

Economical and Cautious Approaches to Local Path Planning for a Mobile Robot

P. Grant and P. H. Mowforth

The Turing Institute, 36 North Hanover St., Glasgow G1 2AD, Scotland
tel: +41-552-6400, fax: +41-552-4245, e-mail {paul,boffin}@turing.ac.uk

This paper describes a mobile robot project which couples both a vision system and a planning system. In particular, we describe how a world model is constructed in the vision system, and how this is utilised in the generation of a point-to-point path for the robot. We show, that through similar techniques, two qualitatively differing plans may be generated - namely the economical path, and the cautious path.

Most mobile robotic systems can be decomposed into a number of sub-systems, each performing a clearly defined task but with the overall goal of gearing perception to action. Figure 1 shows a picture of one of the Turing Institutes mobile robots, called Turnip 1, whilst Figure 2 illustrates these sub-systems.

A clear distinction exists between a *world knowledge base* and a *world model*. The former codes abstract relational information concerning the world in a form which allows a) recognition, b) inference, or c) global planning. The latter codes geometric information concerning the world in a form which allows a) sensor fusion, b) obstacle avoidance or c) local planning. Whilst a world knowledge base may be learned or predefined, the world model provides a buffered summary of our immediated perceptions.

Planning may be further refined into *global planning* and *local planning*. Global planning is concerned with the generation of an ordered sequence of sub-goals culminating in the achievement of a required task. For navigation, these will typically be a list of landmarks connected by *goto* commands. Local planning is concerned with the generation of a point-to-point path between these landmarks. Whilst global planning utilises the world knowledge base, local planning requires sensory information obtained via the world model.

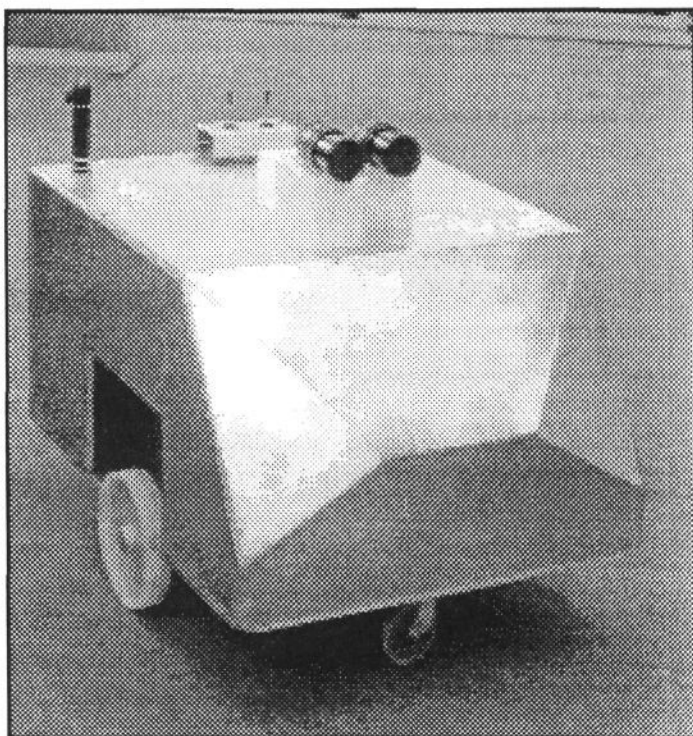


Figure 1: One of the Turnip 1 mobile robot prototypes in the Turing Institute.

This paper outlines an approach for the construction of a world model and, more specifically, how this may be utilised in the generation of a local plan.

