CHAPTER 8
MAP-BASED POSITIONING

Map-based positioning, also known as “map matching,” is a technique in which the robot uses its sensors to create a map of its local environment. This local map is then compared to a global map previously stored in memory. If a match is found, then the robot can compute its actual position and orientation in the environment. The prestored map can be a CAD model of the environment, or it can be constructed from prior sensor data.

The basic procedure for map-based positioning is shown in Figure 8.1.

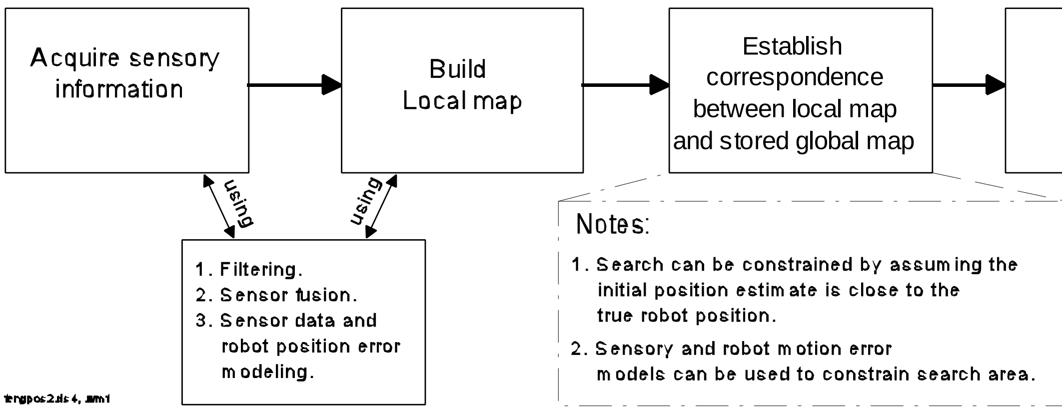


Figure 8.1: General procedure for map-based positioning.

The main advantages of map-based positioning are as follows.

- This method uses the naturally occurring structure of typical indoor environments to derive position information without modifying the environment.
- Map-based positioning can be used to generate an updated map of the environment. Environment maps are important for other mobile robot tasks, such as global path planning or the avoidance of “local minima traps” in some local obstacle avoidance methods.
- Map-based positioning allows a robot to learn a new environment and to improve positioning accuracy through exploration.

Disadvantages of map-based positioning are the specific requirements for satisfactory navigation. For example, map-based positioning requires that:

- there be enough stationary, easily distinguishable features that can be used for matching,
- the sensor map be accurate enough (depending on the tasks) to be useful,
- a significant amount of sensing and processing power be available.

One should note that currently most work in map-based positioning is limited to laboratory settings and to relatively simple environments.

8.1 Map Building

There are two fundamentally different starting points for the map-based positioning process. Either there is a pre-existing map, or the robot has to build its own environment map. Rencken [1993] defined the map building problem as the following: "*Given the robot's position and a set of measurements, what are the sensors seeing*?" Obviously, the map-building ability of a robot is closely related to its sensing capacity.

Talluri and Aggarwal [1993] explained:

> *"The position estimation strategies that use map-based positioning rely on the robot's ability to sense the environment and to build a representation of it, and to use this representation effectively and efficiently. The sensing modalities used significantly affect the map making strategy. Error and uncertainty analyses play an important role in accurate position estimation and map building. It is important to take explicit account of the uncertainties; modeling the errors by probability distributions and using Kalman filtering techniques are good ways to deal with these errors explicitly."*

Talluri and Aggarwal [1993] also summarized the basic requirements for a map:

> *"The type of spatial representation system used by a robot should provide a way to incorporate consistently the newly sensed information into the existing world model. It should also provide the necessary information and procedures for estimating the position and pose of the robot in the environment. Information to do path planning, obstacle avoidance, and other navigation tasks must also be easily extractable from the built world model."*

Hoppen et al. [1990] listed the three main steps of sensor data processing for map building:

1. Feature extraction from raw sensor data.
2. Fusion of data from various sensor types.
3. Automatic generation of an environment model with different degrees of abstraction.

