COVER SHEET

This is a manuscript version of this paper. The paper was first published as:

Taylor, Trevor and Geva, Shlomo and Boles, Wageeh W. (2005) Early Results in Vision-based Map Building. In Murase, K. and Sekiyama, K. and Kubota, N. and Naniwa, T. and Sitte, J., Eds. Proceedings 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005), pages pp. 207-216, Awara-Spa, Fukui, Japan.

Early Results in Vision-based Map Building

Trevor Taylor, Shlomo Geva, Wageeh W. Boles

Queensland University of Technology, Brisbane, Australia, T.Taylor@qut.edu.au

Abstract

Two key objectives of robot vision are autonomous navigation and mapping. Digital cameras have become relatively cheap in recent years and have appeared in a variety of consumer devices, such as PDAs (Personal Digital Assistants). It therefore makes sense to try to build systems that use vision as their primary input instead of the more traditional sonar and infrared sensors that have been used in the past. The camera can also be used for a range of other tasks. Service robots and toys typically operate in an indoor environment but rarely have a map of their environment when they are first turned on. This paper therefore addresses the problem of vision-based mapping where a camera has deliberately been chosen as the only sensor.

1 Introduction

Our work demonstrates the exploration and mapping of an unknown environment using only video images from a single, cheap, colour camera. The approach involves performing pirouettes (spinning on the spot) to create the equivalent of traditional sonar sweeps, and limiting the moves that the robot can make to avoid real-time decision making.

Vision has several advantages over other sensors for use in navigation and mapping – it is passive, low-power, less susceptible to noise, and most significantly it provides colour information that can assist in distinguishing between obstacles. It also has some unique features that make it more difficult to use, several of which are addressed in this paper.

Creating a map requires accurate knowledge of the distances from the robot to the surrounding obstacles. In the past, the range to obstacles has often been determined using sonar, infrared or laser sensors [4]. Stereoscopic systems which derive a depth map from the images from a pair of cameras, e.g. [3], have also been used. However, for reasons of low cost, simplicity, and due to size constraints we use a single camera.

In our experimental environment the vision system consists of a small wireless colour camera mounted on a two-wheeled Yujin soccer robot as shown in Fig. 1a. The camera is smaller than the 9V battery it runs off.

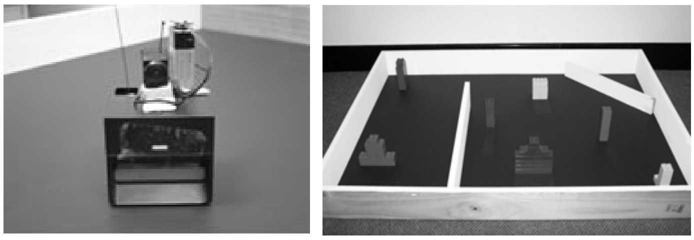

Fig. 1. (a) Yujin Soccer Robot with Wireless Camera **(b)** Test Environment

The robot can roam around a test environment which is similar to a robot soccer field, but without the lines on the floor (Fig. 1b). Vision processing takes place on a PC, which controls the robot via a wireless modem (just like robot soccer). Eventually we will use a PDA with a camera to control a larger robot. Therefore the test environment is a scale model.

The first objective of the vision system is to distinguish the floor from obstacles in order to derive a Radial Obstacle Profile (ROP) [13]. The ROP is a local map of the surrounding obstacles with the robot at the centre and is similar to a sonar or radar sweep, as shown by the thick line in Fig. 2.

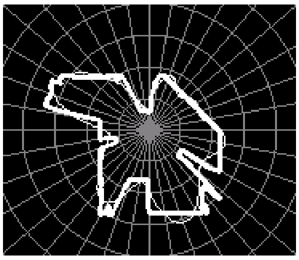

Fig. 2. Radial Obstacle Profile (ROP) produced by a Pirouette

To obtain the ROP, we assume that the ground plane is flat and that obstacles sit on the floor, which is true for indoor environments. This constraint results in a unique Inverse Perspective Mapping from the pixel coordinates in the image to real-world coordinates on the floor. (See [13] for the details.) The mapping for all pixels can easily be calculated and stored in a lookup table. This data can be used to create a “top down” view of the surrounding obstacles (Fig. 2).