1 Generating a world model

The Turnip 1 mobile features two forward facing cameras. The images are used as a basis for computing intrinsic scene information using stereo and motion data. The image matching is carried out using a Multiple Scale Signal Matching algorithm, *MSSM*[3], which provides smooth continuous solutions in the matched output. The algorithm

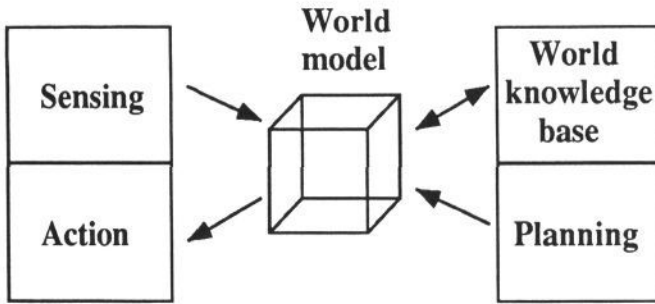


Figure 2: Major software subsystems of the mobile robot.

has several similarities to that described by Anandan [1]. Figure 3 shows the outputs of the signal matcher expressed as p, q and d maps. The p and q maps express the horizontal and vertical distortions respectively between successive frames from each camera whilst the d maps express the disparity information between stereo pairs of images. In this way, the p, q and d maps allow every visible patch of surface to be tracked smoothly and continuously in both time and space.

With each new frame, a new world model is constructed as a 3D voxel occupancy grid. As data fusion has already been achieved, there is no need to express probabilities in the structure. The occupancy grid, called TOSpace, is a three state model for which voxels are labelled as Transparent (free space), Opaque (solid surfaces) or unknown (occluded). The model uses the natural scaling offered by the signal matching results. As such, using only one scale, it uses a robot-centered, elastically deformed magnification bubble in which the world is expressed with ever coarser resolution away from the robot out to effective infinity. In order to preserve surface data described at other points in time and space, it is necessary to use a model which oversamples, i.e. hyper-resolution. At present, the visual short-term memory for the model uses only the last three frames.

For further details of the TOSpace world model see Shapiro and Mowforth (in preparation).

2 Local planning

TOSpace offers a useful framework for carrying out local planning. As the model is continuously being rebuilt after each perception cycle, all the potential obstacles in the environment are labelled.

Within the local environment of the robot, fol-

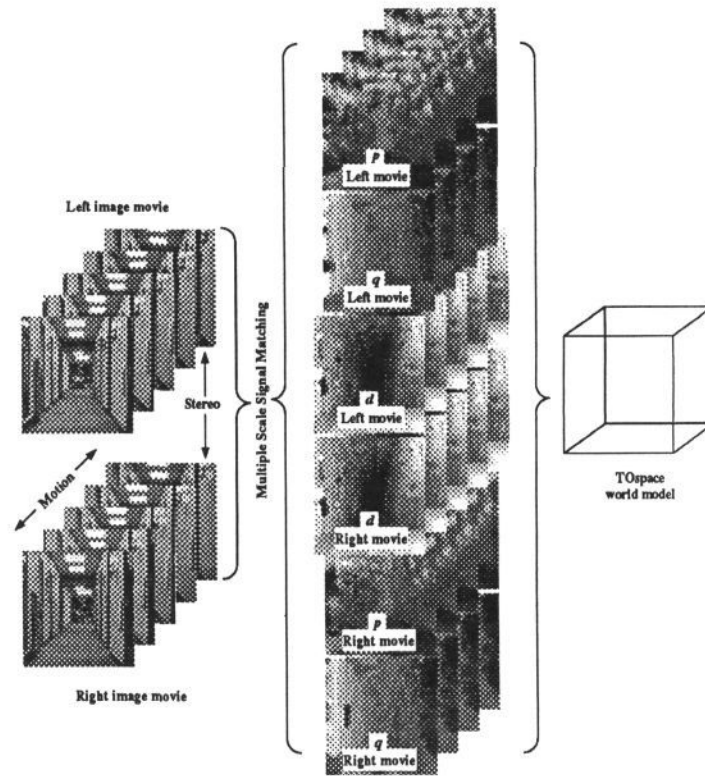


Figure 3: Overview of the robot vision system from the cameras to the world model.

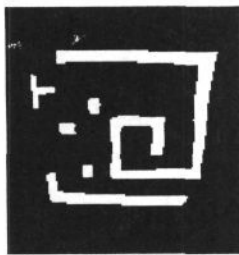
lowing each perception cycle, the model is used to generate a proximity description at each voxel. The task is achieved using the distance transform [2]. This simple operation labels voxels with gray-levels whose values correspond directly to the distance from surfaces. The distance transformed world model is termed *proximity space*.

