Using CAD Drawings for Robot Navigation[1]

Aniket Murarka and Benjamin Kuipers

Artificial Intelligence Lab, Computer Science Department, University of Texas at Austin, TX-78712, USA

Abstract

Exploration and navigation of an environment by a robot usually involves the steps of mapping, localization and path planning. Here we look at the problem of navigation in an environment about which the robot has some *a priori* information available, namely in the form of an architectural CAD drawing. The CAD drawing is utilized to obtain: (i) a topological map of the environment which is used for large scale path planning between regions in the environment and (ii) a skeleton of each region in the environment for path planning within regions. We then propose that a hierarchy of representations consisting of the topological map, skeleton and a reactive hazard-avoidance control system can be used effectively for navigation and exploration by a robot.

Keywords

CAD drawings, architectural drawings, mapping, path planning, navigation, mobile robot, topological map, Voronoi diagram, medial axis transform, layered control.

1. Introduction

Autonomous exploration and navigation of an environment by a mobile robot is a difficult task involving the steps of mapping, localization and path planning. Most research in these areas deals with unknown environments. However, in many situations such as institutional offices and campus environments, an architectural CAD description is readily available. Hence, our decision to focus on the problem of robot navigation in such environments.

The CAD drawing serves as an *a priori* global map of the environment. This map may not be an accurate representation of the environment but nevertheless it makes the process of exploration, particularly of large-scale space, simpler for the robot.

The problem dealt with in this paper, therefore, is of how to use the information in such CAD maps effectively for exploration and navigation. This will involve the following main steps:

- Obtaining the necessary information from the CAD map.
- Representing this information in forms suitable for planning.
- And finally planning based on the representations.

The architectural CAD drawings we use are 2-dimensional and use line segments and other geometrical entities to represent the environment.

1.1. Related Work

In this section we look at related work done on navigation in known environments and mainly consider approaches that use line segment based representations of the world.

Most methods use 2-D line segment based maps of the environment ([3], [12]). In [12], the map contains the locations of obstacles and landmarks that are used by the robot for localization. To perform path planning the map is converted into an occupancy grid map. Crowley [3] divides the free space in the map into a network of convex regions to perform global path planning. This representation is more compact than the occupancy grid used in [12].

Kosaka and Kak [7] employ a 3-D geometric model of the world. This model is used to build a *skeleton* of the 2-D floor plan of the environment which is used for path planning. Fennema *et. al* [4] use solid modeling techniques to construct a detailed 3-D model of the world. Path planning is then done using this model at several levels of abstraction.

Thrun *et al.* [13] assume that the environment is initially unknown and construct an occupancy grid map of the environment by exploration. A topological graph of disjoint regions is obtained from the grid and multiple path planners then determine a set of motion commands for the robot to follow.

Most of the approaches described above use a layered control architecture with each layer performing different tasks for navigation. Such control architectures used in mobile robots today usually have three layers for (i) deliberative computation, (ii) plan execution and (iii) reactive feedback control ([5]).

There are many more approaches to the problems of mapping and path planning and details on them can be found in [6], [9].

[1] This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by NSF grants IRI-9504138 and CDA 9617327, and by funding from Tivoli Corporation.

1.2. Our Approach

The architectural CAD drawings we use here represent the environment as collections of line segments, arcs and other geometric entities. However, in order to simplify and focus the problem we assume that the CAD map is made up of straight line segments and ignore other geometric entities. [2] Furthermore, we restrict the present research to indoor environments.

As stated earlier our attempt here is to use the metrical CAD map effectively for navigation by coming up with a better representation of the map and using appropriate techniques for path planning. Therefore our approach is as follows:

1. We first process the CAD drawing to extract geometrical descriptions of the rooms and doors present in it. This effectively divides up the CAD drawing into discrete regions (i.e. rooms) each of which can be dealt with independently. Section 2 describes this in detail.
2. The collection of rooms and doors is used obtain a topological map of the environment, with the rooms as its nodes and doors as its edges. The topological map thus provides for a compact representation of the environment and is used for large-scale path planning. Section 3 discusses this in greater detail.
3. The skeleton (also known as the medial axis transform) of each room in the CAD drawing is then constructed. The medial axis transform is very similar to the Voronoi diagram and is used for path planning within each room or region. This is discussed in section 4.

