 4.5cm, then the map will be incorrect by a factor of 10%. However, if there are errors in estimating the rotations, this will distort the map and could even cause the robot to lose its position. Therefore it is vital that the angle of rotation be determined as accurately as possible.

To allow the robot to confirm the amount of a rotation, the Radial Obstacle Profile (ROP) is used. The ROP is calculated for use in the Mapping algorithm as a form of "visual sonar", and is constructed as follows:

0-7803-8645-0/04/$20.00 © 2004 IEEE

1. Determine which parts of the image constitute the floor (as distinct from obstacles);

2. Calculate the boundary between floor and obstacles, called the obstacle profile;

3. Convert the obstacle profile into radial coordinates using an Inverse Perspective Mapping to obtain the ROP.

Note that the obstacle profile can also be converted into real-world Cartesian coordinates for a "top-down view". This is much easier for a human operator to comprehend, and it is useful when making forward motions, but it is not the main subject of this paper.

Many systems that use vision for navigation, e.g. [5,6,7,8], take the approach of identifying the floor (also called the ground plane or free space) as the first step. An approach based primarily on edge detection is effective if the floor is of a reasonably uniform color [9].

These systems all assume a planar floor of uniform color, which is the case for many office environments. In our previous work [10] we outlined one method for handling the color variations that normally occur in images due to illumination. (This is an active field of research which is referred to as color constancy.) Other research [11] has relied on training the robot to distinguish between floor pixels and all other objects, but this still requires distinct colors between the floor and obstacles. Even in commercial products [12], the limitation of a uniform floor color still exists.

In this particular research a single camera (i.e. monocular vision) is used for reasons of cost, size and weight, but other researchers have used binocular (stereo) vision [13,14] and even triocular vision [15]. Panoramic cameras have also been used [16], but they are more expensive to build and image processing is complicated due to radial distortion introduced into the image.

For now, assume that the floor can be identified somehow – the exact method is not important to this discussion.

Calculating the ROP involves an Inverse Perspective Mapping and conversion to real-world polar coordinates. For a given camera geometry, the process of converting the obstacle profile to polar coordinates can be performed quickly and easily via a table lookup, as explained in [10]. The final step is to separate the data into "accumulator bins" with a particular granularity, e.g. one degree of arc.

The image resolution determines the ROP granularity that is possible without interpolation. For our camera with a 60º field of view and a 320 x 240 pixel resolution, the ROP can be in increments of about a fifth of a degree. To date we have used a one degree increment for simplicity.

The ROP has the useful property that, because it is a simple linear array, any rotation of the robot is equivalent to sliding the array elements to the left or the right (depending on the direction of rotation). Therefore, two successive ROPs can be compared to determine the amount by which the robot has rotated by sliding one array across the other and calculating a dissimilarity measure at each step. The location of the minimum should correspond to the angular disparity (angle of rotation) between the two ROPs.

Once the relationships between successive ROPs have been established, multiple ROPs can be combined to produce a full radial sweep. In effect, the robot performs a vision-based pirouette to build up a complete sweep based on ROP information.

II. MEASURING ROTATIONS

The two basic motions that our robot is allowed to make are rotations and forward moves (translations). This paper is concerned only with rotations.

The camera Field of View (FOV) sets a limit on how far the robot is allowed to turn in a single rotation. In order to have a reasonable area of overlap between successive images, the robot should not turn any more than half of the FOV. For our camera, with a FOV of 60º, we found that it is reasonable to turn no more than 20º to 30º in a single rotation so that there is a good overlap between images. The FOV therefore becomes a limiting factor in applying this work. Although it would be possible to use a camera with a wide-angle lens, or even an omni-directional camera, one objective of this research is to use simple, cheap, readily available components.

A. Obtaining Range Data from Video Images

The first step in mapping is to distinguish between free space and obstacles. There are various methods for doing this as outlined above. In essence, this is a segmentation problem. The output from this step is a Floor Map, an example of which is shown in Fig. 1 below.

