Mobile Robot Path Tracking Using Visual Servoing

Andrew Thomson and Jacky Baltes

Centre for Imaging Technology and Robotics
University of Auckland
j.baltes@auckland.ac.nz

Abstract

This paper describes a path tracking controller for mobile robots using visual servoing. A highly efficient algorithm suitable for cheap and low power micro-processor is described. The algorithm uses a highly focused search in the image to approximate the offset and gradient of the path. These features are determined solely by a sweep through two rows of the image. An empirical evaluation shows that the algorithm is efficient and robustness. Furthermore, the empirical evaluation investigates the relationship between the average error and the look ahead distance as well as the weighting between the offset and gradient information.

Keywords: Visual Servoing, Path Tracking, Mobile Robots

1 Introduction

The goal of this research is to develop a simple and robust control method which allows the robot to follow a marked path on the floor. This type of robot may be used in a warehouse or plant for moving components from one work station to another. Instead of the standard conveyor belts, a mobile robot would be more flexible and could be easier adapted to changes in the environment.

This paper addresses the problem of visual servoing for path following. Our approach is different from other work in this area because of it requires few computational resources and makes few assumptions about the robot kinematics or the shape of the path.

The problem of vision guided navigation has been investigated by many researchers. So far, most approaches have been applied to autonomous vehicles. These approaches use sophisticated methods for feature extraction (e.g., Kalman filters), and require accurate mathematical models of the underlying kinematics and dynamics of the robot.

In recent years, advances in processing power have made it possible to try and solve these problems using micro-controllers. Y. Ma et al propose a method based on visual servoing and image translations [Ma et al. 1999]. They only tested their approach in simulation.

Hashimoto and Noritsugo present a similar approach. The shape of the curve is estimated from the image and suitable control inputs are set [Hashimoto and Noritsugo 1997]. The shape estimation of the path itself is non-trivial and computationally expensive.

Masutani et al and Kobayashi et al present visual servoing approaches that are augmented by machine learning. [Masutani et al. 1994; Kobayashi et al. 1996].

Section 2 describes the robotic platform that was used in this research. The image processing and the control algorithms are described in section 3. An empirical evaluation showing the efficiency and robustness of the system are shown in section 4. Section 5 concludes the paper.

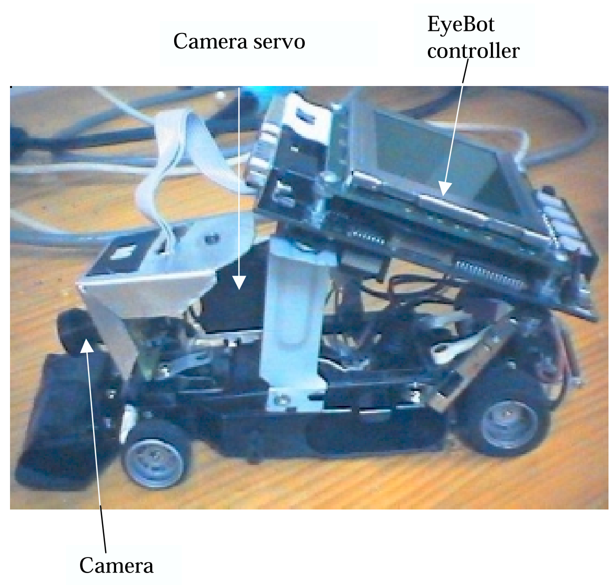

Figure 1: Local Vision Autonomous Robot Platform

2 Local Vision Robotic Platform

This section describes the robotic platforms used in the experiments. All robots used in the lab were designed with two specific objectives: (a) the robots use cheap, commonly available components, and (b) the robots are versatile and able to fulfill a variety of different roles.

The robot used in this experiment is a member of the "4 Stooges," a team of robots that compete in the RoboCup competition [(ed.) et al. 2001; Baltes 2001]. The chassis was taken from a Tamiya remote controlled toy car. The original driving motor and steering servo are used.

The receiver of the RC car was replaced with an Eyebot controller, designed by Thomas Braunl [Bräunl 2002]. The controller is based on a Motorola MC68332 micro-controller and includes 1 MB RAM and 512 KB ROM. The Eyebot controller has IO ports for a CMOS camera, DC motors, RC servos, infrared transceivers, serial ports, and parallel ports.