Once the topological map and skeleton of the environment have been constructed, the following hierarchy of representations is proposed for robot navigation and exploration:

- The topological map at the top level of the hierarchy for large-scale path planning.
- The room-specific skeleton at the next level to propose trajectories suitable to reach a destination within an empty room.
- A reactive hazard-avoidance control system (like that proposed by Brooks [2]) at the third level for motion within a room to enable the robot to reach its destination even if the room is far from empty.

This hierarchy thus provides for a layered control architecture very similar to the generic model mentioned in section 1.1. The sections that follow discuss in detail our approach to the problem.

[2] The algorithms we describe can be applied to other geometrical shapes by first approximating them with straight line segments.

2. Obtaining Geometrical Information from CAD Drawings

Indoor environments, such as university buildings or corporate offices, can be represented as a collection of rooms or corridors connected by doors. This information can be extracted from CAD drawings of such environments and is the first step of the method described in section 1.2. However since the drawing is not error free, it is first modified to remove errors and then processed to extract the doors and rooms. We begin with a description of the CAD drawings.

2.1 Architectural CAD drawings

The Architectural CAD drawings used were available as AutoCAD drawing interchange format (*dxf*) files. Figure 1 shows such a CAD drawing of the 4th floor of Taylor Hall, a building at the University of Texas at Austin.

AutoCAD *dxf* files represent objects (walls, doors, stairs etc) in the environment as a collection of geometric entities such as line segments, arcs, text etc. For example, walls are represented by line segments and doors by line segments and arcs (see figure 1). Note that in the CAD file rooms are not a separate object i.e. line segments representing a room are not associated with each other in any way.

All the geometric entities are further grouped into different layers usually depending on the object they represent. For example, all line segments and arcs representing doors belong to a single layer called *A-door* in the CAD file. Table 1 shows some layers and the associated objects and entities they contain.

To start processing, the CAD database is first parsed and all relevant information regarding geometric entities stored in data structures. Doors in the CAD drawing are also identified using the layer information described above.

Layer	Geometric Entities or Objects
A-door	Line segments and arcs representing doors.
A-wall-env	Line segments representing outer walls of the building.
S-colm	Line segments representing vertical columns.

Table 1: Separation of objects/entities into layers.

2.2. Removing Errors from the CAD Drawing

Before the required information from the CAD drawing can be extracted, the drawing is first pre-processed to remove errors present in it. There are two types of errors present in the CAD drawings – numerical errors and drawing errors. Numerical errors have to be dealt with throughout – they can never be removed. Errors that arise due to

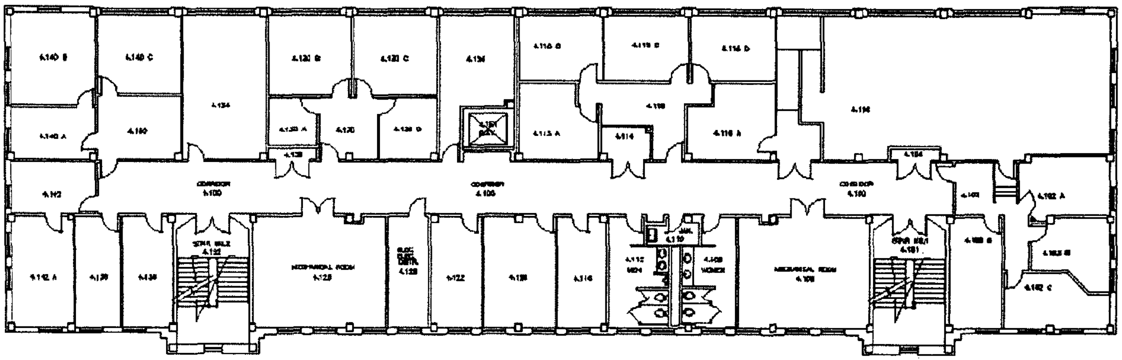


Figure 1: CAD drawing of Taylor Hall 4th floor.

inaccuracies or mistakes made while making the drawings are termed drawing errors. The CAD file is therefore pre-processed so that most drawing errors are removed before any other operation is performed on the file. Details on the types of drawing errors and how they are removed can be found in [11].

2.3. Extracting Rooms from the CAD Drawing

Since rooms are not represented as separate objects in the CAD drawing so each room has to be extracted from it. A set of names (text) is associated with each room in the CAD drawing (see figure 1). In most cases the room's name is within its boundaries and gives us a location within the room. Using this location we can determine the line segments that represent the walls and doors that make up the room associated with that name. The process is as follows.