When deciding what constitutes a good local plan, a spectrum of optimality exists. However, two common criteria for optimality are the path of shortest distance (an economical route) or the path of most avoidance (a cautious route).

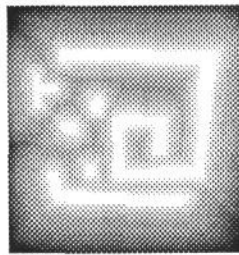
2.1 Generating an Economical Route

The economical route generation involves a four stage computation:

1. Generate proximity space with respect to obstacles.
2. Threshold to eliminate robot geometry (configuration space).
3. Generate proximity space with respect to the robot start position.



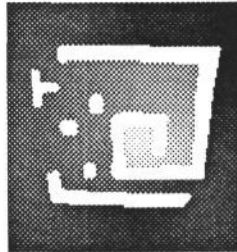
1. Original Image



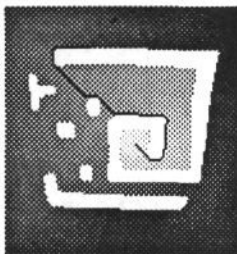
2. Proximity Space wrt Objects



3. Enlarged Objects



4. Proximity Space wrt Robot



5. Generated Plan from Start to Goal

Figure 4: Stages in Generating Shortest Path from Location (70, 70) to (7, 60)

4. Generate path of steepest decent from goal to start.

The first two stages have the effect of enlarging obstacle size whilst reducing robot size accordingly (down to unit voxel). Stage 3 uses the distance transform to form a proximity space model where non-feature points have assumed a value denoting their distance from the robots start (current) position. Feature points are given a write protection, resulting in a *refraction* around the enlarged obstacles. Stage 4 extracts the route from start to goal. Working back from the goal location, an 8-neighbour search is performed. The minimum valued voxel (ie. that which is closest to the robots start position) is selected as the next path voxel. This process is iterated until the start voxel is reached.

Figure 4 illustrates the process on a 2D scene.

2.2 Generating a Cautious Route

The cautious route generation involves a five stage computation:

1. Generate proximity space with respect to obstacles.
2. Extract points of local maxima.
3. Generate routes from start and goal location to closest point of local maxima.
4. Extract connectivity information from result.
5. Use a search to generate a plan through local maxima.

Stage 1 is exactly as described in the previous case. Stage 2 is performed by scanning the image and checking orthogonal/diagonal voxel pairs. Points of local maxima occur where a point is equidistant from two or more feature points on different obstacles. These points are extracted by the following simple classification:

local max iff:
 $(\text{current-voxel} > \text{voxel1}) \text{ and } (\text{current-voxel} > \text{voxel2})$
 or
 $(\text{current-voxel} \geq \text{voxel1}) \text{ and } (\text{current-voxel} > \text{voxel2})$

This results in the generation of line segments passing between obstacles. However, points of local maxima occur due to the non-convex nature of an obstacle. This may be due to rotation, noise, or the objects themselves (eg. having a corner). A post-processing step is the removal of such line segments, since traversal along one of these would result in a collision with an obstacle.

Since it is unlikely the start and goal locations will be a point of local maxima, stage 3 generates a path between the start (or goal) and the closest point line segment. This is achieved through exactly the same algorithm as generating the economical path.

Stage 4 extracts topological information from the result. This then forms input to a planner, which performs a search to generate those line segments to be traversed.

Figure 5 illustrates the process on a 2D scene.

It is worth noting that the generation of local route plans does not require absolute linear scal-