The local map information from the ROP is then incorporated into an Occupancy Grid (a global map) by the application of Bayes rule with an appropriate sensor model. This is a classical approach [11] which should not require further explanation except for the sensor model (see below).

Note that new information is continually being added to the map, and so moving objects will not remain in the map permanently. In effect the robot is constantly learning about its environment.

It is well known that mapping by autonomous robots also requires simultaneous localization so that they do not lose their position due to accumulated odometry errors, e.g. [1]. A localization process will therefore be the next step in our work.

For the purposes of development and testing we have written a simple simulation using OpenGL to create artificial camera images. Simulation is not subject to odometry errors, and so localization is not required.

We have developed an exploration algorithm based on the Distance Transform which ensures that all unknown space which is reachable will be explored. The mapping and exploration algorithms have been successfully tested on the real robot and the results are shown later in the paper.

2 Practical Considerations

Although cameras and computers have improved enormously in recent years, there are still limitations that restrict what can be accomplished. Some of these are discussed in this section.

2.1 Camera Limitations

Many cheap cameras have poor quality lenses. The most obvious effect is radial distortion, which causes problems with the Inverse Perspective Mapping. Camera calibration can be performed using the Intel Open Computer Vision software, and the distortion can be removed from the images. An example of correcting for radial distortion is given in Figs. 3a and 3c.

Another common effect in cheap cameras is vignetting. This problem arises because the camera's CCD array is rectangular but the lens is round. It appears as a darkening of the image towards the corners. Although this problem is mentioned in textbooks on computer vision, e.g. [5], no solutions are offered. We have adopted a very simple approach that improves image quality, but cannot completely remove the vignetting effects. It is loosely based on the work of Yu [14].

Pixel values in the image are adjusted by a factor that is proportional to the square of the distance from the center of the image. This is done independently for each of the Red, Green and Blue components, which allows compensation for incorrect colour tint to be applied at the same time. The resulting improvement can be seen in Figs. 3a and 3b.

Our camera only has a 60º field of view, which is fairly typical of such cameras. It must therefore be tilted downwards to see the floor in front of the robot, but even then there is a blind area in front of the robot. Tilting the camera also introduces complications due to perspective effects which are discussed below in Section 5.

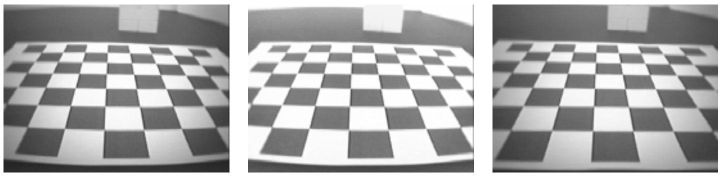

Fig. 3. Image correction **(a)** Original Image **(b)** Anti-Vignetting **(c)** Undistorted

2.2 Computational Requirements

Processing of images is computationally intensive which makes real-time steering problematic. To eliminate this problem, our robot is restricted to only two types of movements: rotation on the spot, or forward/backward moves in a straight line. We refer to this as piece-wise linear motion. It also fits in well with our use of Distance Transforms for exploration.

There are advantages to having the robot stationary when it captures images. Firstly, there is no vibration to affect the position of the camera. For example, if the camera moved up or down slightly then the pixel locations of the obstacle edges in the image would also change resulting in errors in the Inverse Perspective Mapping.

Secondly, the shutter speed of the camera cannot be controlled and it is fairly slow. This means that the images are blurred if the robot is moving when they are captured. Blurring tends to suppress edges, making it harder to detect obstacles. However, with piece-wise linear motion the robot is not moving when images are captured.