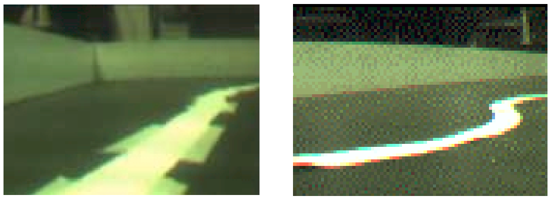

Figure 2: Views from the camera of the robot

3 Image Processing and Control

The path tracking controller is a visual servoing system. It does not make any assumptions about the shape of the path to follow and does not have information about the kinematics of the robot.

The approach is solely vision based. The controller is trying to send control the robot in such a way that the image of the path is centered in the view of the robot.

The path is indicated by a white line on the playing field. Figure 2 shows two views of the robots camera when tracking a path.

There are a variety of different methods for calculating features for path tracking from the image. However, the required computations are expensive. The goal of the image processing described in this image is to approximate the offset and gradient of the path with minimal resources.

To reduce the computational cost of the algorithm, the image processing for the path tracking control is limited to the region in front of the robot. The height of the region is set such that the point is at most 17 cm in front of the robot. Furthermore, only the boundary of the region is processed. This means since the images from the camera have a width of 80 pixels, at most 200 pixels need to be checked.

A simple color detection and segmentation routine is used to extract the path from the image. Then the offset O_p and gradient G_p of the path are approximated using two feature points in the image (see Fig. 3).

Point P_1 is the middle point of the path on the bottom or left edge of the image. Point P_2 is the middle point of the top or right edge of the image. The offset and gradient of the path are approximated using equation 1 and 2 respectively.

The width of the image is denoted by w. $P_i.x$ is the x-coordinate of point i, and $P_i.y$ is the y coordinate of point i respectively.

$$\text{Offset} = \frac{(P_1.x - w/2) + (P_2.x - w/2)}{w/2} \quad (1)$$

$$\text{Gradient} = \frac{1}{w/2}\left(\frac{P_1.x - P_2.x}{P_1.y - P_2.y}\right) \quad (2)$$

Note that the offset and gradient are approximations and normalized to a value between -1 and 1.

There are a number of special cases that need to be considered. For example, if the path is horizontal in the image, then there are two possibilities for the gradient of the path. In this case, the gradient is determined to be +1, which means that the robot will turn to the right to follow the path.

For a detailed description of the algorithm the reader is referred to [Thomson 2001].

Figure 4 shows the scanning that is done by our algorithm on a more complex example.

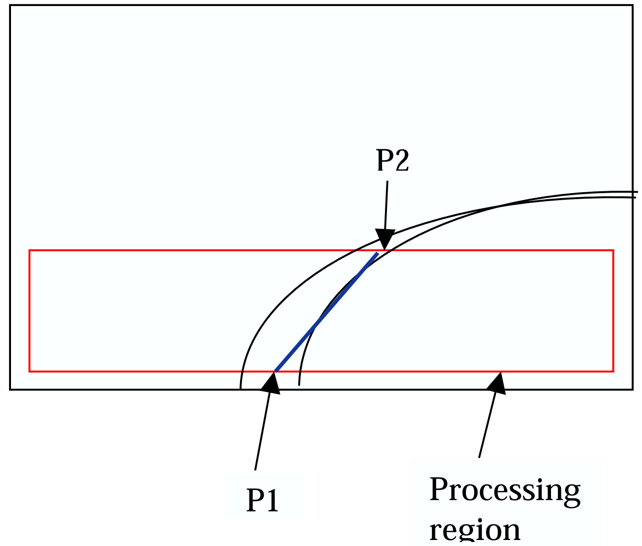

Figure 3: Determining the Offset And Gradient From the Image. Only the boundary of the region is processed.

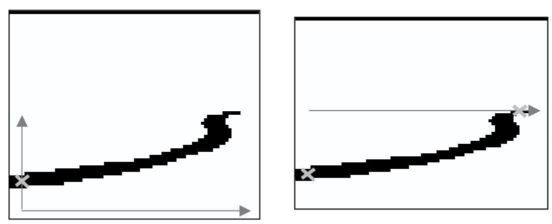

Figure 4: Image Processing Example

The algorithm scans for point P_1 on the bottom edge of the image. Since it is unable to find the path it scans on the left edge of the image and finds point P_1.

