IEEE International Workshop on Intelligent Robots and Systems
IROS '90

Towards a Real-Time Navigation Strategy for a Mobile Robot.

M.D.Adams, P.J.Probert
Robotics Research Group
Department of Engineering Science
Oxford University
U.K.

1 Abstract

We describe the design of a real-time obstacle avoidance and navigation strategy for a mobile robot, using simulated time of flight infra-red data. Algorithms have been developed in order to overcome the undesirable effect of potential traps within the field, and a new approach for dealing with sensing whilst moving is demonstrated. At present we show simulated results of a vehicle moving under artificial potential force equations, which is designed to achieve modification of behaviour at a speed close to the normal operational speed of a real mobile.

2 Introduction

One of the fundamental problems in mobile robotics research is that of providing a reliable model of the environment, and then extracting the information necessary to guide a mobile to a given location. In this paper we describe our progress towards a method for real-time obstacle avoidance and navigation using a single rotating infra-red sensor.

We examine particularly the following issues:

- A method for overcoming potential traps caused by local minima within a potential field.
- A potential field algorithm, developed specifically for mobile robot navigation, whilst on the move.

To qualify as a *real-time* capability, we believe that a robot must be able to avoid an obstacle whilst it is traveling close to its normal speed, which for our vehicles is at about 0.4 m/sec.

We attack the problem of navigation in our environment, by applying an artificial potential field algorithm to sampled data provided by a continuously rotating sensor. Much previous work has focused upon the use of artificial potential fields in the navigation problem [3], [5], [8], but navigation using such an approach, whilst on the move, still appears to be a relatively unresearched issue.

3 An Optical Sensor for Navigation.

In order to bring our work into context, we describe in this section an infra-red optical range finder, which is currently under construction at Oxford. A simulation of the data produced from such a sensor is the starting point for our navigational ideas and experiments outlined in sections 4, 5 and 6.

In the past we have experimented with range finders which transmit infra-red light into the environment and produce an output proportional to the *amplitude* of the returned signal. Although the range finders suffer from limited range (approx 4 metres) due to the inverse square law attenuation of the light, their ability to resolve discontinuities within that range is good. This is because unlike sonar, it is possible to focus I.R light down to very narrow beam widths (approx 1.5°). The major problem with these amplitude measuring devices is their dependence upon the reflectivity of the detected surface. For example a change in colour or reflectivity on a wall is indistinguishable from a change in depth as far as the sensor receiver circuit is concerned.

A more appealing sensor, with similar beam width advantages, is an optical range finder currently in use at A.T & T Bell Laboratories, U.S.A, [4]. This sensor again uses infra-red light for detection purposes, but produces an output which is proportional to the *time of flight* of the beam, in a relatively inexpensive way. The sensor is therefore largely unaffected by the reflectance of the environment, except for cases of extremely high reflectance or absorbtivity.

Results from the sensor, which has been used for navigational experiments on a vehicle named Blanche [7], are impressive, in that depth readings correspond extremely well to actual object distances.

In the following sections we begin to develop our algorithms which make use of, at present, simulated data. In developing the navigational algorithms, we have produced a simulation which relates as closely as possible to the real situation of a single rotating sensor onboard a mobile vehicle.

4 The Potential Field Algorithm.

At present our input sensor data takes the form of simulated infra-red time-of-flight range readings, with which we wish to model our environment, whilst 'on the fly'. These facts give us strict conditions to follow, when considering the possible rate of information extraction from the sensor, speed of algorithm execution, and the rate at which a vehicle will respond to updated velocity control signals. When dealing with a real vehicle, it would be ambitious to expect any response to changes in the desired velocity control signal, more than five times per second. On the other hand, because we are dealing with an infra-red light sensor, the speed of retrieving the depth information is restricted only to the minute electronic delays presented by the infra-red driving equipment [7]. These considerations have had

a directing influence on our navigation strategy, which is explained in detail below.

With reference to figure 1, we consider the mobile to be the origin of a polar coordinate system, within which depth readings, provided by the continuously rotating sensor, are recorded.

