Autonomous Robots 8, 161–171 (2000)

Using Real-Time Stereo Vision for Mobile Robot Navigation

DON MURRAY AND JAMES J. LITTLE
Computer Science Department, University of British Columbia, Canada
donm@cs.ubc.ca; little@cs.ubc.ca

Abstract. This paper describes a working vision-based mobile robot that navigates and autonomously explores its environment while building occupancy grid maps of the environment. We present a method for reducing stereo vision disparity images to two-dimensional map information. Stereo vision has several attributes that set it apart from other sensors more commonly used for occupancy grid mapping. We discuss these attributes, the errors that some of them create, and how to overcome them. We reduce errors by segmenting disparity images based on continuous disparity surfaces to reject "spikes" caused by stereo mismatches. Stereo vision processing and map updates are done at 5 Hz and the robot moves at speeds of 300 cm/s.

Keywords: stereo vision, mobile robot navigation

1. Introduction

Perception is a crucial part of the design of mobile robots. We want mobile robots to operate in unknown, unstructured environments. To achieve this goal, the robot must be able to perceive its environment sufficiently well to allow it to operate safely within that environment.

Most robots that navigate in unconstrained environments use sonar transducers or laser range sensors as their primary spatial sensor (Nickerson et al., 1993; Dudek et al., 1997a, b; Brooks et al., 1988; Brooks, 1987). Mobile robots often use computer vision, but usually for feature tracking or landmark sensing and not often for occupancy grid mapping or obstacle detection.

In this paper, we present a working implementation of a robot that uses correlation-based stereo vision and two-dimensional occupancy grid mapping to navigate and autonomously explore unknown and dynamic indoor environments. Stereo vision mapping is very sensitive to errors, as the process of collapsing the data from 3D to 2D encourages errors in the form of "spikes" that are propagated into the map. We examine the characteristics of correlation-based stereo that give rise to these errors and make some suggestions on how to overcome them and improve the reliability of the resultant maps. Several examples of stereo range sensing are given, as well as some autonomously generated occupancy grid maps.

2. Architecture

2.1. Mobile Robot: José

We used a Real World Interfaces (RWI)[1] B-14 mobile robot, *José* (shown in Fig. 1), to conduct our experiments in vision-based robot navigation. *José* has a Pentium™ PC running the Linux operating system as its onboard processing. This robot is a significant improvement over *Spinoza*, our other mobile robot that was reported in Tucakov et al. (1997). *Spinoza* uses embedded processors exclusively and, although it is a powerful system, it proved to be a difficult development environment. *José* is equipped with an Aironet ethernet radio modem that allows communication to a host computer, as well as a Matrox Meteor RGB frame grabber connected to a Triclops trinocular stereo vision camera module.

The Triclops stereo vision module was developed at the UBC Laboratory for Computational Intelligence (LCI).[2] The stereo vision module has 3 identical wide-angle (90° degrees field-of-view) cameras. The

Figure 1. *José*, the mobile robot.

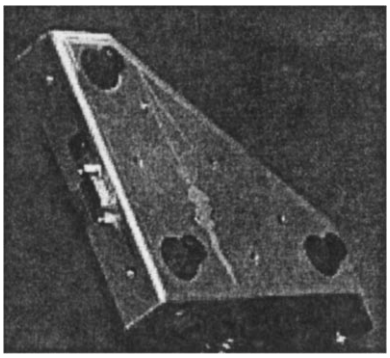

Figure 2. The Triclops stereo head.

system is calibrated using Tsai's approach (Tsai, 1987). Correction for lens distortion, as well as misalignment of the cameras, is performed in software to yield three rectified images. These rectified images conform to a pinhole camera model with square pixels. The camera coordinate frames are co-planar and aligned so that the the epipolar lines of the camera pairs lie along the rows and columns of the images. The Triclops stereo head is shown in Fig. 2.

2.2. Trinocular Stereo Algorithm

The trinocular stereo approach is based on the multibaseline stereo developed by Okutomi and Kanade (1993). Each pixel in the reference image is compared with pixels along the epipolar lines in the top and left images. The comparison measure used is sum of absolute differences (SAD). The results of the two image pairs (left/right, top/bottom) are summed to yield a combined score. Multibaseline stereo avoids ambiguity because the sum of the comparison measures is unlikely to cause a mismatch—an erroneous minimum in one pair is unlikely to coincide with an erroneous minimum in another pair.

The disparity results are validated in two ways. First, there is a "sufficient texture" test. This test checks that there is sufficient variation in the image patch that is to be correlated by examining the local sum of the Laplacian of Gaussian of the image. Low texture areas score low in this sum. If there is insufficient variation the results will not be reliable, thus the pixel is rejected because there will be too much ambiguity in the matches. Secondly, there is a "quality of match" test. The value of the score is normalized by the sum of absolute values of the patch surrounding the pixel. If the result is above a threshold, the match is consider to be insufficiently distinctive and therefore has a high probability that it is mismatch. This kind of failure generally occurs in occluded regions where the pixel cannot be properly matched.