Point P_2 is found by scanning the top edge of the region. In this case, the point is found and no further processing is required. Otherwise, the right edge of the region is searched for the path.

If the path is to the right/left, then the robot should turn right/left. Also, if the path is sloping towards the right/left, then the robot should turn to the right/left. A problem occurs if for example the robot is to the right of the path, but the path is sloping towards the left. Therefore, the offset and the gradient need to be combined when computing the desired steering behavior.

A weighted sum of the offset and gradient of the image are computed and the resulting value is used as input to a proportional controller which selects the steering angle.

4 Evaluation

The system was evaluated using the path shown in Fig. 5. The path includes some challenging elements for the robot. For example, the sharp right turn on the top left is equal to the maximum turn radius of the robot and is close to the walls. This means that once the robot is away from the track at this point, it will not be able to recover until after the turn. The S-curves on the right hand side and the sharp right turn on the left side after a long straight where the robot builds up speed are other challenge points.

The accuracy of the path tracking control as well as the speed of the robot were computed using a global vision system overlooking the path. The position information from the global vision system is shown in Fig. 5.

There were two parameters whose value was not readily apparent in the algorithm described in section 3: (a) the look ahead distance of the robot, and (b) the relative weighting between the offset and gradient in the steering control.

To evaluate the quality of the path tracking the average speed of the robot as well as the error were recored and compared.

The size of the look ahead window was varied between 10, 20, and 30 pixels.

Each run consisted of five laps around the path and the results for each lap were averaged.

4.1 Average Error versus Offset Weighting (10 Pixel Lookahead Distance)

Given the geometry of the camera assembly on the robot, a 10 pixel lookahead resulted in the robot being able to see only a 2.5cm tall region, roughly 8.8 cm in front of the robot. The experiments showed that in this case, the offset information is more important than the gradient information. As can be seen in Fig. 6 shows a dramatic increase as the weighting of the offset information is increased. There is no further improvement if the gradient weighting is greater than 1.2 (or 60%).

Figure 7 shows the path of the robot using 50% and 100% weighting of the offset.

In contrast, the path when using only 5% weighting of the offset has very little resemblance with the actual path as is shown in Fig. 8.

Further investigation showed that given the narrow field of view, the gradient information alone does not present sufficient information about the actual slope of the path.

4.2 Average Errors versus Speed (10 Pixel Lookahead Distance)

We also investigated the relationship between the average speed of the robot and the average error (See Fig. 9). There was no significant correlation between the average or maximum speed and the average error. In fact, the fastest speed was achieved by the control algorithms with the best average error.

To our surprise increasing the height of the processing region to 20 and 30 pixels (13cm and 17cm respectively) did not lead to an improvement in performance. There was no statistically significant difference in performance when using 20 or 30 pixel look ahead.

4.3 Correlation between the Number of Steering Commands and the Average Error (10 Pixel Lookahead Distance)

We also investigated the required work by the controller when following the path. Intuitively, when following a path one would expect the quality of path tracking to improve with the number of