3 Mapping the Local Environment

The first stage in creating a global map is to produce a map of the surrounding environment, i.e. a Local Map. The robot is instructed to turn around on the spot in a series of small rotations, i.e. 12 steps of 30º each, to provide a good overlap and allow averaging of the results. A Radial Obstacle Profile (ROP) is obtained from the resulting sequence of images [13].

We assume that the floor is reasonably uniform in colour and that this colour is distinct from the obstacles. The actual colour of the floor is not important because the robot determines during its initialization sequence.

Despite the fact that this is a significant limitation, it is quite common in the literature, e.g. [8, 10] and reflects the difficulty of handling textured surfaces. High-frequency textures can be eliminated by applying a low-pass filter. Horswill [7] referred to this as the “Background Texture Constraint” and expressed it as a lower bound on the spatial frequency of the floor texture. Martin [12] found by applying a genetic algorithm that the best means of detecting obstacles in this case was to use an edge detector.

Therefore, to segment the image we use a Canny edge detector to locate edges. A flood fill is then performed from the bottom of the image upwards until an edge is encountered.

Care has to be taken to ensure that the fill is not performed inside obstacles. Therefore the colour of the seed pixel is first checked against the known floor colour. If it differs significantly from the floor, then a search is performed across the bottom of the image until the floor is found.

3.1 Sensor Model for Vision

Errors due to quantization and noise in the video image mean that the exact location of obstacles is not certain. We consider the case of a small error, such as plus or minus one pixel. Examining the inverse perspective mapping lookup table for a 1.0cm grid with our particular camera geometry, we found that the error in the radial distance was less than 0.5cm for ranges up to about 35-40cm. (It varies across the image).

We therefore ignore information beyond a maximum distance, e.g. 40cm. If the range to an obstacle is less than this, then the sensor model looks like Fig. 4. This is in fact the ideal sensor model for a range sensor. A slight "blurring" of the peak might be more realistic, with the extent of the spread increasing with the range from the camera. In the immediate foreground, however, spatial resolution is far better than the grid size.

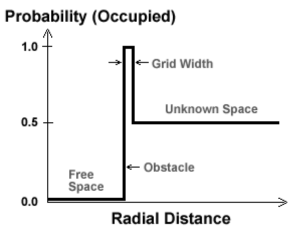

Fig. 4. Sensor Model for Vision

Because of the camera's limited field of view, there is a blind area in front of the robot. When the robot moves too close to an obstacle, it might not be able to see the bottom of the obstacle, i.e. the obstacle might fill the entire image. In this case, the only conclusion that can be drawn is that there is an obstacle somewhere in the blind space, so the blind area is marked with a probability halfway between Unknown and Occupied.

Systemic errors can also occur, such as incorrectly identifying the floor, or, more commonly, identifying obstacles where there are none, e.g. shadows. These errors are very hard to quantify, but the probability of them occurring should be low. (Otherwise the vision system is seriously impaired). They have therefore been ignored in the sensor model.

4 Exploration – Building a Global Map

The robot must be able to map the environment autonomously, which requires an exploration algorithm. We have chosen to use the Distance Transform because it indicates when exploration is complete, i.e. there will be no more unknown spaces that are reachable.

The first step is to convert the map to Configuration Space. This is done by expanding the obstacles in the map by the radius of the robot so it is not possible for the robot to collide with an obstacle as it follows a path. Fig. 5 is a Configuration Space map after the robot has finished exploring. The map uses the standard convention for occupancy grids where white represents free space, black is obstacles and grey is unknown.

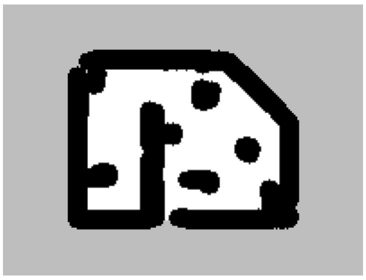

Fig. 5. Configuration Space Map

All of the unknown spaces in the map are marked as goals. Then a Distance Transform is performed from the robot's current location, thereby finding the shortest path to the nearest unknown space.