Both these validation methods are tunable via thresholds, trading off between quality and quantity of data. If the thresholds are set too high, only the most reliable data is accepted and thus much of the image is invalidated. Despite the reliability of this data, its coverage may be too sparse to be useful. If the thresholds are set too low, there will be excellent coverage throughout the disparity image, but this will include noisy data that will create errors in the map.

3. Stereo Vision and Occupancy Grids

3.1. Review of Occupancy Grids

Occupancy grid mapping, pioneered by Moravec and Elfes (Moravec and Elfes, 1985; Elfes, 1989), is the most widely used robot mapping technique because it is simple and robust and also because it is flexible enough to accommodate many kinds of spatial sensors. It also adapts well to dynamic environments. An occupancy grid divides the environment into a discrete grid and assigns each grid location a value related to the probability that the location is occupied by an obstacle. Initially, all grid values are set to a 50% value (i.e., equal probability for occupied and unoccupied). Sensor readings supply uncertainty regions within which an obstacle is expected to be. The grid locations that fall within these regions of uncertainty have their values increased while locations in the sensing path between the robot and the obstacle have their probabilities decreased.

3.2. Stereo Sensor Model

To apply the occupancy grid method one must have a sensor model. Matthies and Grandjean (1987) provide a rigorous investigation of range sensing with stereo vision. They found disparity estimates to have Gaussian distributed random errors with standard deviations as small as 0.05 pixels. These standard deviations were consistent over different resolutions of images. While it is true that they used a different comparison score (sum of squared differences), cameras and calibration, their results show the magnitude of accuracy that can be achieved through careful correlation stereo vision. Our own experiments with sub-pixel interpolation indicate that the Triclops stereo vision module produces results with standard deviations well below one pixel. However, for real-time considerations, we have not used sub-pixel interpolation in our stereo algorithm. Due to the resulting quantization of the disparity we can approximate our stereo model by the following:

$$\begin{aligned} P(d \mid Z) &= 1 \quad \texttt{for } Z = Z(d+0.5) \rightarrow Z(d-0.5) \\ &= 0 \quad \texttt{otherwise} \end{aligned} \tag{1}$$

For our stereo vision system, with aligned optical axes and therefore focus at infinity, the relation of disparity to depth is given by

$$Z(d) = \frac{fB}{d}$$

where Z is the depth, f is the focal length of the cameras, B is the baseline between the cameras and d is the disparity. The basis for expanding this 1D approximation into 2D is illustrated in Fig. 3. This figure represents the intersection of the lines-of-sight (LOS) for individual pixels in stereo pair of cameras. One can see that for a given pixel i in the right (or reference) camera and a disparity result d from stereo matching, a 2D position can be determined by triangulation. In addition to the 2D position at the intersection of the LOS of the centres of the two pixels, there is also a "diamond" around this position which can be taken as an error bound. This resembles the elliptical region used as a model in Matthies and Shafer (1987). Given Eq. (1) we determine the region of uncertainty for a given pixel and disparity as shown in Fig. 4. The corners of the trapezoidal region of uncertainty can be found by calculating $Z = Z(d \pm 0.5)$ and then determining $X = \frac{(x \pm 0.5)Z}{f}$ where x is the image plane coordinate along the rows of the image and f is the focal length. The clear region in which an obstacle should not appear is the triangle formed by the robot's position and the closest two corners of the trapezoid.

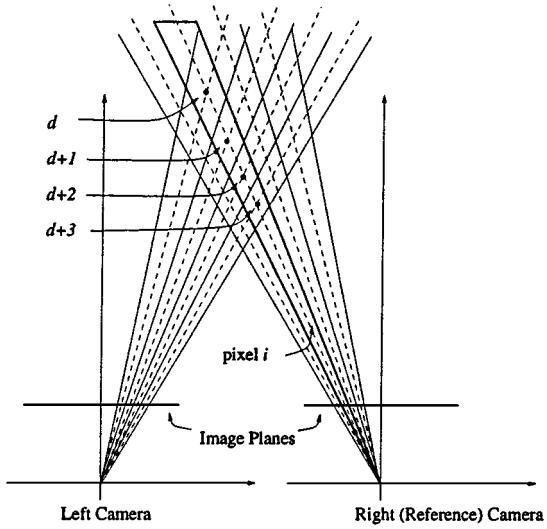

Figure 3. Stereo triangulation: dashed lines indicate lines-of-sight from the centre of the pixels, solid lines indicate lines of sight at the edges of the pixels.