From the Floor Map, the boundary between the floor and surrounding obstacles (the obstacle profile) can be determined for each vertical column in the image. Horswill [5] referred to this as a Radial Depth Map, but it is neither Radial nor a true Depth map. Although depth (distance from the robot) is directly related to the height of a pixel in an image, this relationship is highly non-linear. Furthermore, Horswill simply used the pixel coordinates of the boundary, which are not radial coordinates, although it is a reasonable first approximation and it worked well in his application.

Video images are a two-dimensional view of a three-dimensional world. The process of capturing an image introduces a perspective transformation. In order to perform reliable mapping, the information in the images needs to be converted back into real-world coordinates through an Inverse Perspective Mapping. This allows a map to be produced that shows a top-down view. Badal et al. [17] referred to this as an Instantaneous Obstacle Map (IOM), but they used it for reactive navigation, not for map building.

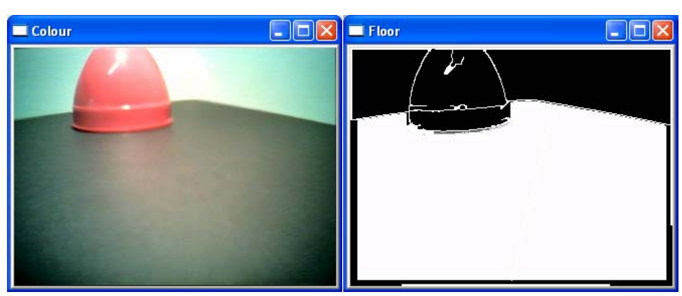

Figure 1. A Robot View and corresponding Floor Map.

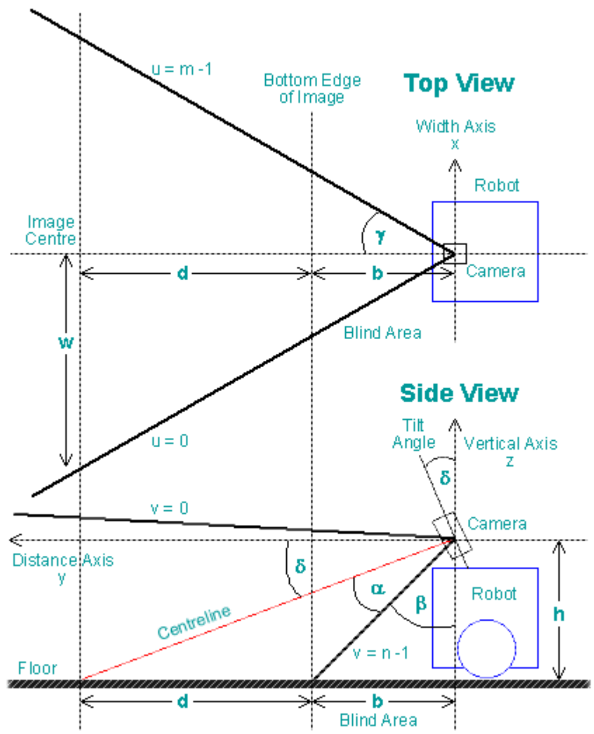

Figure 2. Coordinate System and Parameters for IPM.

Assume that the camera is mounted in the center of the robot and is at the origin of a world coordinate system relative to the robot. It is possible to do an Inverse Perspective Mapping (IPM) [10] as shown in Fig. 2.

Four parameters must be measured for the IPM calculation (see Fig. 2 above):

b – Blind area between the camera (in the center of robot) and the point visible in the bottom scanline of the image;

h – Height of the camera off the floor;

d – Distance along the floor between the bottom scanline and the center scanline of the image; and

w – Distance across the floor measured from the robot centerline to the edge of the image at the center scanline of the image.