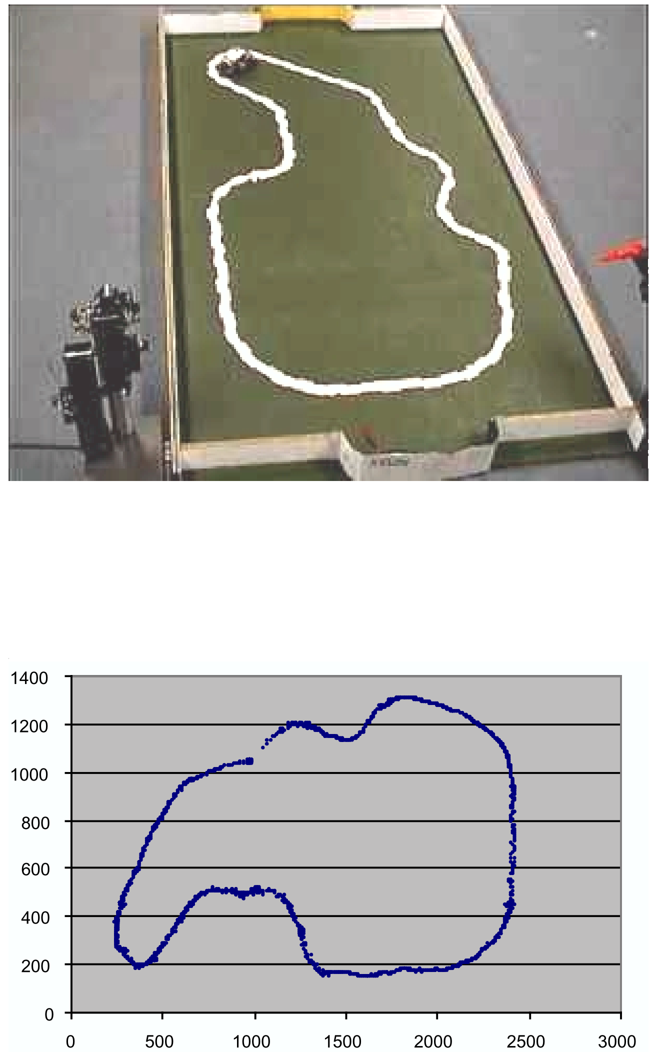

Figure 5: Evaluation Path and Information From the Global Vision System

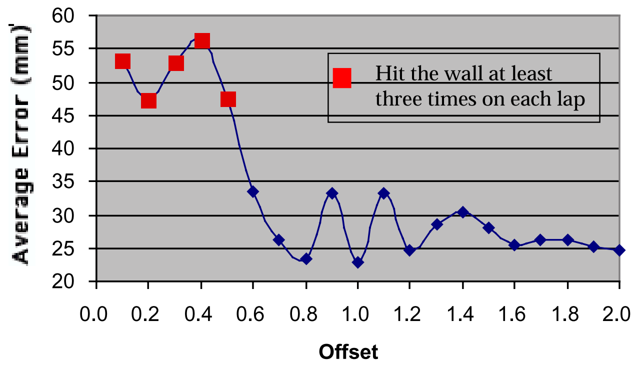

Figure 6: Average Error versus Offset Weighting (10 Pixel Look Ahead)

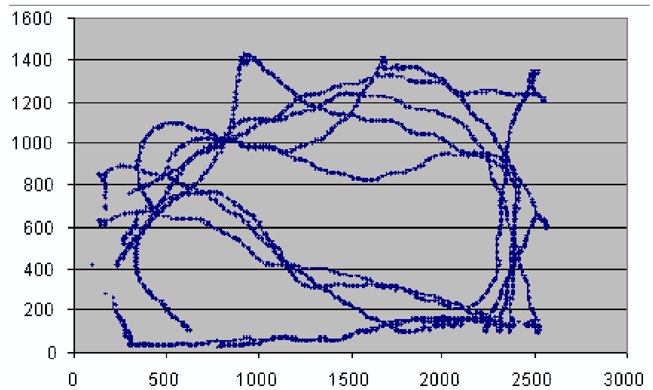

Figure 8: Path with 5% Offset Weighting (10 Pixel Lookahead Distance)

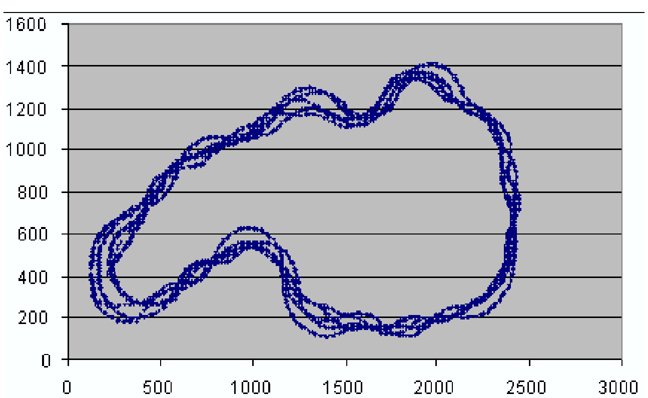

Figure 7: Path with 50% Offset Weighting (10 Pixel Lookahead Distance)

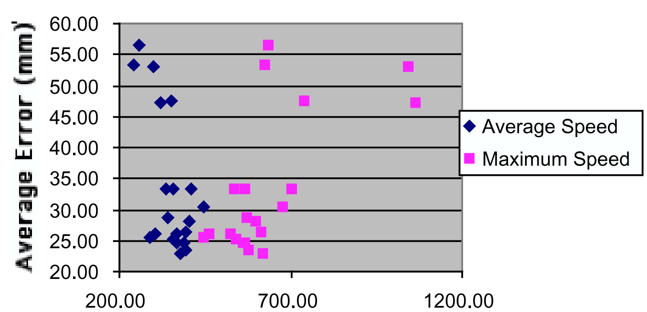

Figure 9: Avg. Error vs Speed (10 Pixel Lookahead Distance)