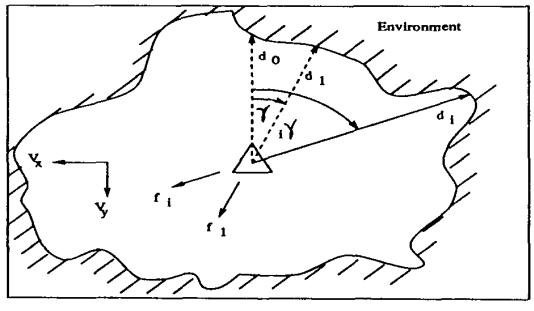

Figure 1: The mobile is the origin of a polar coordinate system, upon which force vectors are calculated directly from infra-red range readings.

The angle γ will be referred to as the *sampling angle*, and the radial distance d_i as the i th depth estimate, recorded directly from the sensor output. Corresponding to each range vector $\vec{d_i}$, a force vector $\vec{f_i}$ is immediately calculated obeying the simple artificial potential field equations:

$$|\vec{f_i}| = \frac{A}{|\vec{d_i}|^2} \tag{1}$$

$$\arg(\vec{f_i}) = \arg(\vec{d_i}) + \pi \tag{2}$$

where A is a fixed constant. At the end of a 360° sweep of the environment, the forces are *resolved* and *summed* in order to give an output which can be considered to be the desired new repulsive velocity components V_y and V_x, both parallel and perpendicular to the vehicle's centre line, caused by the current environment surrounding the mobile.

$$V_y = \sum_{i=1}^{2\pi/\gamma} f_i \cos(i\gamma) \tag{3}$$

$$V_x = \sum_{i=1}^{2\pi/\gamma} f_i \sin(i\gamma) \tag{4}$$

Note that the summation is completed when $i = (2\pi/\gamma)$ which must be an integral value when γ is measured in radians.

Figure 2 shows a mobile and the desired goal position within an office type environment, measuring approx. 8 metres square. The large cross is the desired goal position and each small cross corresponds to a single depth estimate, d_i taken within a single 360° scan. 180 such depth readings were recorded - ie: $\gamma = 2°$. The new position is also shown for the vehicle, which has moved under a constant attractive force from the goal and also the repulsive forces provided by the environment, shown in equations 3 and 4. In this new position, the vehicle is again ready to repeat the process. Figure 3 shows the completed path of the vehicle under this simple artificial potential field type navigation strategy.

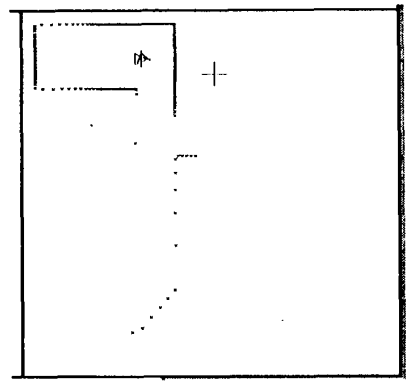

Figure 2: A simulated infra-red time of flight scan. Each cross shows a single depth reading, and a force is produced upon the vehicle, in the opposite direction. The scan shows 180 such depth readings.

As can be seen the mobile has not reached the goal, but has fallen into a *local minima* or *potential trap*. The force of attraction from the goal is counter balanced by the forces of repulsion from the wall, in front of the mobile.

5 Goal Relocation

In the past many researchers have offered methods for overcoming potential traps of this nature, most of which require a-priori knowledge of the environment, before the sensing process begins, [8], [6]. An alternative method recently suggested by Borenstein [2], is that of wall following when a potential minima is reached.

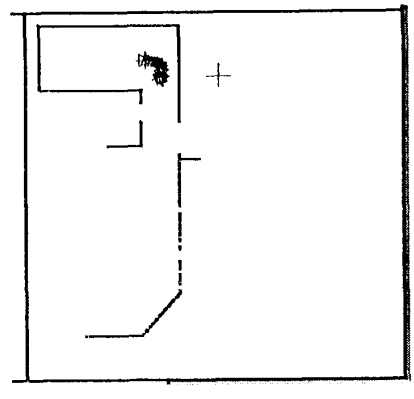

Figure 3: The force of attraction from the goal and the forces of repulsion from the wall, in front of the mobile, cancel. The vehicle has reached a local minima and is trapped.