And Crowley [1989] summarized the construction and maintenance of a composite local world model as a three-step process:

1. Building an abstract description of the most recent sensor data (a sensor model).
2. Matching and determining the correspondence between the most recent sensor models and the current contents of the composite local model.
3. Modifying the components of the composite local model and reinforcing or decaying the confidences to reflect the results of matching.

A problem related to map-building is "autonomous exploration." In order to build a map, the robot must explore its environment to map uncharted areas. Typically it is assumed that the robot begins its exploration without having any knowledge of the environment. Then, a certain motion strategy is followed which aims at maximizing the amount of charted area in the least amount of time. Such a motion strategy is called exploration strategy, and it depends strongly on the kind of sensors used. One example for a simple exploration strategy based on a lidar sensor is given by [Edlinger and Puttkamer, 1994].

8.1.1 Map-Building and Sensor Fusion

Many researchers believe that no single sensor modality alone can adequately capture all relevant features of a real environment. To overcome this problem, it is necessary to combine data from different sensor modalities, a process known as *sensor fusion*. Here are a few examples:

- Buchberger et al. [1993] and Jörg [1994; 1995] developed a mechanism that utilizes heterogeneous information obtained from a laser-radar and a sonar system in order to construct a reliable and complete world model.
- Courtney and Jain [1994] integrated three common sensing sources (sonar, vision, and infrared) for sensor-based spatial representation. They implemented a feature-level approach to sensor fusion from multisensory grid maps using a mathematical method based on *spatial moments* and *moment invariants*, which are defined as follows:

The two-dimensional $(p+q)$th order spacial moments of a grid map $G(x,y)$ are defined as

$$m_{pq} = \sum_x \sum_y x^p y^q G(x,y) \qquad p,q = 0,1,2,.. \tag{8.1}$$

Using the centroid, translation-invariant central moments (moments don't change with the translation of the grid map in the world coordinate system) are formulated:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q G(x,y) \tag{8.2}$$

From the second- and third-order central moments, a set of seven moment invariants that are independent of translation, rotation, and scale can be derived. A more detailed treatment of spatial moments and moment invariants is given in [Gonzalez and Wintz, 1977].

8.1.2 Phenomenological vs. Geometric Representation, Engelson and McDermott [1992]

Most research in sensor-based map building attempts to minimize mapping errors at the earliest stage — when the sensor data is entered into the map. Engelson and McDermott [1992] suggest that this methodology will reach a point of diminishing returns, and hence further research should focus on explicit error detection and correction. The authors observed that the geometric approach attempts to build a more-or-less detailed geometric description of the environment from perceptual data. This has the intuitive advantage of having a reasonably well-defined relation to the real world. However, there is, as yet, no truly satisfactory representation of uncertain geometry, and it is unclear whether the volumes of information that one could potentially gather about the shape of the world are really useful.

To overcome this problem Engelson and McDermott suggested the use of a *topological* approach that constitutes a *phenomenological representation* of the robot's potential interactions with the world, and so directly supports navigation planning. Positions are represented relative to local reference frames to avoid unnecessary accumulation of relative errors. Geometric relations between frames are also explicitly represented. New reference frames are created whenever the robot's position uncertainty grows too high; frames are merged when the uncertainty between them falls sufficiently low. This policy ensures locally bounded uncertainty. Engelson and McDermott showed that such error correction can be done without keeping track of all mapping decisions ever made. The methodology makes use of the environmental structure to determine the essential information needed to correct mapping errors. The authors also showed that it is not necessary for the decision that caused an error to be specifically identified for the error to be corrected. It is enough that the *type* of error can be identified. The approach has been implemented only in a simulated environment, where the effectiveness of the phenomenological representation was demonstrated.

8.2 Map Matching

One of the most important and challenging aspects of map-based navigation is *map matching*, i.e., establishing the correspondence between a current *local map* and the stored global map [Kak et al., 1990]. Work on map matching in the computer vision community is often focused on the general problem of matching an image of arbitrary position and orientation relative to a model (e.g., [Talluri and Aggarwal, 1993]). In general, matching is achieved by first extracting features, followed by determination of the correct correspondence between image and model features, usually by some form of constrained search [Cox, 1991].

