Proceedings of the 1999 IEEE
International Conference on Robotics & Automation
Detroit, Michigan • May 1999

# Perception-Based Motion Planning for Indoor Exploration

Peter Leven, Seth Hutchinson
p.leven@computer.org, seth@uiuc.edu
University of Illinois at Urbana-Champaign
Urbana, IL, USA

Darius Burschka*, Georg Färber
{burschka|faerber}@lpr.ei.tum.de
Technische Universität München
Munich, Germany

## Abstract

*This paper proposes an approach for motion planning in indoor environments based on incomplete and uncertain information from a line-based binocular stereo system. The primary goal of the planning process is to plan an optimal path through an unknown or partially known environment, depending on the information gained from exploration and the current mission goal. This paper presents an adaptable motion planner that supports sensor-based map construction, object recognition and navigation in an unknown environment while carrying out a mission. Also presented are some preliminary experimental results that demonstrate the utility of the approach.*

## 1 Introduction

A mobile robot needs knowledge about the environment to plan its actions and to fulfill the mission goals, which are often specified relative to the known obstacles in the local area. The environmental models can be known beforehand, but gradual changes in the environment deteriorate the usability of those models for sensor data interpretation. A better approach is to explore the environment and maintain the geometric models by using the sensor system.

The idea for the planning system is to consider the global mission goal and hints from the sensor systems and interpretation modules to achieve the best information gain in a given period of time. Time constraints are also considered to optimize the planned path.

This paper describes an adaptable, two-part planning system that supports sensor-based map construction, object recognition, and navigation in unknown environments while achieving its missions. The upper level part of the planner, the strategist, plans paths for the robot using a topological graph of the environment. The purpose of this graph is to take advantage of the structure available in indoor environments to simplify the global planning task. The lower level part of the planner, the navigator, handles the details of moving the robot amid obstacles.

The remainder of this paper is organized as follows. First, an overview of the system in which the planner is integrated is given in Section 2. This is followed by a description of the planning approach itself in Section 3. Some experimental results are described in Section 4, and some concluding remarks are given in Section 5.

## 2 System Overview

A motion planner does not work by itself; it requires additional components to handle sensing of the environment, management of the information obtained from the environment, and control of the motions of the robot. The architecture of the overall system that includes the planner is shown in Figure 1, where the arrows in the figure represent the primary direction of information flow (commands and data) among the components. In this figure, the parts that correspond to the motion planner are those labeled "strategist" and "navigator." The role of the other parts of the system are discussed below.

**Sensors** The robot has one or more sensors with which it gathers information about its environment. Each physical sensor has three types of modules associated with it: a control module, a calibration module, and one or more processing modules. The control module is the interface to the hardware of the sensor, and it maintains the state information of the sensor. For example, for a stereo camera system, the control

*Darius Burschka is now with the Department of Computer Science at Yale University.

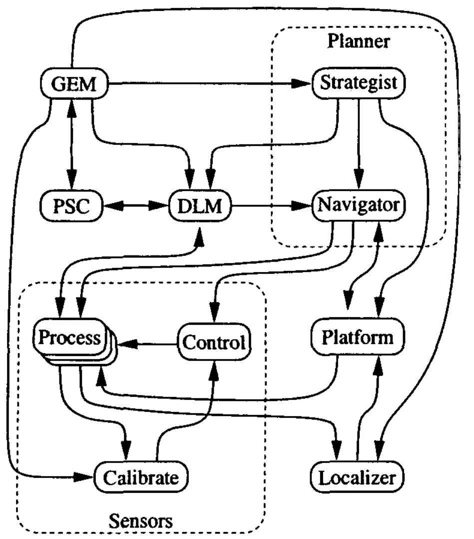


Figure 1: The flow of information in the system.

module controls the tilt, pan, and vergence angles of the cameras. The calibration module uses sensor information about objects in the environment, model information about those objects, and information about the current position of the robot to recalibrate the sensor parameters and the sensor hardware state.

