Stereo vision based mapping and navigation for mobile robots

Cullen Jennings Don Murray
Department of Computer Science,
University of British Columbia, Vancouver
{*jennings, donm*}*@cs.ubc.ca*

Abstract

This paper describes a visually guided robot that can plan paths, construct maps and explore an indoor environment. The robot uses a trinocular stereo vision system to produce highly accurate depth images at 2 Hz allowing it to safely travel through the environment at 0.5 m/s. The algorithm integrates stereo vision, occupancy grid mapping, and potential field path planning techniques to form a robust and cohesive robotic system for mapping and navigation. Stereo vision is shown to be a viable alternative to active sensing devices such as sonar and laser range finders.

1 Introduction

This paper addresses the problem of robot navigation in an unknown and dynamic environment. To be mobile, it must be able to safely navigate within its environment and reliably get from A to B. We envisage a robot that can be placed in a dynamic and unknown environment and can, unaided, discover and maintain sufficient information about its surroundings to enable it to accomplish these tasks.

For such a robot, the first requirement is adequate spatial sensing. The second requirement is that it be able to retain and integrate sensor readings over a period of time to create a reliable map for navigation. Since the environment is dynamic, the robot must be able to adapt to changes within its work area. Finally, to obtain a complete map without supervision, exploration capabilities are desirable.

There has been considerable research in the field of mobile robot navigation [2][9][5][11]. However, relatively few complete and working systems have been reported[3][12]. Most systems to date have used laser range finder and sonar array sensing devices. Very few mobile robot systems have used real–time stereo vision for acquiring 3-D sensory data. This can be attributed to the difficulty in calibrating stereo vision systems, their expense, and the high computation cost for computing accurate and dense stereo data in real–time.

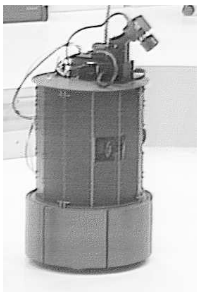

Figure 1: Spinoza

The system presented in this paper combines several approaches to sensing, mapping, path planning and exploration. The experiment was conducted with the Laboratory for Computational Intelligence mobile robot Spinoza at the University of British Columbia [14]. Spinoza (shown in Figure 1) uses a trinocular stereo system that greatly improves sensing accuracy and reliability relative to binocular systems. The vision system combines highly calibrated wide angle cameras and high speed dedicated digital signal processors (DSPs) to provide fast, dense depth sensing.

Obstacles detected through the vision system are mapped into an "occupancy grid", a raster map of the robot's environment. This map is updated continuously as stereo data is acquired, and is used for planning paths and exploration.

Path planning is achieved by minimizing a cost function that is a weighted representation of the proximity to obstacles and the distance to an attractive goal. The cost function combines shortest path search methods [7] with potential field approaches [6] [1] to achieve direct paths that pass obstacles at sufficiently safe distances.

The same path planning algorithm is adapted for exploration by assigning goal attributes to unknown regions in the map. The robot moves towards the nearest unexplored region, updating the map as it goes, until all accessible areas have been explored.

The system is robust and performs at speeds significantly faster than those previously reported.

2 System Architecture

The Spinoza robot consists of an Real World Interface (RWI) B-12 base with 3 black and white video cameras for trinocular stereo. There are two TMS320C40 DSPs for image processing, two T805 transputers for control and a T225 transputer for communication. The robot can communicate to a Sun Ultra Sparc 1 workstation over high speed radio modems. One of the DSPs has two SGS-Thompson A110 convolver chips added to accelerate stereo processing. The DSPs were chosen because of their image processing power while the transputers were chosen for their ease of use in real time applications, communications, and distributed processing.