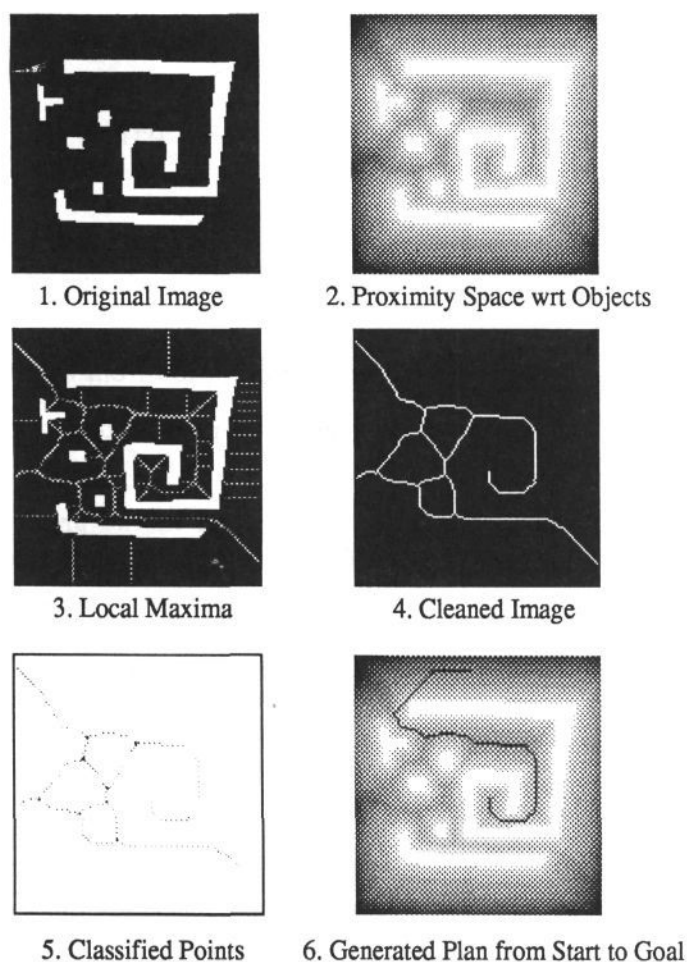


Figure 5: Stages in generating clearest path from location (70, 70) to (7, 60)

ing of the world model. For most practical situations, a non-linear scaling (as found in the TOSpace model) is sufficient in order to navigate to known locations. Figure 6 shows a horizontal slice through the TOSpace model of the corridor as shown in figure 3. The side walls of the corridor get closer (disparity reduces) away from the viewpoint. The slice contains an overlay of two local plans generated using the model.

Acknowledgements: this work has been supported by the Turing Institute, Hunting Engineering Ltd and The National Engineering Laboratory.

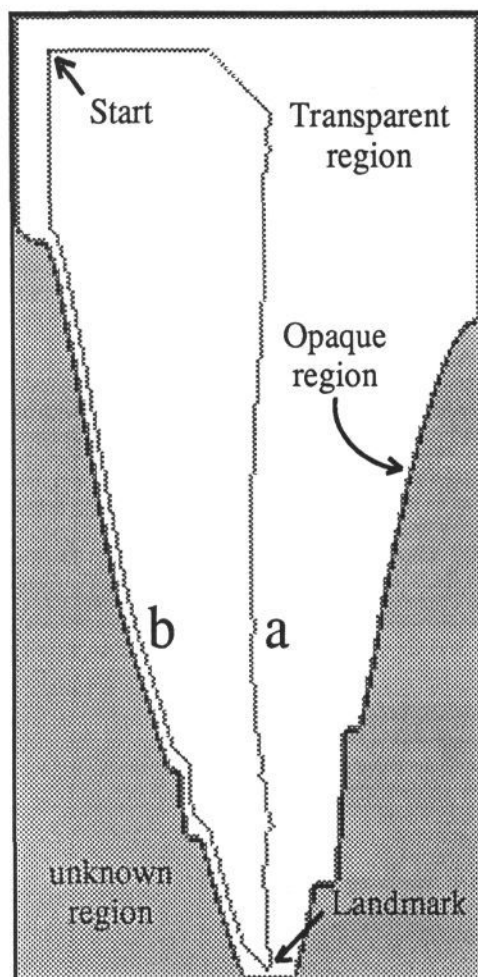


Figure 6: A horizontal slice through a TOSpace world model taken in the corridor with overlays for a cautious (a) and an economical (b) local route plan.

References

- [1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283-310, 1989.
- [2] G. Borgefors. Distance transforms in digital images. *Computer Vision Graphics and Image Processing*, 34(3):344-371, 1986.
- [3] Z. P. Jin and P. H. Mowforth. *A discrete approach to signal matching*. Research memo TIRM-89-036, The Turing Institute, Glasgow, Scotland, January 1989.