Whereas a sensor, in general, requires only one module for control and another for calibration, it can have many different processing modules associated with it. For example, the cameras of a stereo camera system can be used for both three-dimensional line segment extraction and optical flow computation. The different processing modules can also be active at different times, to maximize the precision of their measurements, for example. For the stereo camera system example, three-dimensional line extraction is most accurate when the cameras are not in motion because the vibration of the cameras is reduced. On the other hand, optical flow processing can obtain information about a static scene only when the cameras are in motion.

Each processing module uses calibration and state information about the sensor as well as state information about the robot to determine the location of features it finds in its data. The processing modules can also use information stored in a dynamic local map (described below) from prior processing operations to simplify or improve the processing of new data from the sensors. For the stereo camera example, this includes limiting the number of matching operations needed to find a segment in both camera images.

**DLM** Raw sensor information can overwhelm a motion planner and prevent it from finding a path because of the amount of noise and uncertainty that the raw sensor information contains. The dynamic local map (DLM) serves as a feature filtering module that accepts features from the sensor processing modules as well as from interpretive modules that generate hypotheses for the sources of the features. The feature filtering has three purposes. (1) It is used to reduce the effects of noise in the sensor system. Sensor noise can come from a variety of sources, such as vibrations of the robot and sensors, calibration errors, and shadows and other lighting effects. (2) It provides verification of hypotheses about objects in the environment. (3) It serves as a basis for fusing information from the different sensors. As a result of filtering, each feature is assigned a confidence value that reflects the likelihood that the feature comes from an object in the environment. More detail about the DLM can be found in [1].

**PSC** To reduce the amount of information that is needed to describe the obstacles in the environment, it is helpful to recognize the objects from which the features detected by the sensor system originate. Object recognition is especially important when the task of the robot is specified in terms of the objects in the environment. The Predictive Spatial Completion (PSC) module uses sensor information stored in the DLM to construct geometric models of objects in the environment. To accomplish this, the PSC uses model knowledge about the types of objects expected in the environment and sensor information to generate hypothetical completions of the sensor features that match object hypotheses. The hypothetical completions are then stored in the DLM. As more sensor features are received by the DLM, the hypothetical features either gain support by being matched with new sensor features, or they are rejected after a predefined interval. The PSC is described in more detail in [2].

**GEM** A database is needed to store information that is gained from the environment in order for that information to be used again. In addition, a database is needed for the templates that are used for object recognition. The Geometric Environmental Model (GEM) stores a hierarchical geometric representation of the environment. This model includes a graph structure of the connected regions of the environment and the objects found in those regions. The model for articulated objects is also hierarchical, with the root of the graph describing the geometry of the fixed portion of the object and edges from the root describing joints attached to the fixed portion of the object. Information useful for the different sensors is also stored with the representation of the object. The model is described in more detail in [3].

**Localizer** The accuracy of the information derived from the sensors depends in part on how well the robot knows where it is and how it has moved. The localization module uses data about the objects in the environment and the geometric information from the environment model about these objects to produce an estimate of the position of the robot. This estimate is provided to the platform module to be incorporated with the dead reckoning information to more accurately localize the robot. Techniques used for localization are described in more detail in [4] for vision-based localization and in [5] for laser range finder-based localization.

**Platform** To make the planning system more independent of the type of mobile robot that is used, a robot-independent interface is used. The platform module is the interface to the drive hardware of the robot and maintains the position and orientation state information about the robot. The position and orientation information is derived primarily from dead reckoning, with the localizing module supplying corrections. Since the data processing by the localization module requires some time to complete, the position estimate produced may correspond to a prior position of the robot. A Kalman filter-type approach is used to integrate the position estimates from the localization module and the position estimate from the dead reckoning information.

## 3 Motion planning

To find a solution to a problem, it is often easier to split the problem into two or more parts and construct the overall solution by combining the solutions to the subproblems. In motion planning, this is commonly achieved by splitting the motion planning problem into two parts: one that plans on an abstraction of the environment and the other that handles the details of moving the robot amidst obstacles. An early example of this was reported in [6], in which the abstraction used first-order predicate calculus to plan the actions of the robot.

Roadmap motion planning methods are another variation on the multi-part technique for solving motion planning problems. In this case, the steps of the solution involve connecting the current and goal configurations of the robot to the network and finding a path along the network connecting the two connection points [7]. Randomized roadmap methods are similar, except that they involve the use of a local planner during the construction of the roadmap and during the planning phase to connect the initial configuration and the goal configuration to the network [8].