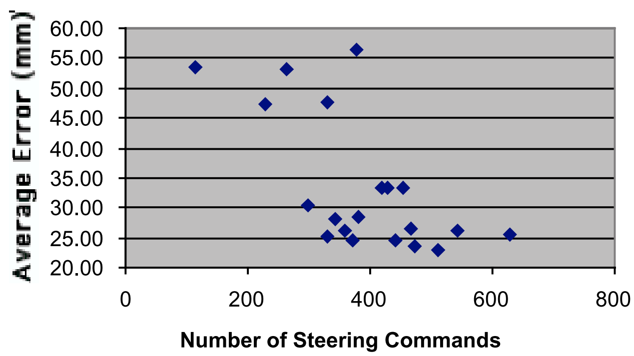

Figure 10: Number of Steering Commands vs Avg. Error (10 Pixel Lookahead Distance)

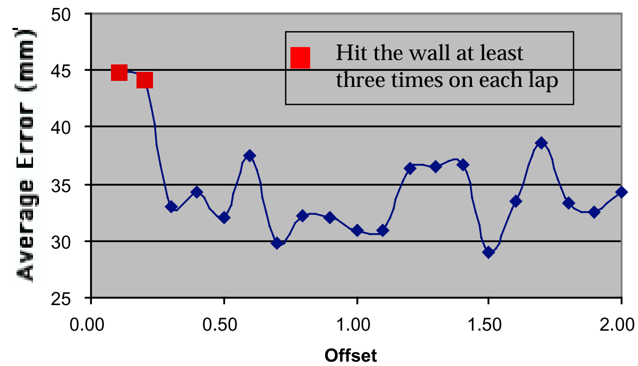

Figure 11: Average Error versus Offset Weighting (20 Pixel Look Ahead)

steering commands. However, in practice, radical changes to the steering angle makes the car unstable at higher speeds.

This overall improvement in performance can be seen in Fig. 10.

4.4 20 Pixel Lookahead Distance

The same experiments were repeated using a look ahead distance of 20 pixels, which is equivalent to approximately 13cm in front of the robot.

The results for these test were similar to the performance with 10 pixel look ahead. The additional information about the gradient did not have significant impact on performance.

Figure 12 shows the path of the robot using 75% weighting on the offset, which was the best result obtained with 20cm lookahead.

4.5 30 Pixel Lookahead Distance

By increasing the height of the processing region to 30 pixels from the bottom, the robot is able to look ahead a distance of 17cm in front of the robot.

One anomaly occurred with an offset weighting of 100%, which resulted in an average error of 37mm. Since in this case, the gradient is not used, it was surprising that the performance was influenced by the look ahead distance. However, this is due to the fact that the offset of the path is calculated as the average of the x coordinates of points P1 and P2. Increasing the look ahead distance starts to introduce errors into this approximation. The result for 30cm lookahead is shown in Fig. 13.

Figure 14 shows the path of the robot using 75% weighting on the offset, which was the best result that we obtained with a lookahead distance of 30cm. As can be seen in the plot, the robot missed the track once, but recovered by circling around.

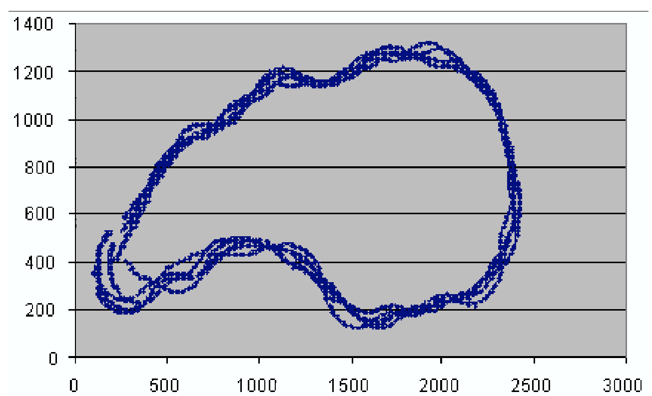

Figure 12: Path with 75% Offset Weighting (20 Pixel Look Ahead)