The Distance Transform as implemented using the Borgefors algorithm [2] is often referred to as a two-pass process (forward and backward) that is similar to convolution. However, for some maps it is necessary to make multiple passes [9]. In our experiments this has rarely been a problem.

In the literature, it is often assumed that the map cell size matches the cell size for robot motions. If the cell size is small compared to the robot, this leads to paths that hug the walls and frequent direction changes.

These "too close" paths, as they are called by Zelinsky [15], present two problems. Firstly, the robot might swipe an object in passing, especially when moving around corners. The second problem for our camera geometry is that the robot might move so close to an obstacle that the base of the obstacle falls into the blind area in front of the robot. This makes it impossible for the robot to determine exactly where the obstacle is located.

To address the first problem, the robot must always "look before it moves". This means that the free space in front of the robot must be re-assessed immediately before each forward motion. A useful side-effect of this is that the robot should detect objects that move into its field of view, i.e. the environment does not have to be static.

The second problem can only be tackled by using active vision, i.e. moving to a better vantage point in order to actually see the exact location of the obstacle. The simplest motion is backwards, provided that the space behind the robot is already known to be clear based on previous information. We have not yet implemented this approach.

Using a larger cell size for the Distance Transform is another way to minimize the possibility of these problems occurring. This results in a margin around obstacles, as can be seen in the maps of Fig. 6. The robot is the small circle in the bottom left and its path is shown starting from near the middle of the map.

Fig. 6. Completed Maps **(a)** Simulation **(b)** Real Robot

Larger cell sizes also reduce the number of moves. Fewer moves should reduce the cumulative odometry errors. However, large cell sizes prevent the robot from entering narrow corridors or passing through tight gaps.

Fig. 6b shows a map from the real robot starting at about the same position as in the simulation (Fig. 6a). The map is distorted due to odometry errors. Also, there are spikes at the edges that occurred because the robot's shadow fell on the wall and it interpreted this as the floor (which is black). However, the basic shape of the map and the path followed are similar to the simulation, which confirms the algorithms.

5 Future Work

There are several areas that we are actively investigating. The following sections outline only a few of them.

5.1 Using Colour

Colour is a powerful tool for identifying and distinguishing between obstacles. We plan to use the colour of obstacles to assist in the localization process by incorporating it into the ROP. However, there is a significant problem referred to in the field as Colour Constancy, or the ability of humans to perceive a wide range of colours as actually resulting from the same base colour. To address this, we use the Improved HLS system [6] and quantize the Hue into 12 colours. Unfortunately, Hue has no representation for Black and White which are common colours in man-made environments. Therefore, when the saturation is too low or too high we quantize the “colour” into Black, White or Grey.

5.2 Image Understanding

Vision involves more than Image Processing – it requires understanding what is seen. For example, when viewing a scene with a camera that is tilted downwards, the vertical edges of obstacles do not appear vertical. Perspective effects result in edges in the image that differ from vertical by an amount which depends on how far away the obstacle is from the camera and also how far it is from the centre of the image. See Fig. 7 below.

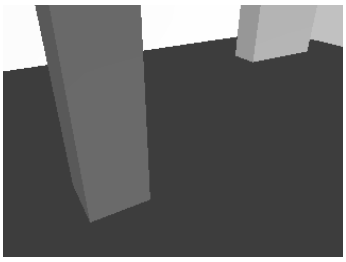

Fig. 7. Perspective Effects on Vertical Edges

Currently, our system ignores edges that are close to vertical when constructing the ROP, but it can still become confused in extreme cases such as in Fig. 7. Because near-vertical edges can also arise due to walls, we are investigating ways of recognizing obstacles so that the inclination of their vertical edges can be calculated and compared with the image.