In order to keep in context with the concept of a continuously rotating sensor, we introduce a method of *goal relocation*, in which the goal is temporarily moved according to the sensed geometry of the robots current environment. From odometric

estimates and a single scan of the environment, the algorithm decides whether or not the robot has a clear path to its goal. If it hasn't, then the goal relocation algorithm 'looks for a gap' and temporarily moves the goal into that gap. More specifically, the robot records the depth estimate d_i in the direction of the goal, and notes whether or not this *critical depth* is less than the radial distance to the goal, observed from long term odometric estimates. If it is, then the next depth estimate, d_{i+1} is recorded as normal, and the third range reading, d_{i+2} can be found in terms of d_{i+1} and d_i, if it is to lie on the same straight line. Mathematically, if:

$$d_{i+2} > \frac{d_i d_{i+1}}{(2d_i \cos\gamma) - d_{i+1}} \tag{5}$$

then a gap or convex corner has been found. A temporary goal is then relocated after depth reading d_{i+2} was recorded, in order to allow for the width of the robot. This temporary goal is shown as the largest cross in figure 4.

Again the small crosses show sensor readings from the environment, but the slightly larger crosses show from top to bottom, the detection of the *critical depth*, to the temporary relocation of the goal. (The smaller + to the right is the original goal).

Figure 5 shows the almost completed path of the vehicle.

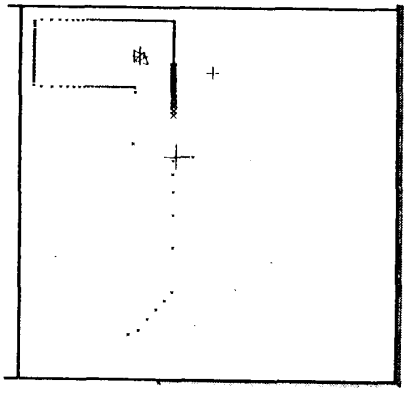

Figure 4: A single scan of the environment, showing the effect of goal relocation.

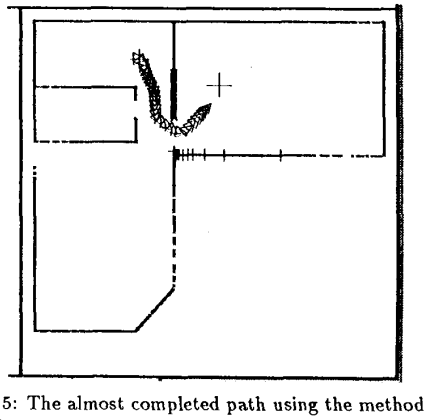

Figure 5: The almost completed path using the method of goal relocation.

The small +'s show the history of the temporary relocated goals, throughout the path. Clearly these trace out a set of 'beacons' or 'aim points' that the robot has used whilst navigating. The advantages of effectively splitting the path using this technique is clearly shown, and offers a solution to many potential trap situations.

6 The Effect of Motion

So far we have considered strictly stop - scan - move - stop type motion for a mobile robot. In a real situation it would be more realistic to consider the sensor data acquisition process taking place during vehicle motion.

In order to predict the next velocity, the results from the previous 360° scan are used. If the scanner (on board the mobile) has moved whilst taking that scan, then clearly each depth reading will incur some error, when compared to the corresponding depth reading, d_i that would have resulted with the vehicle stationary in its position at the end of that scan. This effect is shown in an exaggerated form in figure 6.

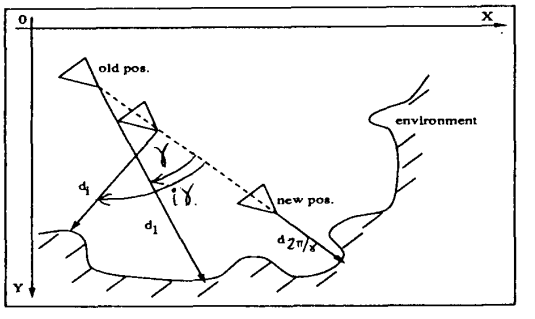

Figure 6: The effect of motion upon the sensor data acquisition process.