In Fig. 2 above, α is one-half of the vertical FOV; γ is one-half of the horizontal FOV; and δ is the camera tilt angle. (γ and α are related through the aspect ratio of the camera, which is usually 4:3 for conventional video cameras.)

Image pixel coordinates are represented by (u,v) in the horizontal and vertical directions respectively. Note that, by convention, v is measured down from the top of the image. The camera resolution is m by n pixels.

The real-world coordinates (x,y) can then be calculated as follows:

$$\beta = \tan^{-1}\left(\frac{b}{h}\right), \tag{1}$$

$$\alpha = \tan^{-1}\left(\frac{b+d}{h}\right) - \beta, \tag{2}$$

$$\gamma = \tan^{-1}\left(\frac{w}{b+d}\right), \tag{3}$$

$$y = h\tan\left(\beta + \left(\frac{2\alpha(n-1-v)}{(n-1)}\right)\right), \tag{4}$$

$$x = y\tan\left(\gamma\left(\frac{(2u-m+1)}{(m-1)}\right)\right). \tag{5}$$

A lookup table can be constructed that transforms image pixel coordinates to world coordinates. Similarly, a lookup table can be created that maps pixel coordinates to polar coordinates, i.e. the radial distance from the robot's center and the angle measured from the robot's centerline.

B. *Vision-based Pirouettes*

The calculations outlined above allow the ROP to be obtained. It is then possible to compare successive ROPs as the robot turns. In our case, the robot can be commanded to turn through a specified angle, but the turns are based on experimentally calibrated amounts of time required to turn the robot through given angles at particular wheel speeds. This means that turns are not completely reliable because the time intervals cannot be measured or executed with high precision.

To verify the amount of a turn, the ROP following a turn can be slid across the ROP obtained prior to the turn looking for the best match. By comparing the ROPs it is possible to determine the number of degrees of difference between the two. However, there are some complications.

The amount of overlap between ROPs might be small. As one ROP slides across the other, any portions that are not in the field of view of both images cannot be compared. This means that the number of elements being compared might be small, reducing the reliability of this process. The number of elements available for comparison is affected by the angle of rotation as a portion of the camera FOV, and the granularity of the ROP, which is in turn determined by the camera resolution.

A further complication is that the robot might not turn exactly on the spot. Objects at large distances from the robot can change position in successive images by as little as one vertical pixel (due to perspective). The resulting offset (after Inverse Perspective Mapping) causes significant differences in the ROP between two images, even though a human operator would readily recognize that the two ROPs exhibit the same profile. This is the subject of ongoing research.

Once the angles between successive images have been determined, the ROPs can be combined. Combining all of the ROPs for a full 360 degrees of rotation results in a "radial sweep", which is similar to the classical sonar sweep.

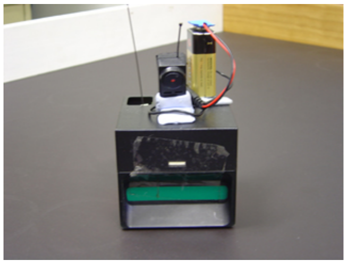

Figure 3. Yujin Robot with Wireless Camera.

III. Experimental Results

Our work so far has involved robots that are small – about 10cm in diameter. Consequently, they are limited in what they can carry, and too small for effective stereo vision. The robots are equipped with wireless color video cameras that run off 9V batteries. These cameras are small, light and cheap. Fig. 3 shows a Yujin soccer robot with a camera attached.

Video images are captured using a USB capture device attached to a PC at a resolution of 320 by 240 in 24-bit color. The PC controls the robots remotely via a wireless link.

The robots operate on custom-built “playing fields”. These are similar to robot soccer fields. The floor is of a uniform color (matte black) and can be distinguished from the walls and other obstacles placed on the field. Even with the supposedly uniform floor color, lighting conditions affect what is seen by the camera and the pixel values for the floor can vary widely. This is quite apparent in the robot’s view in Fig. 1 above where the bottom left and right corners of the image are almost black. Edges are therefore important as well.