The motion planner described here is a two level planner that uses a roadmap defined by the topology of the environment. The upper level of the planner plans paths in the roadmap and uses the lower level planner to connect the nodes at each stage of the plan. This use of the local planner to connect nodes in the roadmap is also performed by the randomized roadmap motion planners, although the randomized planners connect the nodes only once during the pre-processing phase.

The parts of the motion planner are called the strategist, which plans paths in the roadmap, and the navigator, which is the local planner. The strategist is also responsible for accomplishing the current mission of the robot, and it coordinates the actions of the navigator to achieve this. The navigator uses sensor information to guide the actions of the robot to satisfy the goals given to it by the strategist. These two parts will be described in mode detail in the following sections.

### 3.1 Strategist

The strategist is the upper level planner. To accomplish a mission, it plans paths through a global topological graph of the environment and coordinates the actions of the navigator to follow the path in the graph. The graph structure used by the strategist for planning and the actions performed by the strategist during path execution are the subject of this section.

**Graph structure** The topological graph used by the strategist is a set of nodes and edges where each node in the graph corresponds to a room in the environment and each edge corresponds to a doorway connecting a pair of rooms. Most edges in the graph are bidirectional, which means that the doorway can be used in both directions. Unidirectional edges are used for doorways that can be used in only one direction, such as the "in" and "out" doors associated with cafeterias and the like.

Each node is associated with its own local coordinate system that is used for navigation within that room. The location of each doorway in the local coordinate system is also stored. In the case that more than one doorway exists between a pair of rooms, an edge is stored in the graph for each doorway.

An example of a topological map for a portion of an office environment is shown in Figure 2. In the figure, the labels O1 to O8 correspond to offices, L1 and L2 are laboratories across from the offices, C1 is a conference room, P1 is a printer room, and H1 is the hallway connecting them. The dashed lines Figure 2.b correspond to connections outside the region shown

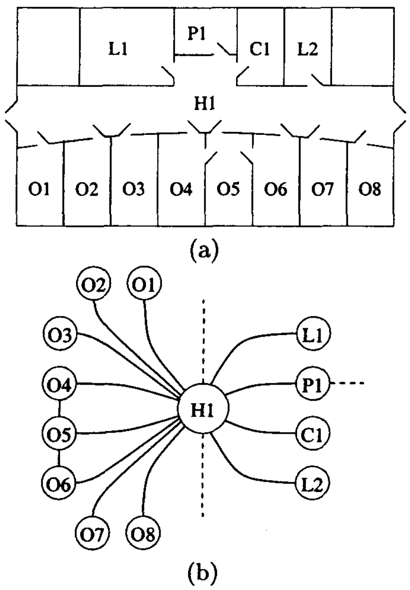


Figure 2: An example office environment and its corresponding graphs.

in Figure 2.a. The dashed line connected to node P1 represents a staircase to another floor that is present in the room.

The topological graph for an environment can be constructed in one of two ways. One method is that the robot itself builds the entire graph as it explores a previously unknown environment. The second method is that the plans for the building are used to build the graph. The latter case will also involve an exploratory phase in which the robot determines the local coordinate system to be used in each room as well as the positions of the doorways in the room with respect to that local coordinate system. In both cases, the robot will acquire a model of the objects in each room as the robot explores the environment.

**Path execution** To execute a plan, the strategist performs the following actions. For each node in the plan, the strategist uses the navigator to plan a path from the current location of the robot to the next doorway in the plan. Upon reaching the doorway, the state of the DLM and the navigator is reset, and the position and orientation of the robot are set to correspond to the local coordinate system for the next room. The DLM is then loaded with the known features of the new environment, and the navigator uses these features for its initial plan to reach the next goal provided by the strategist. This cycle repeats until the robot finishes its mission.

The strategist controls the behavior of the navigator within each room. The behavior can be as simple as reaching the goal in the shortest amount of time, or it can be dependent on the amount of prior knowledge of the room. For example, if the robot enters a room about which little is known, the strategist may instruct the navigator to explore the room to find out more about what is there. This may also be used if the robot has been away from a room for a long time. In this case, the robot looks for things that may have changed since the last time it was there.