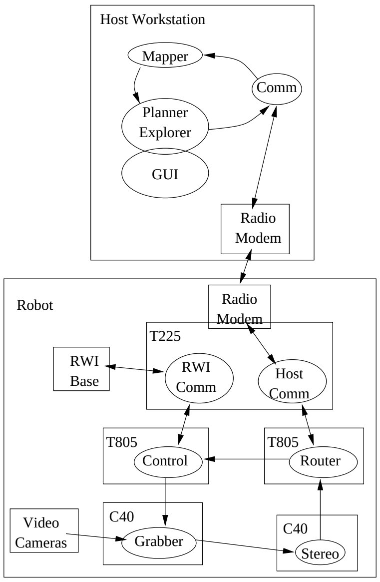

Figure 2: Architecture

Figure 2 shows the data flow through the subsystems and across the hardware can be seen in Figure 2. The cameras are genlocked so that the first C40 can capture a field of video simultaneously from all three. This is transfered to the second C40 which computes a trinocular depth image. It then computes a horizontal planar map that contains the nearest obstacle in each direction the cameras can see. The planar depth map and the current location and heading of the robot (as computed from odometry) is sent via the router and radio modems to the host. On the host, the mapper program uses the planar map to update its occupancy grid representation of the environment. The environment map is passed on to the planner/explorer module which computes safe and desired paths for the robot. These paths are sent back down to the robot where the controller communicates with the RWI base to execute the path.

3 Stereo

Spinoza uses a trinocular stereo system for sensing. Stereo provides dense depth maps and has high angular resolution. This allows it to resolve small or narrow objects significantly better than sonar arrays. Laser range scanners obtain a high degree of accuracy but require a long time to scan an area. Stereo vision is also a passive sensing technique, which is preferable to active sensing for some applications. One of the major limitations to the use of stereo has been its high computational cost. Spinoza overcomes these limitations by employing high–speed convolvers to ensure fast computation and precision calibration to achieve depth images with little noise.

The system gives a high quality depth map that is 128x120 pixels with 20 disparities and computes in 350 ms. This very high performance combined with wide angle lenses allows the robot to see a large area very quickly so that it can map without pausing and move at an acceptable velocity.

The stereo system computes depth maps using three calibrated cameras with an algorithm similar to the multi baseline stereo [10]. On Spinoza the first pair of cameras are in a horizontal plane while the second are in the vertical plane. The trinocular cameras can achieve better results than a typical two camera stereo system because the second pair of cameras can resolve situations that are ambiguous to the first pair. For example when the horizontal camera pair looks at a scene that has only horizontal lines, the depth of the images is ambiguous, but the vertical cameras can find the depth for this scene.

3.1 Camera Calibration

The internal parameters of each camera are computed using an extension to the Tsai method[8] that employs a nonlinear gradient descent technique to compute the centre of radial distortion. This provides a model of the radial distortion and focal length of the lenses. The radial distortion is quite significant because the lenses' 80 degree field of view is quite wide. A warp table is generated that maps the original images into subsampled versions that have been corrected for lens distortion and have epipolar lines that are aligned with the x and y axis of the image. An example input image is shown in Figure 3, and the undistorted image is shown in Figure 4.

The position of each camera is computed relative to a calibration target. This information is used to compute the transformations (rotation and shear) needed to align the upper and left cameras with the right camera. An accurate calibration is important in obtaining stereo matches with little noise and in projecting the stereo data into 3–D coordinates.

Figure 3: Right Image

Figure 4: Corrected Image

The robot is calibrated several times at different angles of rotation, to determine the centre of rotation for the robot. This allows the stereo depth maps to be projected into volumes in 3–D space in a coordinate system relative to the center of the robot.

3.2 Stereo computation

The stereo algorithm starts by computing the Laplacian of the Gaussian for each of the three images. The algorithm then loops over the disparities and computes the sum of the absolute values of the differences (SAD) between the image pairs using a 5 by 5 window. It keeps track of the disparity with the minimum SAD for each pixel. The correlation and filtering is done using two Thompson A110 convolver chips that run in parallel with the C40.

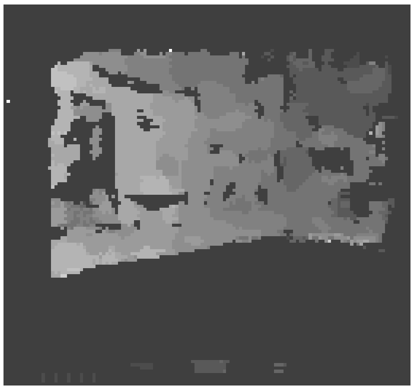

Figure 5: Depth Image

The depth image computed from the scene in Figure 3 is shown in Figure 5. Points that are closer in the image are displayed as a lighter color of gray and invalid points are displayed as completely black. This stereo data is dense and has almost no noise.

3.3 Validation

The SAD value for each pixel is normalized by the SAD value that would be found if the right image was matched against a blank image. If the normalized value is below a threshold, this location is marked as invalid in the depth map. Further pixel noise is removed from the depth map by marking invalid all pixels that don't have the same depth value as the pixel above and below them.