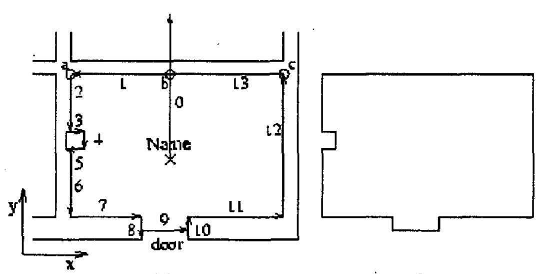
Algorithm

Figure 2a gives an idea of the algorithm used to extract rooms. It shows a typical room in the CAD drawing. A name is chosen and an infinite ray, called *ray 0* (and marked 0 in figure 2a), is drawn from the name's location in the direction of the positive y-axis. Its first point of intersection is with a line segment representing *wall ca* of the room. Part of this wall, from *b* to *a*, then becomes a new ray and is marked 1. We say that *ray 1* is *derived* from line segment *ca*.

Note that we make the left half of wall *ca*, with respect to *ray 0*, the new ray i.e. we make *ba* the new ray. We could as well have chosen the right half (i.e. *bc*) as the new ray. This would have given us a ray in the opposite direction. So we decide beforehand a direction to follow with respect to the current ray i.e. we choose to always go either left or right with respect to the current ray. This has no effect other than changing the order in which the walls and doors of a room are extracted.

The new ray, *ray 1*, so obtained intersects just one line segment (*line 2*) and so we make *line 2* $\rightarrow$ *ray 2* and make *ray 1* $\rightarrow$ *wall 1*. *Wall 1* is from *c* to *a* and is part of *wall ca* of the room. We proceed similarly in the direction of the arrows, as shown, until we reach *ray 13* (*c* to *a*). Since *ray 13* overlaps or touches *ray 1* we assume the room has been completely determined. The algorithm stops execution and part of *ray 13* becomes *wall 13*. This gives *walls 1 to 13* (including *door 9*) of the room and completes the extraction process. Figure 2b shows the extracted room.

A couple of things to be noticed about the extraction process: (i) The initial ray, *ray 0*, is not part of any wall or door of the room. (ii) The whole of *ray 5* does not make up *wall 5*. Only the right half of *ray 5* gives *wall 5*. A similar case occurs with *ray's 8* and *13*. (iii) The rooms do not have to be convex polygons. The method works for non-convex rooms.



(a) (b)

Figure 2: Extracting a room.

Figure 3 shows an exploded view of the rooms extracted from the CAD drawing shown in figure 1. All the rooms were successfully extracted except for one, due to drawing errors that could not be removed. Moreover some rooms are only partially extracted due to the nature of the CAD file – this is discussed in detail in the next section.

There are quite a few implementation details associated with the process presented here but due to lack of space they are not discussed. They can be found in [11].

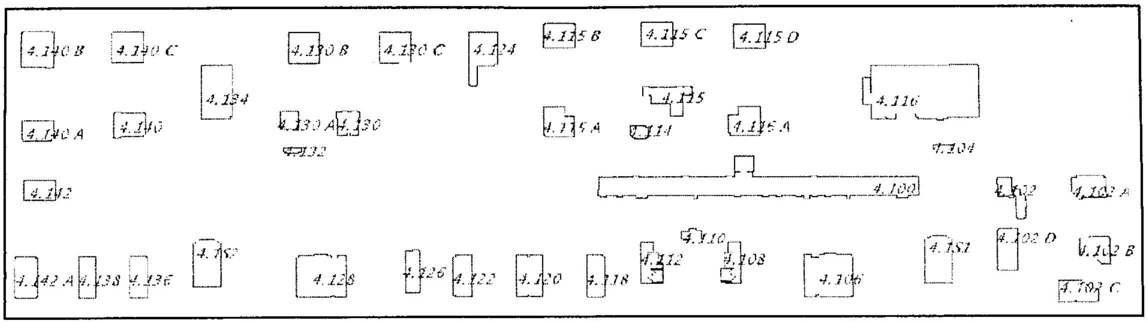


Figure 3: Rooms extracted from the CAD drawing.