Initially the robots were programmed to use an algorithm that drove them towards the maximum amount of free space that was visible, i.e. towards the maxima in the ROPs. If the robot became trapped, it would back up and turn 90º to the left or right. This simple behavior, effectively a random walk, will soon be replaced by a more complex exploration and mapping algorithm.

A. *Radial Obstacle Profiles*

Radial Obstacle Profiles can be calculated once the parameters for the IPM calculations are known. When the robot control program starts up, it builds a lookup table to translate image pixel coordinates to polar world coordinates. The robot is then commanded to capture successive images as it turns around. After each turn, the ROP is calculated.

In addition to the ROP, a top-down view is also calculated, also using a lookup table. This is drawn onto the Sweep map. Fig. 4 below shows the ROP and Sweep diagrams for the robot view from Fig. 1. This is only one step in a full sweep.

The ROP diagram (Fig. 4) shows the radial distance from the robot as a function of the angle in the field of view. Each horizontal grid line represents 10cm in the real world, and the vertical grid lines are 10 degrees apart. Note that plotting the viewing angle in this way on a linear scale results in what might be perceived as a distortion of the obstacle profile because straight edges become curved.

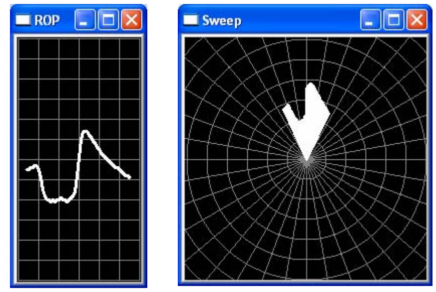

Figure 4. Example of a Radial Obstacle Profile and Radial Sweep.

The Sweep diagram is a top-down view, but with a radial grid. The grid markings are at the same intervals as in the ROP diagram. The “V” shape at the base of the floor area is a consequence of the FOV of the camera, and is very close to the 60º in the camera specifications. The hollow in the map is the obstacle that is visible in Fig. 1. Notice that there is a “shadow” behind the obstacle because the robot cannot see this area from its vantage point. To completely map the area, the robot would have to move to a different location, and this exploratory behavior is the subject of further research.

In practice, it has been found that range estimates become unreliable beyond a certain distance (between 60cm and 100cm), depending on the tilt angle of the camera. This is due to the effects of the perspective transformation which means that a difference of a single vertical pixel in the image is equivalent to a very large distance in the real world. In fact, it is possible for the radial distance to range up to infinity well before the top of the image is reached. This happens if the camera can see above the horizon, as per Fig. 2 above.

When it comes to mapping, it should be noted that if the range exceeds the maximum reliable distance as set in the software, then this should not be recorded as an obstacle. This is a “soft” boundary.

B. *Radial Sweeps*

In preliminary work with the ROPs we have been able to construct radial sweeps by assuming that the robot turns by exactly the requested amount. Two issues have been encountered when comparing ROPs: the registration between successive ROPs is sometimes not good enough for accurate matching; and the segmentation process can occasionally produce outliers which introduce spikes into the ROP.

Another difficulty with matching successive ROPs is that large open spaces do not give sufficient detail for comparison. In effect, this is an aperture problem similar to the familiar problem in optic flow when motion is perpendicular to the line of vision.

Fig. 5 below shows how the program can combine all of the top-down views from a dozen images taken at 30º intervals to build a full 360º sweep. The sweep (the gray area) has been overlaid on a picture of the actual playing field. In this sweep, the maximum visual range was set so that it covered the entire field. (We usually make the robot more “near sighted” than this so that it cannot see the whole field in one sweep.)

Figure 5. A full Radial Sweep overlaid on the Playing Field.