Such matching algorithms can be classified as either *icon-based* or *feature-based*. Schaffer et al. [1992] summarized these two approaches:

> *"Iconic-based pose estimation pairs sensory data points with features from the map, based on minimum distance. The robot pose is solved for that minimizes the distance error between the range points and their corresponding map features. The robot pose is solved [such as to] minimize the distance error between the range points and their corresponding map features. Based on the new pose, the correspondences are recomputed and the process repeats until the change in aggregate distance error between points and line segments falls below a threshold. This algorithm differs from the feature-based method in that it matches every range data point to the map rather than corresponding the range data into a small set of features to be matched to the map. The feature-based estimator, in general, is faster than the iconic estimator and does not require a good initial heading estimate. The iconic estimator can use fewer points than the feature-based estimator, can handle less-than-ideal environments, and is more accurate. Both estimators are robust to some error in the map."*

Kak et al. [1990] pointed out that one problem in map matching is that the sensor readings and the world model may be of different formats. One typical solution to this problem is that the approximate position based on odometry is utilized to generate (from the prestored global model), an estimated visual scene that would be “seen” by robot. This estimated scene is then matched against the actual scene viewed by the onboard sensors. Once the matches are established between the features of the two images (expected and actual), the position of the robot can be estimated with reduced uncertainty. This approach is also supported by Rencken [1994], as will be discussed in more detail below.

In order to match the current sensory data to the stored environment model reliably, several features must be used simultaneously. This is particularly true for a range image-based system since the types of features are limited to a range image map. Long walls and edges are the most commonly used features in a range image-based system. In general, the more features used in one match, the less likely a mismatch will occur, but the longer it takes to process. A realistic model for the odometry and its associated uncertainty is the basis for the proper functioning of a map-based positioning system. This is because the feature detection as well as the updated position calculation rely on odometric estimates [Chenavier and Crowley, 1992].

8.2.1 Schiele and Crowley [1994]

Schiele and Crowley [1994] discussed different matching techniques for matching two occupancy grids. The first grid is the local grid that is centered on the robot and models its vicinity using the most recent sensor readings. The second grid is a global model of the environment furnished either by learning or by some form of computer-aided design tool. Schiele and Crowley propose that two representations be used in environment modeling with sonars: *parametric primitives* and an *occupancy grid*. Parametric primitives describe the limits of free space in terms of segments or surfaces defined by a list of parameters. However, noise in the sensor signals can make the process of grouping sensor readings to form geometric primitives unreliable. In particular, small obstacles such as table legs are practically impossible to distinguish from noise.

Schiele and Crowley discuss four different matches:

- Matching segment to segment as realized by comparing segments in (1) similarity in orientation, (2) collinearity, and (3) overlap.
- Matching segment to grid.
- Matching grid to segment.
- Matching grid to grid as realized by generating a mask of the local grid. This mask is then transformed into the global grid and correlated with the global grid cells lying under this mask. The value of that correlation increases when the cells are of the same state and decreases when the two cells have different states. Finally finding the transformation that generates the largest correlation value.

Schiele and Crowley pointed out the importance of designing the updating process to take into account the uncertainty of the local grid position. The correction of the estimated position of the robot is very important for the updating process particularly during exploration of unknown environments.

Figure 8.2 shows an example of one of the experiments with the robot in a hallway. Experimental results obtained by Schiele and Crowley show that the most stable position estimation results are obtained by matching segments to segments or grids to grids.

8.2.2 Hinkel and Knieriemen [1988] — The *Angle Histogram*

Hinkel and Knieriemen [1988] from the University of Kaiserslautern, Germany, developed a world-modeling method called the *Angle Histogram*. In their work they used an in-house developed lidar mounted on their mobile robot *Mobot III*. Figure 8.3 shows that lidar system mounted on *Mobot III's*