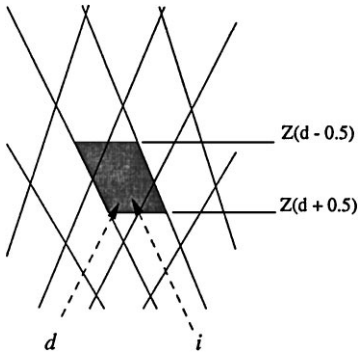

Figure 4. Region of uncertainty for a given pixel i and disparity d.

3.3. Constructing Top-Down Views from Stereo Images

Although occupancy grids may be implemented in any number of dimensions, most mobile robotics applications (including ours) use 2D grids. While stereo data provides information about the world in 3D, much

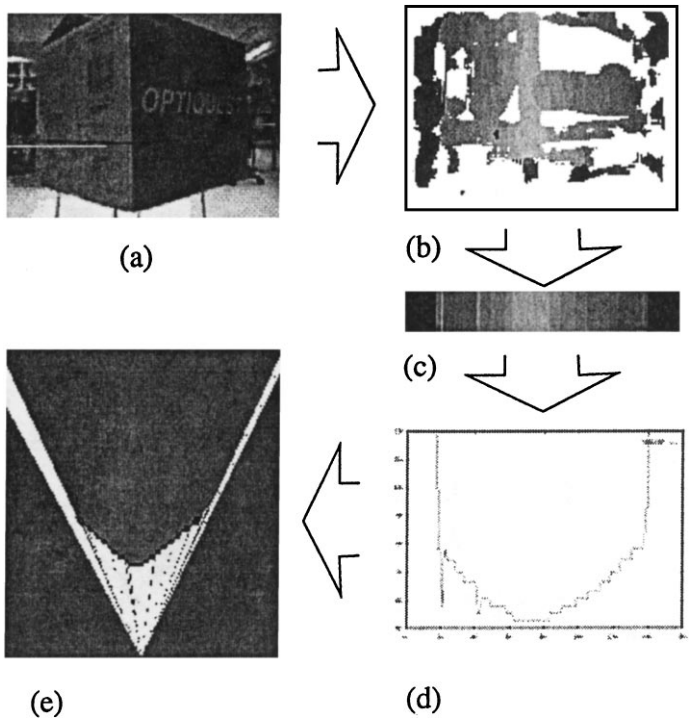

Figure 5. From stereo images to top-down views: (a) grayscale image (b) disparity image (white indicates invalid, otherwise brighter indicates closer to the cameras) (c) the maximum disparity in each column of the disparity image (d) depth versus column graph (depth in centimetres) (e) the resultant estimate of clear, unknown and occupied regions (white is clear, black is occupied and gray is unknown).

of this data is lost in the construction of a 2D occupancy grid map. This reduction in dimension is justified since indoor mobile robots fundamentally inhabit a 2D world. The robot possesses 3 DOF (X, Y, heading) within a 2D plane corresponding to the floor. The robot's body sweeps out a 3D volume above this plane. By projecting all obstacles within this volume to the floor, we can uniquely identify free and obstructed regions in the robot's space.

Figure 5 shows the construction of the 2D occupancy grid sensor reading from a single 3D stereo image. Figure 5(a) shows the reference camera grayscale image (160×120 pixels). The resulting disparity image is shown in Fig. 5(b). The white regions indicate areas of the image which were invalidated and thus not used. Darker areas indicate lower disparities, and are farther away from the camera. Figure 5(c) shows a column-by-column projection of the disparity image, taking the maximum valid disparity in each column. The result is a single row of maximum disparities. These represent the closest obstacle in each column. Figure 5(d) shows these column values converted into distance, and (e) shows these distance values converted into an occupancy grid representation, with black indicating the uncertainty region around the object, and white indicating regions that are clear. Notice the two "spikes" on the left hand portion of Fig. 5(d). These were caused by mismatches in the stereo algorithm and their causes and removal will be discussed in Section 4.1.

The process illustrated in Fig. 5 generates the input into our stereo vision occupancy grid. The mapping system then integrates these values over time, to expand the map and keep it current in the changing world.

3.4. Map Updating

The theory for updating occupancy grid maps according to a probabilistic framework is given in Elfes (1989), Konolige (1997) and others. We adopted a much simpler approach because:

1. it is computationally simpler
2. stereo errors that affect mapping are mostly systematic (as discussed in Section 4.1) and not modeled well with a probabilistic framework

3. the error in the sensor readings mainly depend upon the robot's position, the surface texture, as well as other effects. As a result the errors are not independent between sensor readings.
4. we decided to try a simpler approach first and move to the more complicated one only if the simple one failed.