3. The Topological Map and Path Planning

The topological map describes the environment, obtained from the CAD drawing, as a graph of rooms. The rooms form the nodes of this graph, with rooms with common doors being adjacent. Thus, the doors act as the edges between the rooms. Figure 4 shows the topological map obtained for the CAD drawing shown in figures 1 and 3.

As mentioned earlier all rooms are not extracted completely, the reasons being as follows: During the extraction of room R2, the presence of staircase S1 prevents extraction of that region of the room which has staircase S1 (see figure 4). This occurs because in the CAD file the staircase is in the same layer as the walls and so the extraction algorithm has no way of differentiating between line segments representing these stairs and those representing walls. It therefore treats the staircase line segments as walls. As a result door d1 (common to rooms R1 and R2) and doors d34 (common to R6 and R2) are not extracted along with R2 and hence not considered to be part of room R2. Thus room R2 cannot be connected to room R1 or R6. Exactly the same case occurs with room R3 because of staircase S2 due to which R3 is not considered adjacent to R2. However the staircases in rooms R4 and R5 are in different layers and so they can be ignored.

This problem of some geometric entities, representing different objects, being placed in the same layer occurs for other objects as well in the drawing such as ramps, doors etc. There are other problems also which are discussed in [11].

3.1 Path Planning

The topological map is used for planning paths from one room to another. The path planner finds a path between the robot's current location in a room to the approximate centroid of the destination room. The A* algorithm is used for determining the routes. In order to simplify implementation and increase the speed of the program, the search tree for the A* algorithm uses doors as states and rooms as edges. The heuristic used is Euclidean distance between two doors. However since the rooms are not always convex the path determined may not be optimal (refer [11]).

Thus the topological map can be used for large-scale path planning such as between rooms in a building or in different buildings. To find paths within a room its medial axis or skeleton is constructed.

4. The Medial Axis Transform

The medial axis transform of a polygon, also called its skeleton, is formally defined as follows [10]: Given an object represented by a polygon G, its medial axis is the set of points internal to G such that

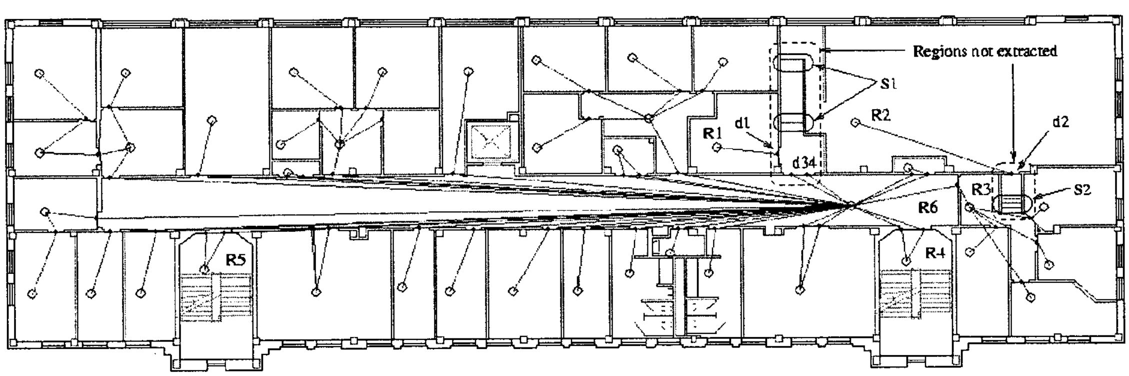


Figure 4: Topological map of the CAD drawing.

there are at least two points on the object's boundary that are equidistant from q and are closest to q. The medial axis of a polygon is very closely related to the Voronoi diagram and is, in fact, totally contained in the set of Voronoi edges of the polygon as shown in figure 5.

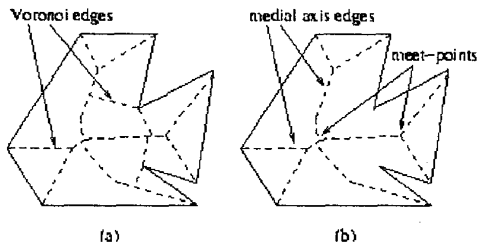


Figure 5: (a) Voronoi diagram and (b) skeleton.

There exist a number of methods for constructing the medial axis of a polygon of which many are based on optimal algorithms for constructing Voronoi diagrams (refer [1] for an extensive survey). However, the method we implement here is not optimal but is based somewhat on the techniques that would be used by a real robot to construct the skeleton or Voronoi diagram of a real environment.