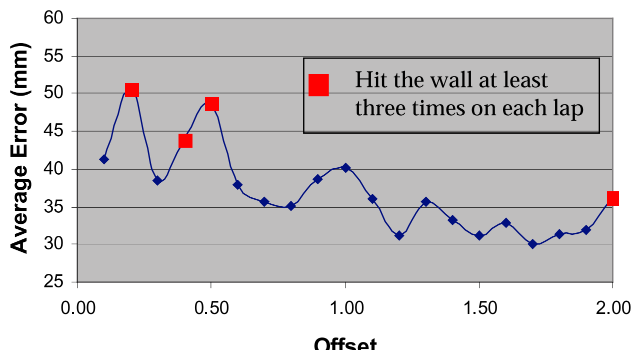

Figure 13: Average Error versus Offset Weighting (30 Pixel Look Ahead)

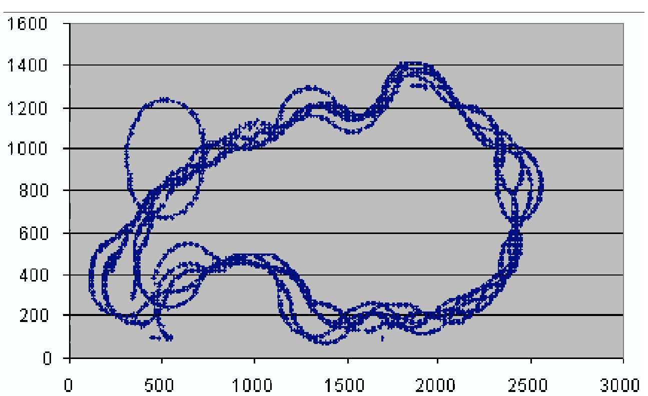

Figure 14: Path with 85% Offset Weighting (30 Pixel Look Ahead)

5 Conclusion

This paper describes a vision servoing system for path tracking control of mobile robots. Since it makes few assumptions about the underlying hardware it is applicable in situations where the dynamics of the robot are unknown or hard to measure.

The image processing algorithm has been designed for processors with very limited computational resources. This means that it can be implemented on mobile robots which only use small microprocessors.

The empirical evaluation shows that the algorithm can efficiently track a non-trivial path. The best results were obtained by weighting the offset and gradient information approximately equal. The algorithm performs well over a range of parameter settings.

We are currently extended the described system with a more complex control algorithm using Fuzzy Logic.

References

BALTES, J. 2001. 4 stooges. In *RoboCup-2000: Robot Soccer World Cup IV*, P. Stone, T. Balch, and G. Kraetszchmar, Eds. Springer Verlag, Berlin, 519–522.

BRÄUNL, T., 2002. Thomas bräunl's homepage. WWW, November. http://robotics.ele.uwa.edu.

(ED.), P. S., ASADA, M., BALCH, T., FUJITA, M., KRAETZSCHMAR, G., LUND, H., SCERRI, P., TADOKORO, S., AND WYETH, G. 2001. Overview of robocup-2000. In *RoboCup-2000: Robot Soccer World Cup IV*, P. Stone, T. Balch, and G. Kraetszchmar, Eds. Springer Verlag, Berlin, 1–28.

HASHIMOTO, K., AND NORITSUGO, T. 1997. Visual servoing of non-holonomic cart. In *Proceedings of the 1997 International Conference on Robotics and Automation*, 1719–1724.

KOBAYASHI, M., MIYAMOTO, Y., MISTUHASI, K., AND TANAKA, Y. 1996. A method of visual servoing for autonomous vehicles. In *Proceedings of the 4th Internation Workshop on Motion Control*, 371–376.

MA, Y., KOSECKA, J., AND SASTRY, S. 1999. Vision guided navigation for a non-holonomic robot. *IEEE Transactions on Robotics and Automation 15*, 3, 521–536.

MASUTANI, Y., MIKAWA, M., MARU, N., AND MIYAZAKI, F. 1994. Visual servoing for non-holonomic mobile robots. In *Proceedings of the 1994 IEEE Conference on Intelligent Robotics and Systems*, 1133–1140.

THOMSON, A. J. 2001. *A Path Following System For Autonomous Robots With Minimal Computing Power*. Msc., University of Auckland, Private Bag 92019, Auckland, New Zealand.