The update rule was simply

$$\texttt{if } i \in \texttt{OCC}(r) \quad G(i) = G(i) + K$$
$$\texttt{if } i \in \texttt{CLEAR}(r) \quad G(i) = G(i) - K$$

where i is an occupancy grid location, r is the sensor reading, $\texttt{OCC}(r)$ represents the region of uncertainty around the sensed obstacle, $\texttt{CLEAR}(r)$ represents the clear region between robot and the sensed obstacle, $G(i)$ represents the current grid value at location i and K is some constant. The values of $G(i)$ are clipped between some $G_{\max}$ and $G_{\min}$ values. In our implementation, $G(i)$, the occupancy grid value, ranges from 0 to 255. Our 50% value indicating an unknown grid cell is determined by $(G_{\max} - G_{\min})/2$. We selected $K = 20$ to have sufficient speed in map updating. For planning purposes, we selected conservative values of $G(i) < 50$ being free of obstacles, and $G(i) > 150$ being definitely obstructed.

This update rule provides a simple linear transition between occupied and unoccupied values. It is not overly sensitive to erroneous readings, yet transitions quickly in the presence of dynamic objects. An unfortunate side effect is that good quality data tends to be smothered by poor quality data when an area in the map that was viewed from close range is later viewed from long range. We apply a heuristic, given in Section 4.3, to remove this side effect.

4. Improving Stereo Vision Mapping

4.1. Characterizing Stereo Errors

The mapping approach described so far is very sensitive to noise. As described in Section 3.2, the depth estimate distribution about the true depth of properly matched pixels is of sufficiently low deviation to be swallowed up by our quantization. However, noise due to stereo mismatches is a serious problem. Indoors scenes containing specular surfaces, repetitive patterns, and time-varying light sources can cause errors that are more or less uniformly distributed across the disparity range of the stereo system. These mismatches can be reduced by validation through comparing left-to-right and right-to-left best matches (Fua, 1993), the validation approaches described in Section 2.2 or by increasing the number of cameras in a multibaseline system (Kanade et al., 1996). However, even with a trinocular stereo system, these errors will appear. They generally appear as "spikes" in the disparity image where the disparity is much larger than the correct value; these can drastically affect the quality of the map. Errors where the disparity is erroneously low create narrow regions of clear area but these do not seriously affect mapping because they are too narrow to traverse.

To overcome the problem of spikes we have tried common image filtering techniques such as median filtering or morphological filtering. These filters can improve the results, but not as significantly as we would like. These techniques fail because they aim to remove or reduce noise from inputs that have unstructured or evenly distributed noise, i.e., noise that appears randomly and consequently will appear most often as single pixels in the image. In disparity images errors often do not have these qualities. Errors in stereo matching occur when two similar image features are in proximity to each other. The algorithm may match one of the features in one image to the wrong feature in the other image. A classic example of this problem is the so-called "picket fence" problem. When looking at a picket fence with binocular stereo, there will be regularly spaced, nearly identical image patches that can be mismatched. This causes several "ghost" fences to appear between the true fence and the robot. Trinocular stereo largely overcomes this problem, but pathological situations still do occur regularly in the unstructured world, especially in man-made environments.

If these kinds of errors occur, they will generally occur over all patches containing the incorrectly matched feature. Thus, noise does not appear in isolated pixels but in dense, connected regions. These coherent errors will confound the above filtering approaches as they will appear as a stable signal, one to be preserved instead of rejected.

As well, filters such as described above may remove valid features if those features are insufficiently thick. This can be a problem when there are thin objects in the scene such as poles or table edges. For low-resolution stereo these objects often appear only one or two pixels wide and may be removed as not sufficiently stable.

4.2. Spike Removal

We developed an approach using surface segmentation to overcome the problem of noise rejection for coherent errors as described in the previous section. To remove spikes caused by feature mismatches, we take into account the attributes of these errors: they are locally stable, but not large, and have no support from surrounding surfaces; they are genuine spikes with sharp disparity discontinuities at all borders. True surfaces in the stereo image should be not only locally consistent, but globally part of a larger 3D surface. By segmenting the image into continuous disparity surfaces, we can establish a good hypothesis based on the size of the surface whether it is a real 3D surface or a noise artifact. To segment the image into surfaces of continuous disparity we apply the following logic:

$$i = L \texttt{ given } \begin{array}{l} j \in N(i) \\ j = L \\ |d_j - d_i| \leq 1 \end{array}$$

where i is any given pixel, L is a surface label, $N(i)$ is a neighborhood of pixels around i and d_i is the disparity value at location i. Entire surfaces are invalidated from the disparity image if the number of pixels that have a given label do not pass a threshold.

This approach has two significant benefits: it can reject cohesive spikes that may fool noise rejection filters; and it can preserve thin structures that are part of a coherent structure.