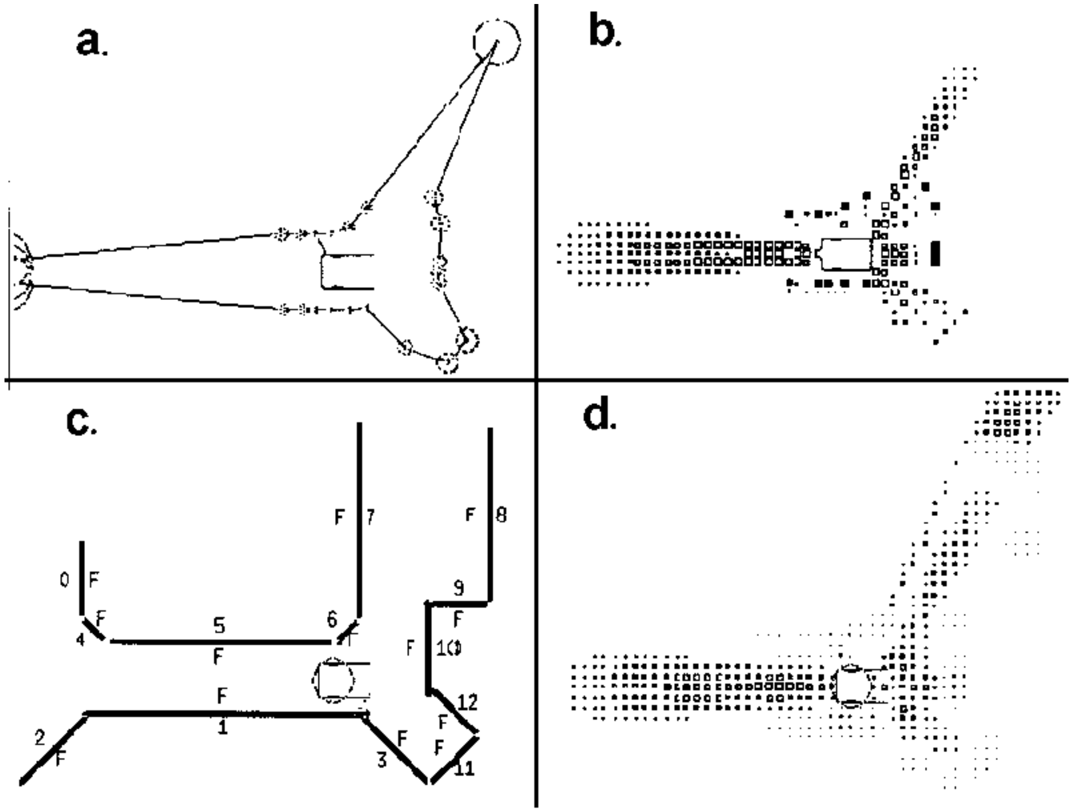


Figure 8.2: Schiele and Crowley's robot models its position in a hallway.
a. Raw ultrasonic range data projected onto external coordinates around the robot.
b. Local grid and the edge segments extracted from this grid.
c. The robot with its uncertainty in estimated position within the global grid.
d. The local grid imposed on the global grid at the position and orientation of best correspondence.
(Reproduced and adapted from [Schiele and Crowley, 1994].)

successor *Mobot IV*. (Note that the photograph in Figure 8.3 is very recent; it shows *Mobot IV* on the left, and *Mobot V*, which was built in 1995, on the right. Also note that an ORS-1 lidar from ESP, discussed in Sec. 4.2.2, is mounted on *Mobot V*.)

A typical scan from the in-house lidar is shown in Figure 8.4. The similarity between the scan quality of the University of Kaiserslautern lidar and that of the ORS-1 lidar (see Fig. 4.32a in Sec. 4.2.6) is striking.

The angle histogram method works as follows. First, a 360 degree scan of the room is taken with the lidar, and the resulting “hits” are recorded in a map. Then the algorithm measures the relative angle δ between any two adjacent hits (see Figure 8.5). After compensating for noise in the readings (caused by the inaccuracies in position between adjacent hits), the angle histogram shown in Figure 8.6a can be built. The uniform direction of the main walls are clearly visible as peaks in the angle histogram. Computing the histogram modulo π results in only two main peaks: one for each pair of parallel walls. This algorithm is very robust with regard to openings in the walls, such as doors and windows, or even cabinets lining the walls.