In order to overcome the distortion incurred due to the motion of the vehicle, we rewrite equations 3 and 4 as follows:

$$V_y = \sum_{i=1}^{2\pi/\gamma} \beta_1^{((2\pi/\gamma)-i)} \frac{A}{d_i^2} \cos(i\gamma) \tag{6}$$

$$V_x = \sum_{i=1}^{2\pi/\gamma} \beta_2^{((2\pi/\gamma)-i)} \frac{A}{d_i^2} \sin(i\gamma) \tag{7}$$

The weighting constants β_1 and β_2 have been chosen in order to adjust each depth reading as it arrives from the sensor. Each value of β is raised to the power $(2\pi/\gamma - i)$, in order to take into account the fact that the greatest error will most likely be in the first depth readings of each scan, when compared with those that would have been recorded with the vehicle stationary in the new position (figure 6). Hence when $i = 2\pi/\gamma$ (ie: at the end of a scan), $\beta^{(2\pi/\gamma - i)} = 1$, meaning that the last depth reading is not adjusted at all, since it is correct. So by premultiplying each depth reading d_i by $\beta^{(2\pi/\gamma - i)}$ we apply maximum correction to the initial readings (small values of i) and least correction to the final readings ($i \approx 2\pi/\gamma$).

Our reason for choosing the weighting constant, β in the above equations, is to distinguish this method of navigation as

a truly *real-time* process. A technique used in the past for reducing the effect of vehicle motion during sensor data acquisition, was to take the vector sum of each depth reading and the vehicle's velocity [4]. This is used to predict what the corresponding depth reading could be, were the mobile stationary in its initial position. In a real-time situation, to read in displacement measurements from the odometers, and calculate vector summations between successive depth estimates, could hinder the speed at which the infra-red sensor rotates, and hence updates the vehicle's velocity.

If a scan were taken on a stationary vehicle, β would be set to unity. Physically speaking, β is a weighting factor used to make the velocity component V_y more dependent upon the force vectors placed upon the vehicle from in front, than those from behind. This is done in order to take into account the fact that the vehicle is moving with an old velocity $K - V_y$, where K is the attractive velocity provided by the goal, into the on-coming environment.

In order to minimise the effect of motion upon the sensing process, we clearly need a solution for β, and therefore introduce the following mathematical model, for the y component of the repulsive velocity (the x component following a similar analysis).

Consider a system which has a continuous input signal, dependent purely upon the shape of the environment ie: system input $= \frac{1}{|\vec{d}_i|^2}$. We multiply this signal with a temporal cosine function, $A\cos(iT)$, which is non zero only at the discrete values of T when T equals γ, the sensor sampling angle. Then, premultiplying by the weighting term β and summing the above discrete signal values, we have:

$$\sum_{i=1}^{2\pi/\gamma} u(i) = \sum_{i=1}^{2\pi/\gamma} \beta^{((2\pi/\gamma)-i)} \frac{A}{|\vec{d}_i|^2} \cos(i\gamma) \qquad (8)$$

which is identical to the y component of our output velocity. Hence by using a single rotating sensor, we can inter-relate temporal and spatial sampling within a non-recursive filter. Figure 7 shows a model for our discretised control system, for our y component of velocity only.

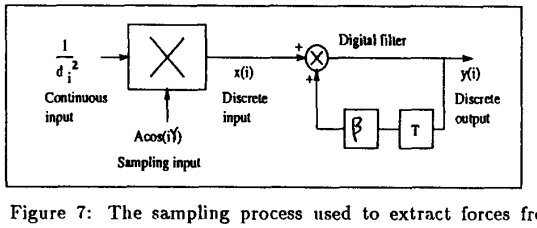

Figure 7: The sampling process used to extract forces from continuously rotating sensors.

The filter has a recursion formula, given by the difference equation:

$$y(i) = x(i) + \beta y(i-1)\,, \quad y(-1) = 0 \qquad (9)$$

giving a z transfer function:

$$\frac{Y(z)}{X(z)} = G(z) = \frac{1}{1 - \beta z^{-1}} \qquad (10)$$

where z is the delay operator e^{sT}, or in our case $e^{s\gamma}$.