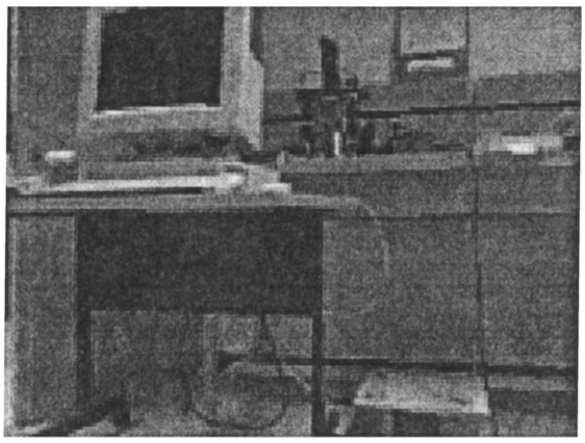

Figure 6. An example scene.

Figure 6 shows an example scene that has a number of spikes within its stereo output. The effectiveness of our spike removal approach is shown in Fig. 7. In (a) and (d), the raw disparity image contains many disparity spikes that corrupt the resulting map. The morphologically filtered disparity image shown in (b) performs marginally better. The map shown in (f), constructed from the surface segmentation approach, has a cohesive and more accurate representation of the actual scene.

4.3. Accuracy Preservation

A problem with our update rule is that low quality data can obscure better data when obstacles are viewed at longer range. Initially, we applied a "horizon" or maximum range threshold on our sensor data to limit the use of low resolution data. While this results in cleaner maps, it is a wasteful loss of potentially useful data and can seriously reduce the efficiency of exploration algorithms.

To overcome this problem we applied a hypothesis-explanation heuristic. Effectively, the occupancy grid is a model of our robot's world. Each time an obstacle is sensed, we hypothesize that there is an obstacle within a specified uncertainty region. At this point, before applying our update, we inspect the uncertainty region for better quality data. If we find more precise evidence of an obstacle within the region, this "explains" our hypothesis. Consequently, the "model" (or map) and the sensor data agree, and there is no need to apply this lower quality update.

To implement this heuristic, a second grid value is maintained. Each grid updated inside an uncertainty region also stores the value of the disparity from which its reading arose. By inspecting update regions for higher disparity readings before applying the update, we avoid the smearing or blurring affect of lower quality data on the map. This solution is simple, robust, sensitive to dynamic objects and applies no artificial horizon on the sensor data.

5. Navigation and Exploration

We have implemented path planning and autonomous exploration on top of the vision-based occupancy grid mapping. The algorithms were initially reported in Murray and Jennings (1997) and Little et al. (1998a).