5.3 Global Localization Using the ROP

Preliminary tests have been performed using a Discrete Cosine Transform (DCT) of the ROP for localization. The approach is as follows:

- Each time an ROP is obtained, perform a DCT and then throw away all but the first 20 coefficients, i.e. a low pass filter. (The effect can be seen as the thin line in Fig. 2, which is much smoother than the thick line.)
- Keep a table of these coefficients and the corresponding locations as the robot explores. (This table is maintained across successive test runs.)
- The coefficients can be compared in a simple nearest neighbour sense to obtain the robot’s most likely location for each new ROP.

In simulation, this process works reasonably well. However, the DCT is phase-sensitive, i.e. it is sensitive to the orientation of the robot. Therefore, we have also experimented with Hu Moments which are invariant to rotations, but they are not as good a predictor of the location as the DCT.

6 Summary and Conclusions

Our preliminary work has illustrated map construction using vision as the sole sensor in both simulated and real environments. We have outlined several of the difficulties associated with cheap cameras.

We have introduced a new variation on the use of the Distance Transform for exploration of an unknown environment. In this approach the map is not known initially, and the Distance Transform grid size is larger than the Occupancy Grid size.

Use of the Discrete Cosine Transform of the ROP shows promise as a means of identifying the robot's location on a global map, and we intend to investigate this further.

Building accurate maps is easy in simulation because there are no odometry errors. However, real robots quickly lose track of their position as they move around. Therefore it is necessary to perform a Localization step, and this will be the next stage of our research. Once this is complete, we intend to run multiple robots simultaneously to speed up the mapping process through collaboration.

References

1. Borenstein J, Everett HR, Feng L, Wehe D (1997) Mobile robot positioning: Sensors and techniques. Journal of Robotic Systems 14:231-249
2. Borgefors G (1986) Distance Transformations in Digital Images. Computer Vision, Graphics and Image Processing 34:344-371
3. Burschka D, Lee S, Hager G (2002) Stereo-Based Obstacle Avoidance in Indoor Environments with Active Sensor Re-Calibration. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA 2066-2072
4. Dudek G, Jenkin MRM (2000) Computational Principles of Mobile Robotics. Cambridge University Press
5. Forsyth DA, Ponce J (2003) Computer Vision: A Modern Approach. Prentice-Hall, Upper Saddle River, NJ
6. Hanbury A (2002) The taming of the hue, saturation and brightness colour space. In Proceedings of the 7th Computer Vision Winter Workshop, Bad Aussee, Austria 234-243
7. Horswill ID (1994) Collision Avoidance by Segmentation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Munich, Germany 902-909
8. Howard A, Kitchen L (1997) Fast Visual Mapping for Mobile Robot Navigation. In Proceedings of the IEEE International Conference on Intelligent Processing Systems, Beijing, China 1251-1255
9. Jarvis RA (1993) Distance Transform Based Path Planning For Robot Navigation. In: Zheng YF (ed) Recent Trends in Mobile Robots, Series in Robotics and Automated Systems. World Scientific 11:3-31

10. Lorigo LM, Brooks RA, Grimson WEL (1997) Visually-guided obstacle avoidance in unstructured environments. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Grenoble, France 1:373-379
11. Moravec HP, Elfes A (1985) High Resolution Maps from Wide Angle Sonar. In Proceedings of the International Conference on Robotics Automation, St. Louis, MI, USA 116-121
12. Martin MC (2001) Visual Obstacle Avoidance Using Genetic Programming: First Results. In Proceedings of the Genetic and Evolutionary Computation Conference 1107-1113
13. Taylor T, Geva S, Boles WW (2004) Vision-based Pirouettes using the Radial Obstacle Profile. In Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics, Singapore 147-152
14. Yu W, Chung Y, Soh J (2004) Vignetting distortion correction method for high quality digital imaging. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK 3:666-669
15. Zelinsky A (1992) A mobile robot exploration algorithm. IEEE Transactions on Robotics and Automation 8:707-717