Z-transform analysis of equation 10, for specific inputs, has produced interesting results, showing the dependence of output signal $Y(n)$, the repulsive velocity component in the direction of motion of the mobile, on the vehicle speed recorded during sensor data acquisition. Consider a general depth reading d_i, figure 6, and its dependence upon the environment and the vehicle's forward velocity V.

For any environment:

$$d_i = f(i, \gamma, V)$$

where $f(i, \gamma, V)$ represents a function of the sensor angle and the vehicle's velocity V.

We can see that from figure 7 that our input $x(i)$ to the digital filter is:

$$x(i) = \frac{1}{d_i^2} A\cos(i\gamma) = g(i, \gamma, V) \qquad (12)$$

The unilateral z-transform of a function $x(i)$ is given by:

$$X(z) = \sum_{i=0}^{\infty} x(i) z^{-i} \qquad (13)$$

so that:

$$X(z) = G(z, \gamma, V) \qquad (14)$$

From equation 10:

$$Y(z) = \frac{G(z, \gamma, V)}{1 - \beta z^{-1}} \qquad (15)$$

Upon taking the inverse transform and replacing $i = 2\pi/\gamma$ (since we are only interested in the last output from the filter for each scan) we get:

$$y(2\pi/\gamma) = h(\gamma, V, \beta) \qquad (16)$$

Note that if the vehicle were stationary in its final position (after the scan) then:

$$y(2\pi/\gamma) = h(\gamma, 1) \qquad (17)$$

since this is equivalent to the above case with $V = 0$ and $\beta = 1.0$. The error e in scanning whilst moving is given by:

$$e = h(\gamma) - h(\gamma, V, \beta) \qquad (18)$$

which is minimised when:

$$\frac{de}{d\beta} = h'(\gamma, V, \beta) = 0 \qquad (19)$$

Hence for minimum error, $\beta = j(\gamma, V)$, ie: β is a function of the vehicle's velocity V, and the sensor incrementing angle γ. If the function in equation 11 was known before the sensing process took place, then an exact solution for β could be found in terms of the velocity V of the vehicle. With a-priori knowledge of the environment, the above method could be used to minimise the errors in scanning due to motion.

In order to overcome the problem without a-priori knowledge however, we need an alternative means for finding β. Again refering to figure 6, we can consider the sensor readings recorded during the motion between the old and new positions, to be a combination of the sensor readings accumulated in the old and new positions whilst stationary, with the inclusions of small error terms:

$$\sum_{i=1}^{2\pi/\gamma} \frac{A}{d_{im}^2}\cos(i\gamma) = \sum_{i=1}^{\pi/\gamma} \frac{A}{(d_{ios}+\epsilon_1(i))^2}\cos(i\gamma) + \sum_{i=(\pi/\gamma+1)}^{2\pi/\gamma} \frac{A}{(d_{ins}+\epsilon_1(i))^2}\cos(i\gamma)$$

so that:

$$\sum_{i=1}^{2\pi/\gamma} \frac{A}{d_{im}^2}\cos(i\gamma) \approx \sum_{i=1}^{\pi/\gamma} \frac{A}{d_{ios}^2}(1-\frac{\epsilon_1(i)}{d_{ios}})\cos(i\gamma) + \sum_{i=(\pi/\gamma+1)}^{2\pi/\gamma} \frac{A}{d_{ins}^2}(1-\frac{\epsilon_2(i)}{d_{ins}})\cos(i\gamma)$$

where d_{im} is a general depth reading recorded whilst moving, d_{ios} is a general depth reading recorded whilst stationary in the old position (figure 6), d_{ins} is a general depth reading recorded whilst stationary in the new position, and the small errors $\epsilon(i)$ depend upon the environment.

6.1 Experimental Results

Figures 8 and 9 show plots of the value of β which minimises the velocity error, due to scanning whilst moving, verses the velocity difference from the previous two scans.