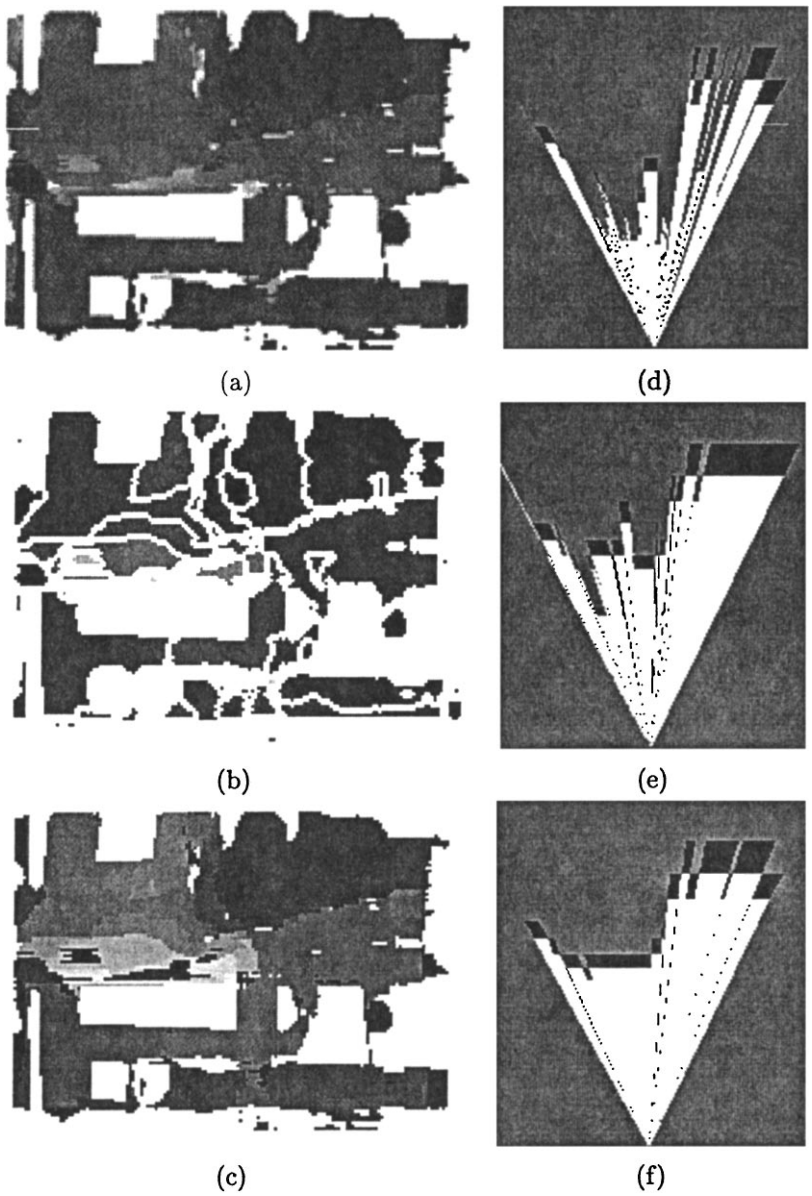

Figure 7. The results of a single stereo disparity image of the scene shown in Fig. 6 with various filters applied and the matching top-down map: (a) and (d) show the raw disparity image (with invalid regions white) and the resultant top-down view, (b) and (e) show the same with the disparity image filtered using morphological erosion on regions of constant disparity, and (c) and (f) show the results using surface segmentation for noise rejection (the surfaces that have been rejected are shown in black in image (c)).

5.1. Path Planning

Given a goal location and the robot position, we want to find the "best" path connecting the two points. The "best" path in this case means a path that is short, but also safe. To be safe, the robot must avoid obstacles that appear in the map.

Our path planning approach is a mixture of shortest path (Lengyel et al., 1990) and potential field (Khatib, 1986; Barraquand et al., 1992) path planning methods. In the shortest path method, the goal location is marked in a grid with a distance value of 0. All other locations are labeled with very high distance values. The algorithm begins at the destination and executes

a breadth-first search through all connected grid locations. Each time a grid location i is visited from grid location j, it is updated by:

$$\begin{array}{lll} \texttt{if} & map(i) & \texttt{blocked} \\ & d(i) & = \infty \\ \texttt{else} & d(i) & = min \begin{cases} d(i) \\ d(j) + c(i, j) \end{cases} \end{array}$$

where $c(i, j)$ is the cost (in this case the distance) associated with moving from site i to site j. A site is `blocked` if it is within a fixed distance to an obstacle grid location. If site i obtains a lower distance value after the visit, this site is marked as included in the next round of the breadth-first search. This breadth-first search expands around the destination in a wave front, parting at and propagating around obstacles. After it has completed, the shortest path from robot to goal is found by connecting adjacent sites with the minimum distance values from the robot position to the goal. This is effectively a distance-to-goal gradient descent method.

The minimum safe distance to an obstacle (used to determine if a grid location is blocked) is a tuning parameter. The behaviour induced by this safe distance parameter is the main drawback to the shortest path approach. In most indoor mobile robot applications, it is difficult to set a single value for this parameter that will satisfy all situations in which the robot may find itself. If a gap between two obstacles is smaller than two times this minimum distance, the robot cannot pass through that gap. This commonly occurs at doorways. However, when circling obstacles in relatively open spaces, the robot will still pass the obstacle at the same minimum distance. This creates an unpleasant trade-off. Large distance cushions increase safety but decrease the flexibility of the system. Small distance cushions create paths that often cut very close to obstacles.

The potential field method addresses the problem of a fixed safe distance constraint. It treats the goal and obstacles respectively with fields of attractive and repulsive forces. Like electric fields, the strength of these fields are inversely proportional to the distance from the source. This creates a potential field landscape over the entire mapped environment. At any point in the map, a path can be determined by applying gradient descent to this landscape. This approach has the property that the safe distance cushion around obstacles adapts to the surrounding environment. In narrow gaps, a path may still be found, but it will always choose the centre of the gap as the best location through which to pass. In open spaces it will give obstacles a wide berth. It has the disadvantage, however, that obstacles a comfortable distance from the shortest distance path will still result in path deflections. It can also be difficult to perform without imposing a boundary around the robot space.