Figure 8.3: *Mobot IV* (left) and *Mobot V* (right) were both developed and built at the University of Kaiserslautern. The different *Mobot* models have served as mobile robot testbeds since the mid-eighties. (Courtesy of the University of Kaiserslautern.)

After computing the angle histogram, all angles of the hits can be normalized, resulting in the

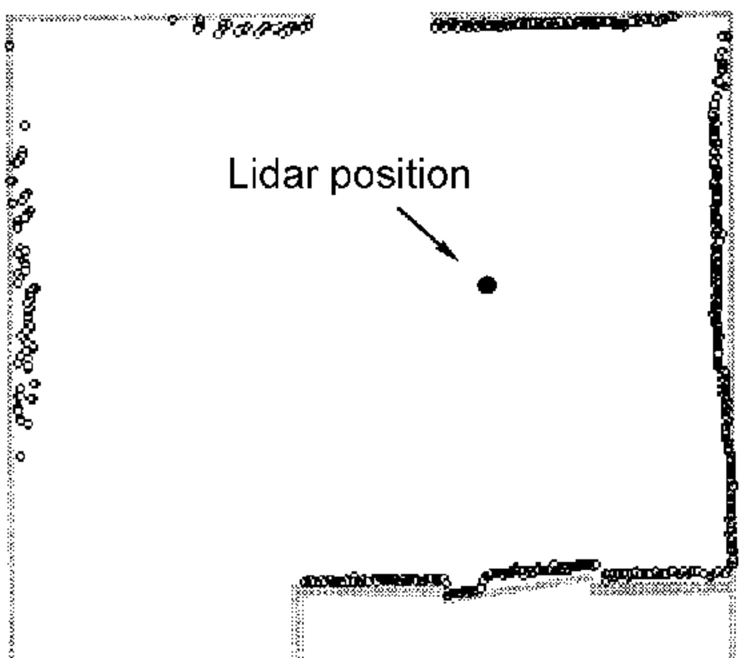


Figure 8.4: A typical scan of a room, produced by the University of Kaiserslautern's in-house developed lidar system. (Courtesy of the University of Kaiserslautern.)

representation shown in Figure 8.6b. After this transformation, two additional histograms, one for the x- and one for the y-direction can be constructed. This time, peaks show the distance to the walls in x and y direction. During operation, new orientation and position data is updated at a rate of 4 Hz. (In conversation with Prof. Von Puttkamer, Director of the Mobile Robotics Laboratory at the University of Kaiserslautern, we learned that this algorithm had since been improved to yield a reliable accuracy of 0.5°.)

8.2.3 Weiß, Wetzler, and Puttkamer — More on the Angle Histogram

Weiß et al. [1994] conducted further experiments with the angle histogram method. Their work aimed at matching rangefinder scans from different locations. The purpose of this work was to compute the translational and rotational displacement of a mobile robot that had traveled during subsequent scans.

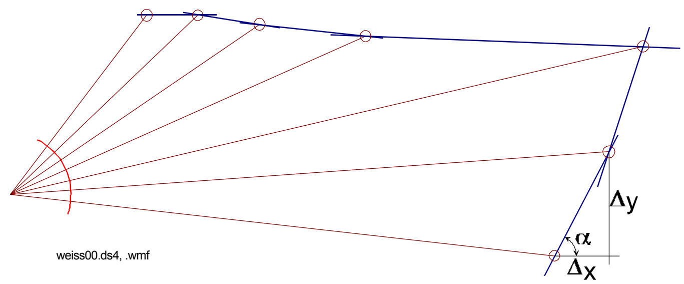


Figure 8.5: Calculating angles for the angle histogram. (Courtesy of [Weiß et al., 1994].)

The authors pointed out that an angle histogram is mostly invariant against rotation and translation. If only the orientation is altered between two scans, then the angle histogram of the second scan will show only a phase shift when compared to the first. However, if the position of the robot is altered, too, then the distribution of angles will also change. Nonetheless, even in that case the new angle histogram will still be a representation of the distribution of directions in the new scan. Thus, in the new angle histogram the same direction that appeared to be the local maximum in the old angle histogram will still appear as a maximum, provided the robot's displacement between the two scans was sufficiently small.