The actions of the strategist are most complicated when the robot is brought to an environment about which nothing is known. This is a non-trivial problem that has been the subject of much prior research. One approach to this problem is described in [9], where a robot equipped with a vision-based recognition system carries out a systematic exploration of its environment using the landmarks it finds along the way as reference points.

At a high level, the actions of the strategist in an unknown environment can be described as follows. First, the strategist uses the navigator to explore the current room, using the robot's initial position as the origin of the coordinate system for the room. During the exploration of the room, doors and their coordinates are identified and are inserted into a topological graph that uses the current room as the root. Depending on the requirements of the situation, the robot may linger longer in the room to build models of the objects in the room, or the robot may move to a different room to continue constructing the graph of the environment. During the initial exploration of the room, the sensor calibration modules are disabled until models of objects in the room are sufficiently stable to permit sensor calibration.

### 3.2 Navigator

The navigator is the lower level planner for the system. It is concerned with planning paths within the current room, using the goals given to it by the strategist. It also coordinates the actions of the sensor system during the execution of the plan to maximize the amount of useful information that can be gathered. The navigator has two modes of operation. The first mode is concerned only with reaching the current goal. The second mode permits the robot to explore the environment. In both situations, the robot is given a goal location to be reached and the amount of time available to reach it.

The operation of the navigator can be described as follows. At each location at which the robot is stopped, the navigator performs a sensor scan of the region around the robot. The new features found in this scan are then incorporated into the obstacle set for the robot, and a new path is planned to the current goal. During execution of the path, the stereo system is used as a visual bumper to detect unexpected obstacles in front of the robot. If the robot successfully reaches the goal, the cycle repeats again until no more goals remain. If the robot encounters an obstacle while in motion, the robot backs up a pre-specified amount along the path it was following, and performs another sensor scan of the environment at that location. A new path is then generated to take into account the new information that is found during the scan.

A sensor scan consists of the turning the robot in a full circle, stopping at predefined angles. At each stopping angle, the stereo vision system performs a scan, first with the cameras level to the floor, and second with the cameras pointing toward the floor. In addition, while the robot is in motion, the optical flow sensor is used to detect surfaces and moving objects. The number of stereo images taken at each level is inversely proportional the amount of prior information that is available about the environment. The angle by which the robot rotates and the number and direction of stereo images are configurable parameters that can be used to optimize the sensor scan to suit different tasks.

During exploration, the local planner plans paths for the robot to maximize the amount of knowledge that the robot acquires in the room. This involves moving the robot to regions that may provide more information to the sensor system. Three types of regions are defined: empty regions, where the robot has no information; uncertain regions, where the robot has some information, but the information is incomplete or imprecise; and cluster regions, where the robot sees a cluster of features that may be used for object recognition. The priority given to each type of region depends on the current task of the robot. For example, a higher priority on the unknown regions means that the entire room is explored faster. A higher priority on imprecise information results in the precision of the features being improved before the robot finishes exploring. Last, a higher priority on clusters of features focuses the robot on object recognition.

## 4 Experimental Results

The mobile robot used for experimentation, shown in Figure 3, consists of a TRC-Labmate platform onto which a set of three PC's, a stereo camera system, a two-dimensional panoramic laser range finder, and a radio ethernet have been mounted. The stereo camera system is the primary sensor of the robot and is used for both three-dimensional line segment extraction and optical flow measurement. The laser range finder is primarily used for localization, and the radio ethernet is for communication with other computers in the lab. RPC is used for communication between the different modules of the system.

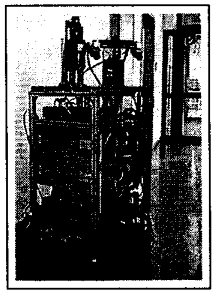

Figure 3: The robot *MARVIN*.

A preliminary version of the system has been tested in a laboratory environment. This setup tests the low-level planner in a relatively static environment with the stereo cameras pre-calibrated and no localization enabled. The robot was allowed to explore the room and it was able to successfully avoid obstacles. Figure 4 shows the planned path in a local area and the locations where the three-dimensional reconstruction was triggered. It shows the content of the navigation map with some tables, walls, and cabinets.