Shiny linoleum floor tends to reflect objects and the stereo detects these reflections as being under the floor. Each pixel below the horizon is looked at and pixel that are on or below the floor are marked invalid to ignore floor texture and floor reflections. The location of the floor relative to the cameras is known from the calibration stage.

Once all the invalid pixels in the image have been found, a planar map is formed by taking the closest disparity in each column of the depth map. This forms a 1 by 128 planar map that represents the nearest obstacle in each of the directions that the cameras can see. The planar map generated from the depth map in Figure 5 is shown in Figure 6.

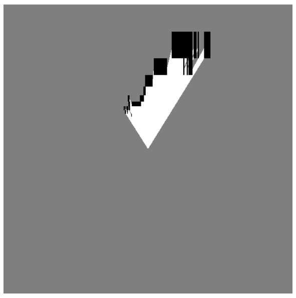

Figure 6: Planar Map

4 Map building

The previous work in robotic map building has revolved around two themes: occupancy or certainty grids, and feature-based methods. Feature-based methods such as demonstrated by Rencken [11] works by locating features in the environment, localizing them, and then using them as known landmarks by which to localize the robot as it searches for the next landmarks. Occupancy grid mapping, as pioneered by Moravec and Elfes [9] [5] is a technique that 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. We selected the occupancy grid approach due to its simplicity, robustness and adaptability to dynamic environments.

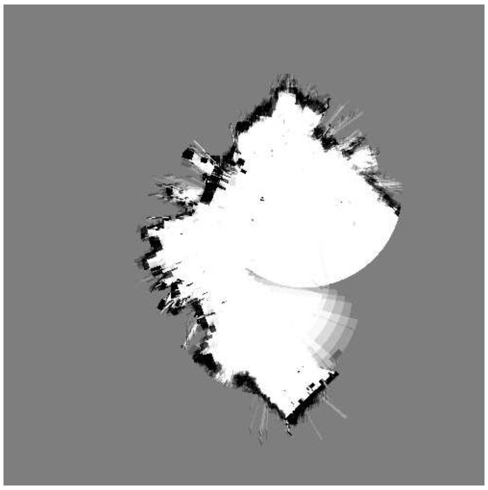

Figure 7: Occupancy Grid Map

In the occupancy grid method, the robot's environment is tessellated into a discrete grid. Each grid location is assigned a value that represents the probability that it is occupied by an obstacle. Initially, all grid values are set to a 50% probability. This represents the "unknown" case. The grid locations that fall within the region of uncertainty about each sensed obstacle point have their values increased, while locations between the robot and the obstacle have their probabilities decreased.

Several strategies exist for updating grid location values. We selected the simplest, which is to increment or decrement location values with each reading. Each grid location value could vary from $0 \rightarrow 255$. The increment/decrement step-size is a tunable parameter. A high value allows the map to adapt quickly to new data, but makes it less reliable in the presence of noise. We chose a value of 30. This is rather high, but our stereo sensing provides reliable, consistent data to the degree that this was acceptable. A sample occupancy grid map is shown in Figure 7. In this figure, black represents 100% certain obstacles while white represents clear space.

5 Navigation

We implemented two high-order navigation tasks: path planning and navigation. In the first, given a map and a goal, we construct a safe but direct path for the robot to follow. In the second, the robot is allowed to choose its paths so as to increase its knowledge of its environment.

5.1 Path planning

In the past, two methods have emerged as dominant for path planning in discrete grid maps. The first is the *distance transform* or *shortest path* method as described in Lengyel *et al.* [7]. In this method, the goal destination is labeled with a distance value of 0. All other locations are labeled with very high values. The algorithm begins at the destination and each iteration visits all locations adjacent to locations visited in the previous iteration. The distance value for site i adjacent to previously visited site j is updated by:

$$
\begin{array}{lll}
\texttt{if} & map(i) & \texttt{occupied} \\
 & d(i) & = \infty \\
\texttt{else} & & \\
 & d(i) & = min \left\{ \begin{array}{l} d(i) \\ d(j) + c(i,j) \end{array} \right.
\end{array}
$$

where $c(i,j)$ is the cost or distance associated with moving from site i to site j.

The distance transform expands around the destination in a wave front, parting at and propagating around obstacles. The shortest path from any location to the destination can then be found by following the connected sites with minimum d until $d = 0$ is reached.

One of the problems with this method is that it inherently comes as close to obstacles as is allowed.

