A. Briggs, D. Scharstein, and S. Abbott. Reliable mobile robot navigation from unreliable visual cues. In *Proceedings of the Fourth International Workshop on Algorithmic Foundations of Robotics (WAFR 2000),* Hanover, NH, March 2000, pages SA97-110.

* Funded by the National Science Foundation

LOCALIZATION AND IDENTIFICATION OF VISUAL LANDMARKS

Dan Knights, Jeff Lanza
Computer Science Department
Middlebury College
Middlebury, VT, 05753
(802) 443 - 4072, (802) 443 - 4221
knights@middlebury.edu, lanza@middlebury.edu
Advisor: Daniel Scharstein, Middlebury College

This project focused on designing and evaluating methods for reading barcodes on visual landmarks. Such landmarks have many applications, including visual tracking, servo-ing, and mobile robot navigation. Our goal was to improve a preliminary version of the barcode reader developed the previous summer in the Middlebury Robotics and Vision Lab.

The only information the robot has about a visual landmark is a grayscale image taken by its onboard camera, which means there exist many variables that complicate the reading of the barcode. For example, the landmark can be very close or very far; landmarks may be rotated or skewed, adding foreshortening effects from perspective. We found that almost every barcode could be read correctly given localization of the correct corner points of the landmark, so our first goal was to design a better algorithm for finding these points. Statistical analysis produced search ranges for the corners, allowing fast reliable detection of the correct corner points.

Given this improved method, reading the barcode became significantly more reliable. In our real-world application, however, interference in the radio transmission of images caused barcodes to be misread, and at times caused our program to "hallucinate" landmarks. In order to deal with these cases, we used more powerful error-detecting codes, which allowed most misread codes to be ignored.

In order to test the reader, we created a database of 393 images with varying landmark orientations, distances from the camera, and lighting conditions. Using these images, we determined that the new barcode reader found the correct position of the landmark 89% of the

time, which is significantly higher than the 75% we found when running our preliminary version of the barcode reader on the same images. Furthermore, the new reader successfully reduced the percentage of times a wrong barcode was reported from 4.33% to 0.76%.

Currently we are extending our algorithms to barcodes that convey varying amounts of information depending on the distance to the camera, and that are tolerant to partial occlusion of the barcode.

* Funded by the National Science Foundation

+ Funded by a fellowship from the Council on Undergraduate Research

THE MIDDGO PROJECT

Bryan McQuade
Department of Mathematics and Computer Science
Middlebury College
Middlebury, VT 05753
bmcquade@middlebury.edu
Advisor: Tim Huang, Middlebury College

The ancient game of go, still the most popular strategy game in the Far East, has proven very challenging for artificial intelligence (AI) researchers trying to develop go-playing programs. In contrast to the best chess, checkers, backgammon, Othello, and Scrabble programs, all of which play at or better than the level of the top human players, the best go programs play only at the level of advanced beginners. The MiddGo project, begun in the Fall of 1999, aims to better understand the challenges posed by go and to investigate approaches for improving the performance of go-playing systems.

Most strategy game playing programs use brute-force minimax tree search with a heuristic evaluation function. In the game of go, however, full tree search is especially expensive. The branching factor is an order of magnitude greater than chess, and some positions require reading tens of moves ahead. Furthermore, evaluating positions is more difficult in go. In a game of chess, the sum of the values of each players' pieces is an effective evaluation function. Such a function evaluates quickly and is easy to implement. In go, however, there is only one type of piece, and each player usually has roughly the same number of pieces on the board. These factors limit the effectiveness of minimax tree search in go and require new methods for move evaluation.

With selective search algorithms, which limit the branching factor by eliminating moves that are unlikely to improve board position, tree search can be more effective. MiddGo has implemented ladder analysis, which quickly identifies a specific pattern of moves that yield the