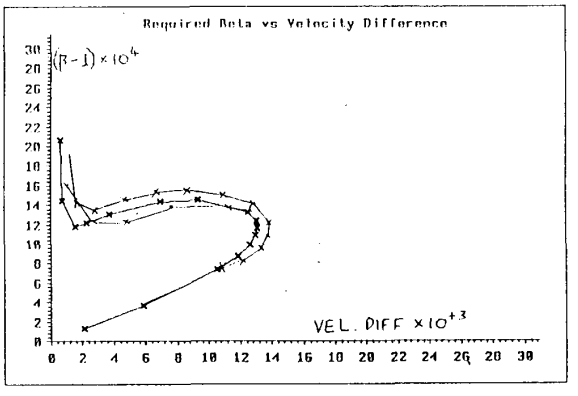

Figure 8: Curves showing the variation of β with velocity difference for different environments, as a mobile approaches a boundary. Each curve converges to the same line as the repulsive velocity exceeds 0.1 m/s.

As can be seen, for tests carried out within different environments, each curve converges to the same line, when the repulsive velocity exceeds approx 0.1 m/s. When the robot is not close to any walls or obstacles, the repulsive velocities are small and the value of β cannot be predicted accurately within different environments. When a vehicle is within close proximity of obstacles or walls, the last recorded repulsive velocity is high and β is approximately the same for all environments, at that velocity difference. A final graphical analysis, shown in figure 10, shows the error in the computed velocities, when leaving $\beta = 1.0$ whilst scanning during motion.

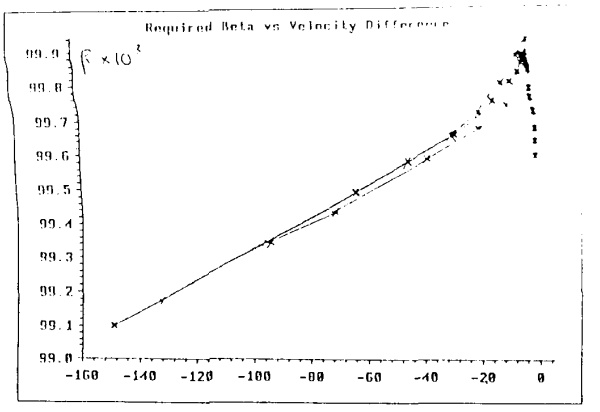

Figure 9: Curves showing the variation of β with velocity difference for different environments, as a mobile moves away from a boundary. Each curve converges to the same line as the repulsive velocity exceeds 0.02 m/s in magnitude.

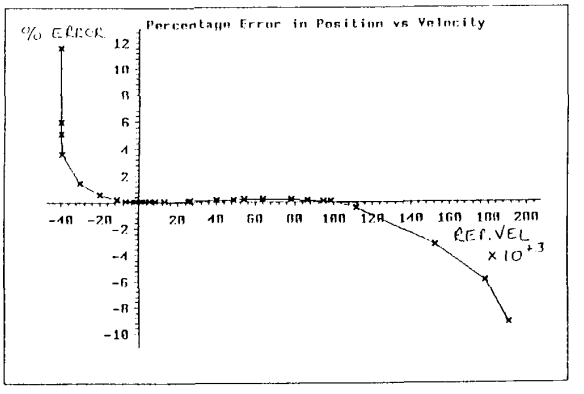

Figure 10: The variation in error of the new calculated velocities verses actual repulsive velocity, when the value of β is constantly set equal to 1.0. When the repulsive velocities are small, the error in leaving β set to 1.0 is negligible.

As can be seen the error is extremely small except when the magnitude of the repulsive velocity exceeds approximately 0.1 m/s (or 0.02 m/s when moving away from a boundary). Hence we do not need to predict β until the curves shown in figures 8 and 9 converge, or physically speaking when a mobile is close to any point within its environment.

Figures 11, 12 and 13 demonstrate the effect of *gain scheduling* the value of β from the look up tables shown in figures 8 and 9, [1].

In figure 11 we see the original stop - scan - move - stop type motion explained earlier. This gives us the desired path for our robot, when its motion is continuous. In figure 12 the value of β is set permanently to 1.0 to show that gain scheduling

is necessary when a mobile approaches in close proximity to its environment. Finally figure 13 shows the path of the vehicle taken when β is adjusted according to the look up tables shown earlier.