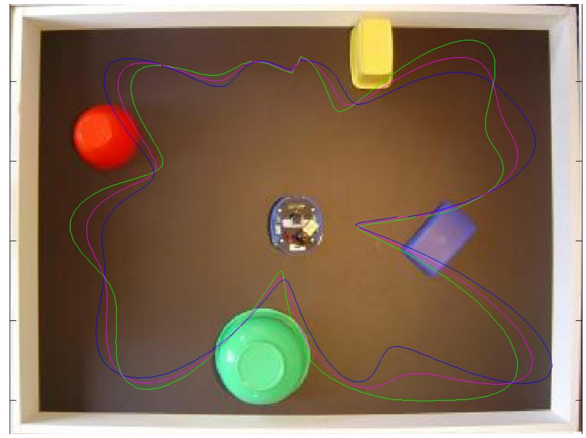

Figure 6. DCT of the ROP overlaid on the Playing Field.

The registration of the sweep with the playing field is not perfect. This is partially an artifact of the diagram due to the difficulty of matching the overhead image to the scale of the data in the map. It is apparent, however, that the robot has identified the basic layout of its environment, at least to the accuracy necessary to start moving around without colliding with obstacles.

Rotation was counter-clockwise in this example, with the robot starting out by facing towards the top center of the playing field. The robot's estimates of the wall position became progressively worse as it turned. Some of this is due to the fact that the camera cannot be exactly centered on the robot, but also the robot does not turn exactly on the spot.

Other minor issues are that the map overlaps the obstacles slightly, and there is an outlier visible in the bottom left-hand corner (the sharp indentation in the map). Clearly it will require multiple sweeps over a period of time to produce an accurate map. Another avenue being investigated is averaging multiple images before processing.

IV. FUTURE RESEARCH

At the time of writing, the robots can wander at random and avoid obstacles. They can also perform 360º sweeps on command. These sweeps have not yet been incorporated into a SLAM system. For full mapping of the environment, the robots will also have to do some path planning in order to move towards unexplored areas.

The determination of the obstacle profile assumes that the floor has a uniform color. Outliers sometimes occur due to reflections on the floor, shadows produced by uneven illumination, pieces of lint or scratches, etc. The problem is aggravated by the low light situations that are common in a typical office environment. These outliers produce spikes or distortions in the ROP which make it difficult to match with the next ROP. A simple smoothing filter (an average over three ROP array elements) has been applied to reduce the problem of outliers, but this does not eliminate it entirely.

If no suitable match can be found, i.e. within a reasonable range of the expected angle, the robot has to assume that there is some problem with the ROP and that it has in fact turned by the anticipated amount.

One approach that we are investigating is taking the Discrete Cosine Transform (DCT) of the ROP. Dropping the higher-frequency coefficients is equivalent to a low pass filter which removes the outliers.

The DCT is phase-sensitive. In other words, if a signal is shifted slightly then its DCT will be different. Therefore, comparing two successive ROPs can be done by shifting one and comparing the DCT coefficients.

We have also tested a slightly different approach where the odd and even ROPs are combined into two full sweeps. This is possible because the amount of a rotation is half of the FOV, i.e. concatenating alternate ROPs makes up a full sweep.

Fig. 6 shows three curves: the odd ROPs, even ROPs and the average of the two. They have been obtained by taking the DCT, truncating the result to only 24 coefficients, and then applying the inverse DCT. The resulting curves do not fit the actual obstacle profile well, but this is not the intention. We expect that the overall shape will be representative of the obstacles in the local environment, and possibly sufficient to allow localization, e.g. using a Self-Organizing Map (SOM).

One significant advantage of using the DCT is that the amount of memory required to save the coefficients, i.e. the description of the view from a particular location, is several orders of magnitude less than storing the original images or even the full ROP.