Another problem is that it is often applied with only 4-connected graphs. This yields a shortest path with a Manhattan distance measure that is often not desirable. Eight-connected graphs (as shown in Figure 8) yield a more direct path that is better in most applications.

	1	
1	0	1
	1	

$\sqrt{2}$	1	$\sqrt{2}$
1	0	1
$\sqrt{2}$	1	$\sqrt{2}$

Figure 8: Manhattan (left) and chamfer (right) distance connectedness

Another popular approach is the use of potential fields [6] [1]. In this case, each obstacle applies a repelling field to the robot, while the goal applies an attractive field. By applying gradient descent to the resultant potential field landscape, one can obtain a safe path, keeping obstacles at a maximum distance during the approach to the goal. One of the drawbacks, however, is the necessity to constrain the problem with a fixed boundary. Without this, it can lead to paths that stray quite far from the goal with the intent of staying away from obstacles.

In our approach, we combine the potential fields with the shortest path method. We first construct a labeling, $o(i)$, for all grid locations that contains the chamfer distance to the nearest obstacle. We then apply the wave front search starting from the goal. However, the distance penalty for each visited site is modified by a cost multiplier α based on the distance to the nearest obstacle. Thus, for a site i visited from j,

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

The weighting function $\alpha(d)$ begins at a high penalty at $d = 1$ and ramps linearly down to 1 at d equal to a safe distance related to the size of the robot and the speed at which it moved.

Figure 9 shows the distance labeling for the map displayed in Figure 12. One can see that the labeling clearly indicates the skeleton of the region surrounded by the obstacles as the safest. Figure 10 shows the result of applying our approach; the total cost function is mapped, as well as the chosen path.

5.2 Exploration

Exploration methods have been implemented using neural networks and landmarks [13][4] as well as other

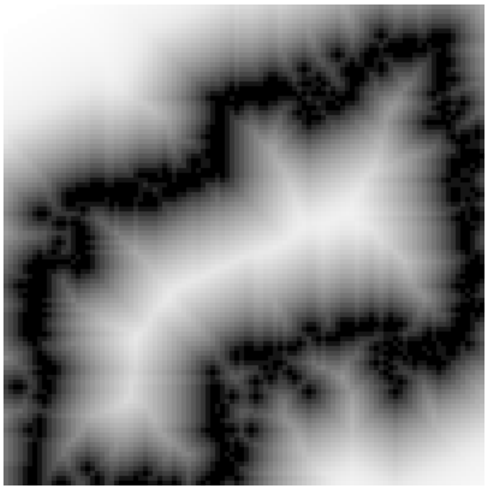

Figure 9: Distance from obstacles labeling

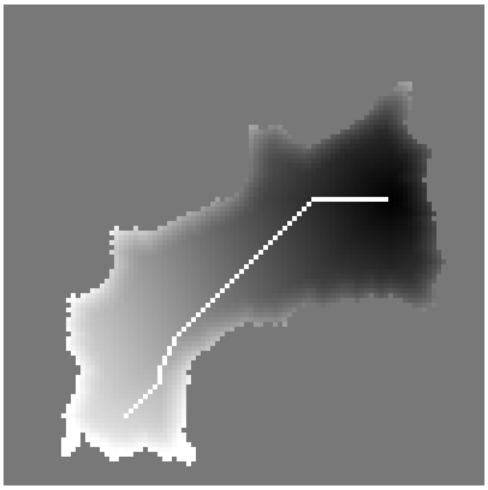

Figure 10: Final cost function with path indicated

techniques. In our approach, grid locations are classified into 3 basic types: blocked, clear and unknown. We would like to reduce all unknown regions until all reachable areas are either clear or blocked. We achieve this by assigning an attractive potential field to all unknown areas, while maintaining the obstacle repulsion fields.

This was implemented by re-using our path planning algorithm, with the unknown grid locations all being assigned as destinations with $d = 0$. The breadth-first search is begun simultaneously at all these locations. By following the path with the minimum cost, the robot is guided to the nearest accessible unknown region. With periodic reevaluation and re-planning, the robot will explore from unknown region to unknown region until no more unknown regions are reachable.

Figure 11 shows an example of exploration in progress. The dark line indicates the path that Spinoza has taken. Figure 12 shows the exploration completed, with no unknown destinations remaining.