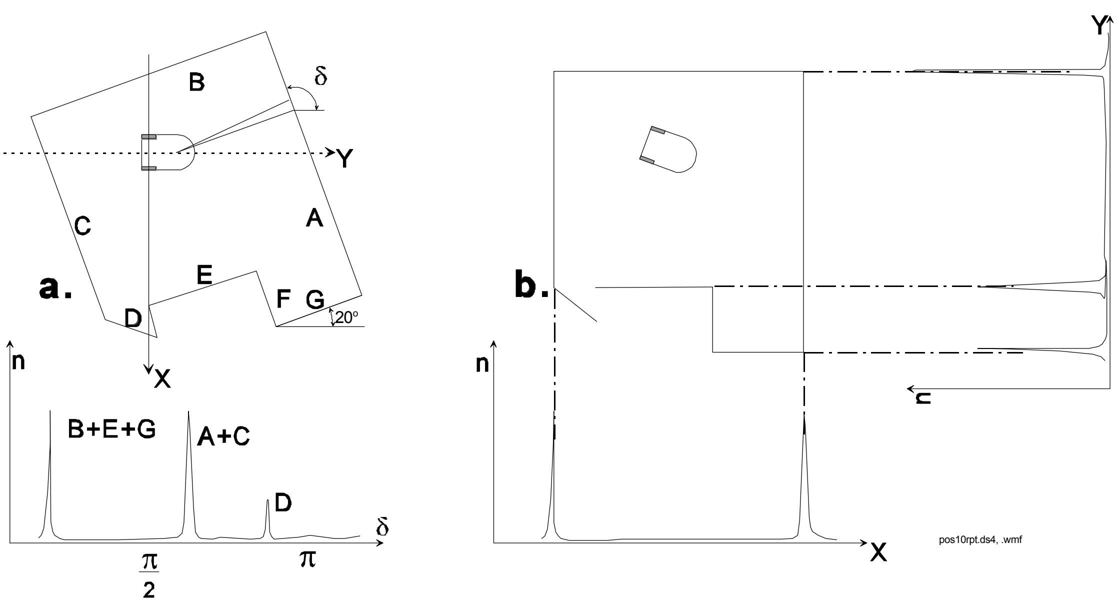


Figure 8.6: Readings from a rotating laser scanner generate the contours of a room.
a. The angle histogram allows the robot to determine its orientation relative to the walls.
b. After normalizing the orientation of the room relative to the robot, an x-y histogram can be built form the same data points. (Adapted from [Hinkel and Knieriemen, 1988].)

Experiments show that this approach is highly stable against noise, and even moving obstacles do not distort the result as long as they do not represent the majority of matchable data. Figure 8.7a shows two scans taken from two different locations. The second scan represents a rotation of +43 degrees, a translation in x-direction of +14 centimeters and a translation in y-direction of +96 centimeters. Figure 8.7b shows the angle histogram associated with the two positions. The maxima for the main directions are -24 and 19 degrees, respectively. These angles correspond to the rotation of the robot relative to the local main direction. One can thus conclude that the rotational displacement of the robot was 19° -(-24°) = +43°. Furthermore, rotation of the first and second range plot by -24 and 19 degrees, respectively, provides the normalized x- and y-plots shown in Fig 8.7c. The cross correlation of the x translation is shown in Figure 8.7d. The maximum occurs at -35 centimeters, which corresponds to -14 centimeters in the rotated scan (Fig. 8.7a). Similarly, the y-translation can be found to be +98 centimeters in the rotated scan. Figure 8.5e shows the result of scan matching after making all rotational and translational corrections.

Definition

A cross-correlation is defined as

$$c(y) = \lim_{X\to\infty} \frac{1}{2X} \int_{-X}^{X} f(x)\, g(x+y)\, dx \quad . \tag{8.3}$$

c(y) is a measure of the cross-correlation between two stochastic functions regarding the phase shift y. The cross-correlation c(y) will have an absolute maximum at s, if f(x) is equal to g(x+s). (Courtesy of [Weiß et al., 1994].)