Ideally we would like to have a path planner that would use the shortest distance method for open space traverses, and the potential field method in confined areas. We have attempted to create this combination of shortest distance and potential field approaches. We use a breadth-first search similar to the shortest distance approach. However, we adjust the cost of going between two sites by an additional factor that is a function of the distance to the nearest obstacle. Thus, our site distance update method is:

$$\begin{array}{lll} \texttt{if} & map(i) & \texttt{occupied} \\ & d(i) & = \infty \\ \texttt{else} & d(i) & = min \begin{cases} d(i) \\ d(j) + \alpha(o(i))c(i, j) \end{cases} \end{array}$$

Here $o(i)$ is the distance to the nearest obstacle for site i. $\alpha(d)$ is a weighting function that increases the cost of sites close to obstacles. In our implementation we set α to be a linear function that starts at some high value and approaches 1 as d approaches some safe distance away from the obstacles. With this approach, in clear areas, the path planner behaves as a shortest distance planner, and in cluttered areas it behaves as a potential field planner. Figure 8 shows an example generated plan.

5.2. Exploration

We achieved autonomous exploration via modification of the path planning algorithm. We begin at the robot position instead of beginning the breadth-first seach at the goal, since there is no goal provided. The objective of our search is to identify the closest "unknown" sites (i.e., sites with 50% probability of occupancy) and determine the path to the closest (or least expensive) one. This is similar to applying an attractive field to all unknown regions. The method begins at the robot position and proceeds with the same breadth-first search as we described in the previous section. When the search reaches a grid location in the map marked as

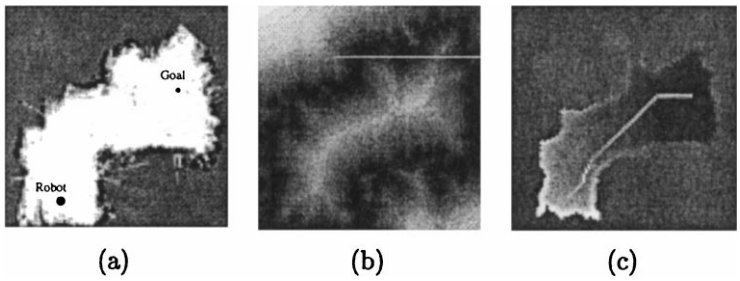

Figure 8. Path plan generation: (a) the map showing the robot position and destination (b) the distance-to-nearest obstacle map (c) the cost map showing the distance to the destination, black indicates near the goal, white indicates far from the goal, as well as the planned path in white.

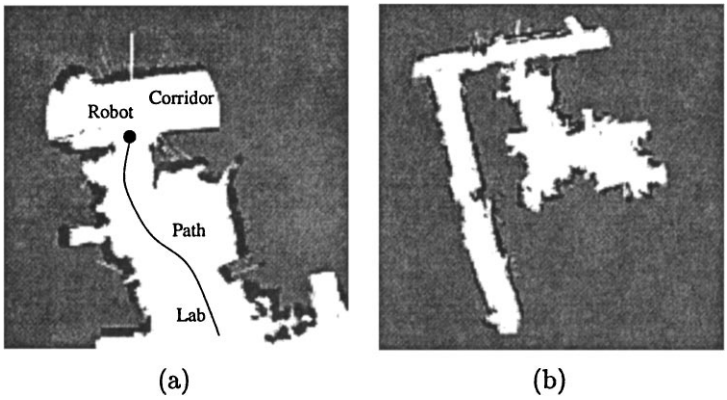

Figure 9. Two example maps generated through autonomous exploration: (a) close-up of the map during exploration (b) the entire map after the robot has left the lab and is exploring down the hallway.

"unknown" it inspects the current distance to the robot for that location. If this location has the shortes distance to the robot among all previously visited unknown sites, its value and grid coordinates are saved as the "best" exploration destination so far. Once all active grid locations in the search are marked with distance values greater that the most attractive exploration destination, the search is complete. To determine the path, begin at the saved grid coordinates of the closest unknown grid location and follow the chain of locations with the minimum distance markings until the robot position is reached.

The robot has successfully explored building floors through several rooms and corridors without human intervention. In Fig. 9(a) and (b) we show two example maps. Figure 9(a) shows a close-up of the map during exploration, as the robot is leaving the lab and entering an adjoining hallway. Figure 9(b) shows the entire map after the robot has successfully left the lab and followed the hallway around a corner.

6. Conclusion

We have shown the stereo vision is a viable alternative to other spatial sensors for constructing occupancy grid maps. Although stereo vision has several characteristics that can make mapping prone to errors, it also has great capability in localizing obstacles quickly and accurately. We showed a few techniques to improve the quality of stereo vision mapping results on a working system. Several example results were given.