3.1 Constructing the Skeleton

The medial axis of the CAD drawing is constructed by a simulated Voronoi robot as it explores the environment. The robot is circular, with center at (x,y)

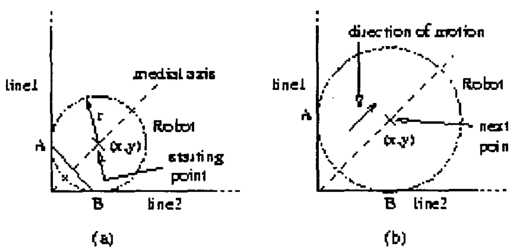


Figure 6: The Voronoi robot.

and radius r and moves by changing x, y and r to stay in contact with walls or obstacles (see figure 6a). The robot moves in steps and in a given direction at a time. The step size is chosen before exploration starts and determines how close the constructed skeleton will be to a perfect skeleton. The Voronoi robot treats each room like a polygon and constructs the skeleton of each room separately. The following are the steps followed by the algorithm:

1. The robot starts the exploration from a point on the medial axis of the room. This point is equidistant from exactly two line segments in the room and is determined beforehand. Figure 6a shows the Voronoi robot starting at a point on the medial axis equidistant from the two line segments (*line1* and *line2*).

2. With the two points of contact on the lines (marked A and B in figure 6a) the robot moves *approximately* [3] along the direction of the perpendicular bisector of the chord formed by them (dashed line which is also the medial axis), shrinking or growing r to maintain contact, as shown in figure 6. Figures 6 and 7 show that as long as there are 2 points of contact the robot remains on the same edge of the medial axis (marked 1, 2 and 3 in figure 7).

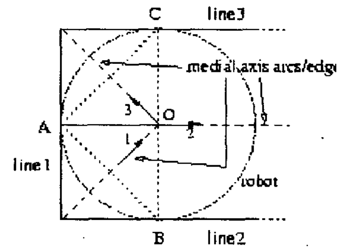


Figure 7: Reaching a meet-point

3. If more than two points of contact occur (e.g. A, B and C in figure 7) the robot defines a meet-point at its center (point O in figure 7) and explores further by moving in the direction of the perpendicular bisectors of the two or more new chords defined by the points of contact (as shown by the arrows in figure 7). Since all the contact points touch the circle they are also equidistant from its center. The robot continues to explore the environment keeping track of all meet-points to make sure it explores the whole environment.

The trace of the robot's center (x, y) is then the medial axis. Figure 8 shows the skeleton obtained for the entire CAD drawing with all doors open. The skeleton thus obtained can be used for suggesting routes to the robot within a room. One thing to note is that due to a finite step size the robot sometimes re-detects the meet-point it detected a step earlier and so while searching for new meet-points the robot should restrict its field of view. [11] provides more information on this and other implementation details.

A problem that arises is that the medial axis of a real environment with obstacles in it can be quite different from the one constructed using the CAD drawing. This can be overcome by ignoring unknown obstacles in the real world. One way of doing this is by using the ceiling of a room to construct the medial axis in the real world. Another way is using vision sensors as done by Kosaka and Kak [7]. Further work will focus on a better solution to the problem.

---

[3] Due to numerical errors, moving along the perpendicular bisector causes the robot to deviate from the axis. One solution is moving towards the *center* of the chord joining the equidistant points (e.g. chord AB in figure 7) or its reflection about the robot's center - depending on the robot's direction of motion.

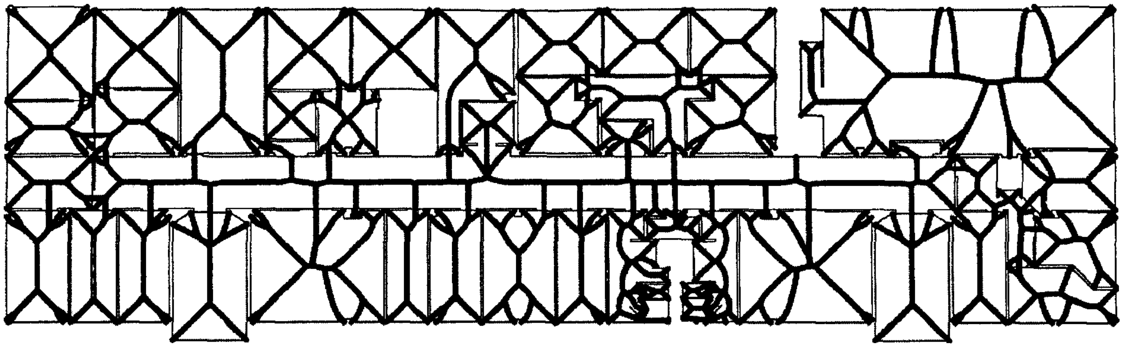


Figure 8: Skeleton of the CAD drawing with doors open.