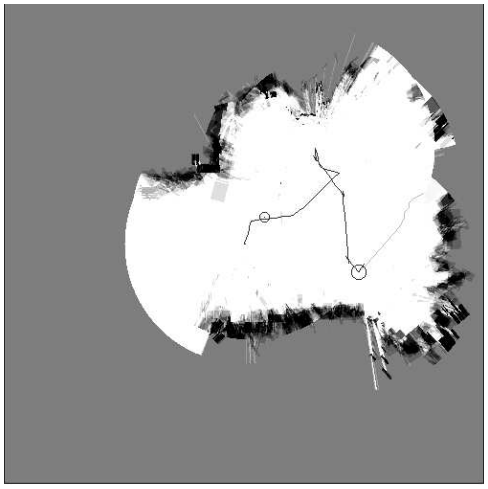

Figure 11: Exploration in progress

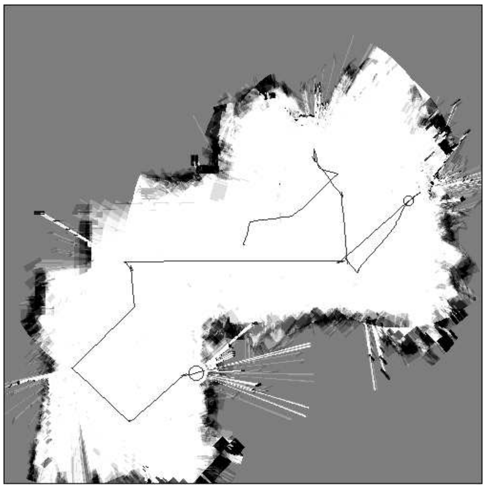

Figure 12: End of exploration

6 Conclusion

The paper demonstrated that a visually guided mobile robot can safely map, explore and navigate unknown indoor environments. We have shown that real–time stereo vision is a viable alternative to active sensing devices for these applications. Robot velocities were limited by the speed at which stereo could be computed. With our current system calculating stereo depth images at 2 Hz, the robot traveled at a speed of 0.5 m/s without degrading safe navigation or map updating.

Occupancy grid mapping provided a good base for raster-based path planning techniques. Planning techniques that combine shortest path searches with repulsion fields from nearby obstacles can strike a balance between the most direct and the most safe path. The exploration module mapped the entire robot environment via an efficient path.

While the Real World Interface robot base odometry proved to be adequate for our task, future work will include localization of robot position to complement odometry readings. Spinoza has an additional color camera on a pan tilt unit that we plan to use for tracking and identifying landmarks in a localization module. In addition, linking of local maps into a connected graph of nodes to represent larger spaces should improve the overall range and capabilities of the system.

7 Acknowledgments

We would like to sincerely thank Rod Barman and Stewart Kingdon for their hard work and effort in making Spinoza work. We would also like to thank Mike Sahota, Vlad Tucakov, Jim Little, and Alan Mackworth for their many ideas and continuous enthusiasm and support for the Spinoza project.

References

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

[2] R. Brooks and T. Lozano-Perez. A subdivision algorithm in configuration space for findpath with rotation. In *Proc. of International Joint Conference on Artificial Intelligence*, pages 799–806, Karlsruhe, West Germany, August 1983.

[3] R. Brown, L. Chew, and B. Donald. Mobile robots, map–making, shape metrics, and localization. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, Atlanta, GA, May 1993.

[4] T. Edlinger and E. Puttkamer. Exploration of an indoor environment by an autonomous mobile robot. In *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems*, pages 1278–1284, Munich, Germany, September 1994.

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

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

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

[8] R. Lenz and R. Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3–d machine vision metrology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–10(5):713–720, September 1988.

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

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

[11] W. Rencken. Autonomous sonar navigation in indoor, unknown and unstructured environments. In *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems*, pages 431–438, Munich, Germany, September 1994.

[12] E. Stuck, A. Manz, D. Green, and S. Elgazar. Map updating and path planning for real–time mobile robot navigation. In *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems*, pages 753–760, Munich, Germany, September 1994.

[13] C. Taylor and D. Kriegman. Exploration strategies for mobile robots. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 248–253, Atlanta, GA, May 1993.

[14] V. Tucakov, M. Sahota, D. Murray, A. Mackworth, J. Little, S. Kingdon, C. Jennings, and R. Barman. A stereoscopic visually guided mobile robot. In *Proc. of Hawaii International Conference on Systems Sciences*, January 1997.