6.1. Remaining Challenges

One serious problem that remains unsolved in our current mapping and navigation implementation is the problem of seeing over, under or past objects. Certain objects are very difficult to see for the stereo vision, due to the large occlusions involved. The most dangerous are table tops that appear between the height of the top and reference cameras. These obstacles are dangerous

because

— they are difficult to successfully match in the left/right camera pair as the the features are aligned with the baseline
— they are difficult to match in the top/bottom camera pair as the views of the obstacle is very different between the two images due to occlusions
— they are positioned to cause the maximum damage to the robot if it should run into them

Unfortunately, even though they are difficult to see for the robot, they do not obscure surfaces behind them. Therefore the robot often can accurately resolve obstacles beyond the table, and since it sees "through" the table, does not add it to the map. This problem has been partially solved by the addition of surface segmentation to the the disparity image filtering. However, this only works when the table top, or portions of it, are successfully resolved by the stereo algorithm. We are looking into incorporating concepts from Konolige's MURIEL system (Konolige, 1997) to address this problem.

Another imposing challenge that remains is the localization problem. Regardless of the quality of the maps, their utility is limited because the robot cannot reuse them from power-up to power-up, and also as odometry drift accumulates, old map data becomes inaccurate. We are currently investigating automatic acquisition and use of 3D landmarks using a variety of features such as corners, vertical lines and intersection of 3D lines. Some preliminary work has been reported in Little et al. (1998b).

We have done some experiments in determining disparity images to sub-pixel resolution with good success. The challenge now is to tune the algorithm to run in near real–time with the onboard processing capability of a PC-based robot. By taking advantage of the MMX instruction set, and using Intel Pentium II processors, we hope to achieve sub-pixel resolution stereo with 320×240 pixel images at speeds of over 3 Hz. This would dramatically increase the sensing resolution and provide many new opportunities (and new difficulties to be overcome) for stereo vision mapping.

Finally, there is the challenge of extending our 2D occupancy grids to 3D voxel-based representations of the world. With a texture-mapped voxel representation of the world we intend to investigate automatic generation of virtual reality models based on real-world scenes.

Acknowledgments

We would like to thank all the members of the LCI lab at UBC for their fruitful insights and commentary. A special thanks goes to David Lowe and Jochen Lang for their perseverance in the face of obstinacy.

This work was supported by grants from the Natural Sciences and Engineering and Research Council of Canada and from the Institute for Robotics and Intelligent Systems, a Canadian Network of Centres of Excellence.

Notes

1. www.rwii.com
2. For more information, see www.ptgrey.com

References

Barraquand, J., Langlois, B., and Latombe, J. 1992. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, Cybernetics*, 22(2):224–241.

Brooks, R.A. 1987. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2: 14–23.

Brooks, R.A., Connell, J., and Ning, P. 1988. Herbert: A second generation mobile robot. A.I. Memo No. 1016, MIT AI Laboratory.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. 1997a. On building and navigating with a local metric environmental map. In *ICAR-97*.

Dudek, G., Milios, E., Jenkin, M., and Wilkes, D. 1997b. Map validation and self-location for a robot with a graph-like map. *Robotics and Autonomous Systems*, 22(2):159–178.

Elfes, A. 1989. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57.

Fua, P. 1993. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1):35–49.

Kanade, T., Yoshida, A., Oda, K., Kano, H., and Tanaka, M. 1996. A video-rate stereo machine and its new applications. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, San Francisco, California, USA.

Khatib, O. 1986. Real–time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98.

Konolige, K. 1997. Improved occupancy grids for map. *Autonomous Robots*, (4):351–367.

Lengyel, J., Reichert, M., Donald, B., and Greenberg, D. 1990. Real–time robot motion planning using rasterizing computer graphics hardware. In *Proc. of SIGGRAPH*, Dallas, Texas, pp. 327–335.

Little, J.J., Jennings, C., and Murray, D.R. 1998a. Vision-based mapping with cooperative robots. In *SPIE-98, Sensor Fusion and Decentralized Control in Robotic Systems*, pp. 2–12.

Little, J.J., Murray, D., and Lu, J. 1998b. Identifying stereo corners for mobile robot localization. In *IROS-98*, pp. 1072–1077.

Matthies, L. and Grandjean, P. 1987. Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors. *RA*, 10(6).

Matthies, L. and Shafer, S.A. 1987. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, 3:239–248.

Moravec, H. and Elfes, A. 1985. High-resolution maps from wide-angle sonar. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, St. Louis, Missouri.

Murray, D. and Jennings, C. 1997. Stereo vision based mapping for a mobile robot. In *Proc. IEEE Conf. on Robotics and Automation, 1997*.

Nickerson, S.B. et al. 1993. Ark: Autonomous navigation of a mobile robot in a known environment. In *IAS-3*, pp. 288–293.

Okutomi, M. and Kanade, T. 1993. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363.

Tsai, R.Y. 1987. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3:323–344.

Tucakov, V., Sahota, M., Murray, D., Mackworth, A., Little, J., Kingdon, S., Jennings, C., and Barman, R. 1997. Spinoza: A stereoscopic visually guided mobile robot. In *Proceedings of the Thirteenth Annual Hawaii International Conference of System Sciences*, pp. 188–197.