The navigator used in the preliminary version uses a two-dimensional occupancy grid similar to the one in [10] of the free space around the robot. This map is constructed by using the information from the sensors stored in the DLM. The primary features that the navigator uses are three-dimensional line segments extracted by the stereo camera system, with additional three-dimensional line segments coming from the panoramic laser scanner.

The three-dimensional line segments used to indicate the locations of the obstacles are those that meet

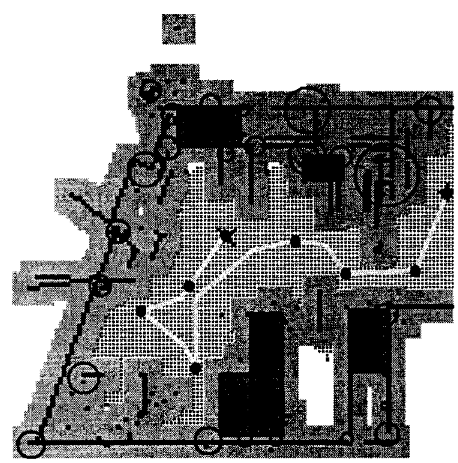

Figure 4: Planned path in an indoor environment.

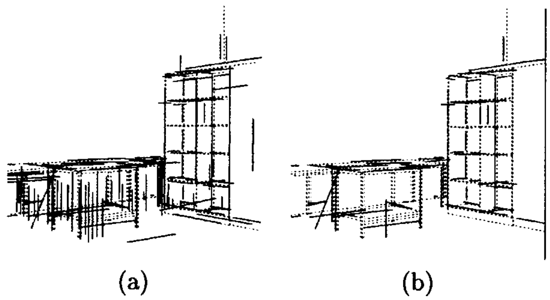

Figure 5: Content of the DLM for an arbitrary camera view: (a) all stored features and (b) features with a higher confidence.

a set of pre-determined criteria, each of which can be adjusted to suit the planning environment. The first criterion is that the line segments must have a likelihood above a preset threshold. This criterion reduces the number of sensor artifacts that are considered to be obstacles. Second, the precision of the endpoints of the line segments must fall within a preset range. The precision of the endpoints indicates the certainty of the sensor system that it found the endpoint of the line segment and the precision with which it could measure its location.

The third criterion that is used to determine whether a feature is part of an obstacle is the three-dimensional location of the segment. The space around the robot is divided into three regions: an upper region above the height of the robot, a middle region below the height of the robot and above a preset height close to the floor, and a lower region close to the floor. The selection of features to be used in path planning in the first two regions is simple: features in the upper region are ignored and features in the middle region are considered to be part of an obstacle. The lower region is a special case. The difficulty in this region is determining whether the feature is part of a pattern on the floor or whether it is part of an obstacle that extends above the floor. A simplistic approach is used in which if features can be found on the side of a floor feature opposite the robot, that feature is considered part of the floor and, therefore, is ignored.

An example for the information reconstructed from image information is shown in Figure 5, where the dashed lines show the correct position of the lines and the solid lines represent the reconstructed line segments.

## 5 Discussion

This paper presented an approach to motion planning for a mobile robot that operates in indoor environments. The motion planner consists of two parts: a strategist that plans paths on a graph representation of the environment and a navigator that plans paths in each node of the graph using information obtained from sensors. Both parts of the planner are adaptable to the requirements of the situation, favoring exploration more when less of the environment is known. Some experimental results were also described. In the following, some limitations of the approach as well as some future directions will be given.

While the planning approach described above is defined to be highly flexible, several limitations remain.

One limitation is outdoor environments. Although the planner was designed for working in indoor environments, a mobile robot would be more convenient to use if it can also navigate through a limited range of outdoor environments. For example, many institutions such as universities consist of more than one building where there are no enclosed connections between the buildings. The environment between the buildings may be sufficiently simple to allow the robot to operate there.

While the topological graph structure may be extended to the "concrete canyons" that typify the centers of many large cites, which can be considered as extended grid-like hallways connecting different rooms, it is not well suited for general outdoor environments with more irregular terrain. An approach that may be used instead is one based on a hypergraph as described in [11].