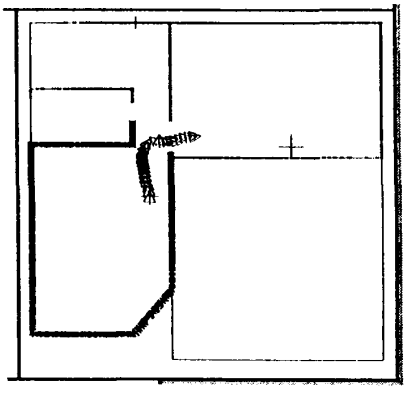

Figure 11: Stop - scan - move type motion using the method of goal relocation to navigate using the potential force equations shown earlier. This is the desired path for a mobile.

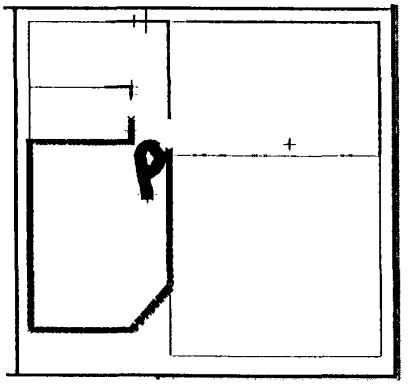

Figure 12: The vehicle is scanning whilst moving, with the value of β set constantly to 1.0. Its rotation near to the convex edge is incorrect due to sensor reading distortion.

7 Conclusions

We have presented in this paper new algorithms for processing infra-red time of flight sensor data, from a continuously rotating range finder. The algorithms have been developed in order not to restrict the flow of data from the ranger, and to run using local sensor data only. Several experiments using simulated data have demonstrated that:

- artificial potential force equations are capable of navigating a mobile successfully, if the goal is placed in a suitable position. A method for automatically relocating temporary goals has been demonstrated.
- distortions within the repulsive velocities, caused by scanning whilst moving, can be dealt with efficiently, and in real-time, by weighting each depth estimate from the environment, as it is read from the sensor.

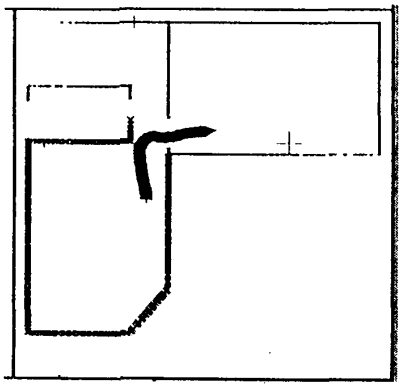

Figure 13: The vehicle is scanning whilst moving, and the value of β is being adjusted according to the look up tables. The effect is clearly shown.

In the near future we intend to examine the response of a real vehicle to changes in velocity demand produced by the algorithm. We further intend to extend our research using real infra-red time of flight data.

8 Acknowledgements

We wish to thank the many people within the Oxford A.G.V laboratory for contributing towards the project. M.D Adams is supported by an SERC grant.

References

[1] Karl J. Astrom and Bjorn Wittenmark. *Computer Controlled Systems*. Prentice-Hall, 1987.

[2] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. In *IEEE Trans. Systems Man and Cybernetics*, pages 1179–1187, 1989.

[3] R.A Brooks. A layered intelligent control system for a mobile robot. In *Third Int. Symp. Robotics Research*, MIT Press, Gouvieux, France, 1986.

[4] Ingemar J. Cox. Blanche: an autonomous robot vehicle for structured environments. In *IEEE J. Robotics and Automation*, page 978 to 982, 1988.

[5] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 500–505, St. Louis, March 1985.

[6] P. Khosla and R. Volpe. Superquadric artificial potentials for obstacle avoidance and approach. In *IEEE J. Robotics and Automation*, page 1778 to 1784, 1988.

[7] Winston L. Nelson and Ingemar J. Cox. Local path control for an autonomous vehicle. In *IEEE J. Robotics and Automation*, page 1504 to 1510, 1988.

[8] Charles W. Warren. Global path planning using artificial potential fields. In *IEEE J. Robotics and Automation*, page 316 to 610, 1989.