5. Conclusion

The topological map along with the skeleton allows the robot to plan a complete path to its destination. This combined with a reactive control system, as proposed in section 1.2, can help the robot find a path to its destination even in the presence of unknown obstacles in the environment. The topological map and skeleton also provide for a compact representation of the environment as compared to occupancy grids or 3-D models as discussed in section 1.1. Also the use of multiple planners simplifies the path planning problem.

The use of readily available CAD drawings enables the robot to easily explore large-scale space. In this respect, another aspect being considered is using this work to complement the Spatial Semantic Hierarchy (SSH) [8] which is designed to model the process by which a robot learns a cognitive map from its experience during travel. The use of a graphical map, like a CAD drawing, with the SSH makes it possible to do things that are very difficult to do with local observations gathered during travel.

Future work will also consist of implementing the hierarchy proposed here on a physical robot and evaluating its effectiveness. Since CAD drawings of outdoor areas are available as well we will look at extending the present work to such environments.

References

[1] F. Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. ACM Computing Surveys, v23, pp. 345-405, 1991.

[2] R.A. Brooks. A robust layered control system for a mobile robot. IEEE Transactions on Robotics and Automation, v2, n1, pp. 14-23, 1986.

[3] J.L. Crowley. Navigation for an intelligent mobile robot. IEEE Journal of Robotics and Automation, v1, n1, pp. 31-41, Mar. 1985.

[4] C. Fennema, A. Hanson, E. Riseman, J.R. Beveridge and R. Kumar. Model-directed mobile robot navigation. IEEE Transactions on Systems, Man and Cybernetics, v20, n6, Nov.-Dec. 1990.

[5] E. Gat. Three-layer architectures. In Artificial Intelligence and Mobile Robots: Case Studies of Successful Robotic Systems, D. Kortenkamp, R.P. Bonasso and R. Murphy, Eds., AAAI Press, CA, MIT Press, MA, pp. 195-210, 1998.

[6] D. Kortenkamp, R.P. Bonasso and R. Murphy (Eds.). Artificial intelligence and mobile robots: Case studies of successful robotic systems. AAAI Press, Menlo Park, CA, MIT Press, Cambridge, MA, 1998.

[7] A. Kosaka and A.C. Kak. Fast vision-guided mobile robot navigation using model based reasoning and prediction of uncertainties. In CVGIP: Image Understanding, v56, n3, pp. 271-329, Nov. 1992.

[8] B. J. Kuipers. The Spatial Semantic Hierarchy. Artificial Intelligence, 119, pp. 191-233, 2000.

[9] J.C. Latombe. Robot motion planning. Kluwer Academic Publishers, Boston, 1991.

[10] D.T. Lee. Medial axis transformation of a planar shape. IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-4, n4, pp. 363-369, July 1982.

[11] A. Murarka. Using CAD drawings for robot navigation. Master's thesis, Mech. Engg. Dept., Univ. of Texas at Austin, USA.

[12] S.B. Nickerson, P. Jasiobedzki, D. Wilkes, M. Jenkin, E. Milios, J. Tsotsos, A. Jepson and O. N. Bains. The ARK project: Autonomous mobile robots for known industrial environments. Robotics and Autonomous Systems, v25, pp. 83-104, 1998.

[13] S. Thrun, A. Bucken, W. Burgard, D. Fox, T. Frohlinghaus, D. Hennig, T. Hofmann, M. Krell and T. Schmidt. Map learning and high-speed navigation in RHINO. In Artificial Intelligence and Mobile Robots: Case Studies of Successful Robotic Systems, D. Kortenkamp, R.P. Bonasso and R. Murphy, Eds., AAAI Press, CA, MIT Press, MA, pp. 21-52, 1998.