Another area that we are investigating is using color to assist in comparing ROPs. The color of the obstacles at each point in an ROP can be recorded. However, it is not simply a matter of saving the RGB pixel value because this suffers from the same color constancy problem as the uneven floor color. Some processing of the color will be necessary in order to ensure that obstacles viewed from different positions are still recorded as having the same color.

Color information could also be used in the Occupancy Grid to help with localization. When the robot is looking for a match to its current pose, it might significantly reduce the search space by only looking for objects that are of a similar color to those that it can currently see.

V. CONCLUSION

This paper has explained how to calculate the Radial Obstacle Profile and shown how it might be used for visually verifying the amount of a rotation.

The ROP allows the creation of "radial sweeps" for mapping. Comparing successive ROPs is one means for a robot to visually confirm the amount of an in-situ rotation. This is very important for producing accurate maps of the robot's surrounding environment and maintaining its pose when performing a mapping task.

More work is still required in the areas of comparing and storing ROP information.

REFERENCES

[1] A. C. Kak and G. N. DeSouza, "Robotic vision: what happened to the visions of yesterday?," *Proc. 16th Int'l Conf. Pattern Recognition* (ICPR), 2002.

[2] H. P. Moravec and A. Elfes, "High-resolution maps from wide-angle sonar," *Proc. IEEE Int'l Conf. Robotics and Automation* (ICRA), St. Louis, MI, 1985.

[3] D. Comaniciu and P. Meer, "Mean shift analysis and applications," *Proc. Int'l Conf. Computer Vision* (ICCV), 1999.

[4] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. Robotics and Automation*, vol. 12, pp. 869-880, 1996.

[5] I. D. Horswill, "Visual collision avoidance by segmentation," *Proc. Image Understanding Workshop*, 1994.

[6] G. Cheng and A. Zelinsky, "Real-time visual behaviours for navigating a mobile robot," *Proc. IEEE Int'l Conf. Intelligent Robots and Systems* (IROS), 1996.

[7] L. M. Lorigo, R. A. Brooks, and W. E. L. Grimson, "Visually-guided obstacle avoidance in unstructured environments," *Proc. IEEE Int'l Conf. Intelligent Robots and Systems* (IROS), vol. 1, pp. 373-379, 1997.

[8] A. Howard and L. Kitchen, "Fast visual mapping for mobile robot navigation," *Proc. IEEE Int'l Conf. Intelligent Processing Systems*, Beijing, China, 1997.

[9] M. C. Martin, "Visual obstacle avoidance using genetic programming: first results," *Proc. Genetic and Evolutionary Computation Conf.* (GECCO), 2001.

[10] T. Taylor, S. Geva, and W. W. Boles, "Monocular vision as a range sensor," *Proc. Int'l Conf. Computational Intelligence for Modelling Control and Automation* (CIMCA), CD-ROM, Gold Coast, Australia, 2004.

[11] S. Lenser and M. Veloso, "Visual sonar: Fast Obstacle Avoidance using Monocular Vision," *Proc. IEEE Int'l Conf. Intelligent Robots and Systems* (IROS), 2003.

[12] "ER1 User Guide," Evolution Robotics, Inc., Pasadena, CA, 2003.

[13] D. Murray and J. J. Little, "Using real-time stereo vision for mobile robot navigation," *Autonomous Robots*, vol. 8, pp. 161-171, 2000.

[14] D. Burschka, S. Lee, and G. Hager, "Stereo-based obstacle avoidance in indoor environments with active sensor re-calibration," *Proc. IEEE Int'l Conf. Robotics and Automation* (ICRA), Washington, DC, 2002.

[15] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, pp. 333-349, 1997.

[16] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor, "Omni-directional vision for robot navigation," *Proc. IEEE Workshop Omnidirectional Vision*, South Carolina, 2000.

[17] S. Badal, S. Ravela, B. Draper, and A. Hanson, "A practical obstacle detection and avoidance system," *Proc. Workshop Applications of Computer Vision* (WACV), Sarasota, FL, 1994.