A second limitation is that the navigator depends on a sufficiently large number of features for localization to limit the dead reckoning errors of the platform. An example of such a region is a hallway with uniformly colored walls and no features such as paintings or other artwork on the walls. This type of environment can be further complicated by flush-mounted doors, such that, when the door is closed, the junctions between the door and the wall are hard to detect reliably. In such a corridor environment, the robot may miss its goal because it could not see it. An additional problem arises when the robot is exploring the environment for the first time to build a map of it. In this case, the location uncertainty accumulates over time, ultimately reaching a point at which the robot can no longer plan a path to the other end of the corridor because of the errors in the estimates of the positions of the walls. An approach that limits this effect is described in [12].

A third limitation is a warehouse-type environment. A warehouse consists of a large room with a regular array of shelving units. The tradeoff here is among the number of nodes to place in the topological graph, the difficulty in defining the junctions between nodes, and the amount of detail stored for each room in the environment. The tradeoff arises because the speed of the DLM to insert new features and verify old features is proportional to the number of features already stored. In addition, a large number of features increases the time required for path planning as well as the amount of space needed to find the path.

Some areas for future work include the following. One problem is the specification of a task for a robot to perform when it has no prior knowledge of its environment. An example situation is the case that a user gets a new robot with the system described above installed, turns the robot on, and, for example, instructs the robot to take the mail from the current room to the mail room.

Another area for future work is the addition of an arm on the robot. The addition of the arm creates a requirement that the sensor system be able to detect surfaces in order to be able to find the obstacles that the arm must avoid. Other issues include how one might control the arm to perform a task, and what kind of visual servoing is needed. Some work in this area is described in [13].

## References


[1] D. Burschka, *Videobasierte Umgebungsexploration am Beispiel eines binokularen Stereo-Kamerasystems.* PhD thesis, TU München, Fakultät für Elektro- und Informationstechnik, 1998.

[2] C. Eberst and G. Färber, "Coupling perception and interpretation by hypothetical features," in *IMACS Conf. on Comp. Eng. in Systems Appl. (CESA'98)*, vol. 4, pp. 53–59, Apr. 1998.

[3] A. Hauck and N. O. Stöffler, "A Hierarchical World Model with Sensor- and Task-Specific Features," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'96)*, pp. 1614–1621, 1996.

[4] D. Burschka and S. Blum, "Identification of 3D reference structures for video-based localization," in *Proc. 3rd Asian Conf. on Computer Vision (ACCV'98)*, vol. 1, pp. 128–135, 1998.

[5] T. Einsele, "Real-Time Self-Localization in Unknown Indoor Environments using a Panorama Laser Range Finder," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'97)*, (Grenoble, France), pp. 697–703, Sept. 1997.

[6] N. Nilsson, "A mobile automaton: An application of artificial intelligence techniques," in *Proc. 1st Intl. Joint Conf. on Artifical Intelligence*, (Washington, D.C.), pp. 509–520, 1969.

[7] J.-C. Latombe, *Robot Motion Planning.* Boston: Kluwer Academic Publishers, 1991.

[8] L. Kavraki, P. Švestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 566–580, Aug. 1996.

[9] C. Taylor and D. Kriegman, "Vision-based motion planning and exploration algorithms for mobile robots," in *Algorithmic Foundations of Robotics* (K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, eds.), pp. 69–83, Wellesley, MA: A K Peters, 1994.

[10] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer*, vol. 22, pp. 46–57, June 1989.

[11] R. Fayek and A. Wong, "Using hypergraph knowledge representation for natural terrain robot navigation and path planning," in *Proc. IEEE Conf. Robotics and Automation*, pp. 3625–3630, 1996.

[12] S. Thrun, D. Fox, and W. Burgard, "Probabilistic mapping of an environment by a mobile robot," in *Proc. IEEE Conf. Robotics and Automation*, pp. 1546–1551, 1998.

[13] T. Fink, A. Hauck, and G. Färber, "Towards an Anthropomorphic Robotical Hand-Eye Coordination," in *IMACS Conf. on Comp. Eng. in Systems Appl. (CESA'98)*, vol. 4, pp. 483–488, Apr. 1998.