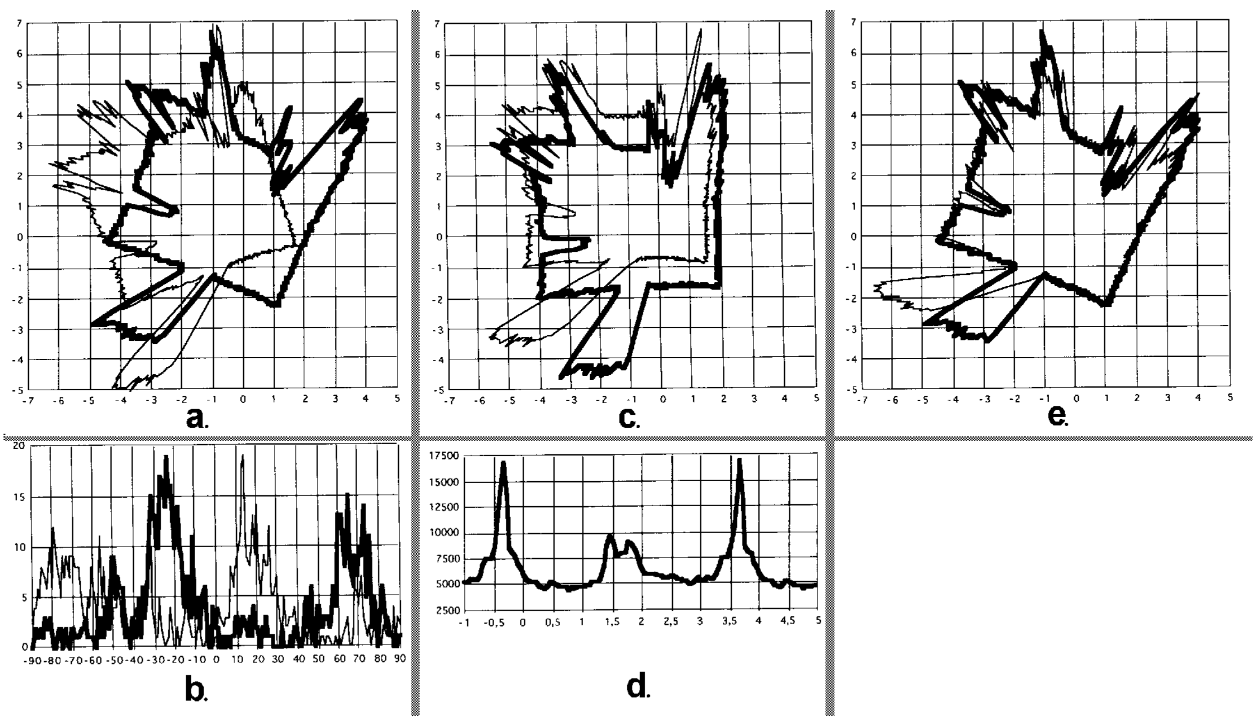


Figure 8.7: Various stages during the matching of two angle histograms. The two histograms were built from scan data taken from two different locations. (Courtesy of [Weiß et al., 1994].)

a. Two scans with rotation of +43°, x-transition of +14 cm, y-transition of +96 cm.
b. Angle histogram of the two positions.
c. Scans rotated according to the maximum of their angle histogram (+24 °, -19°).
d. Cross-correlation of the x-translation (maximum at -35 cm, corresponding to -14 cm in the rotated scan).
e. x-translation correction of +14 cm; y-translation correction of -98 cm.

8.2.4 Siemens' Roamer

Rencken [1993; 1994] at the Siemens Corporate Research and Development Center in Munich, Germany, has made substantial contributions toward solving the boot strap problem resulting from the uncertainty in position and environment. This problem exists when a robot must move around in an unknown environment, with uncertainty in its odometry-derived position. For example, when building a map of the environment, all measurements are necessarily relative to the carrier of the sensors (i.e., the mobile robot). Yet, the position of the robot itself is not known exactly, because of the errors accumulating in odometry.

Rencken addresses the problem as follows: in order to represent features "seen" by its 24 ultrasonic sensors, the robot constructs hypotheses about these features. To account for the typically unreliable information from ultrasonic sensors, features can be classified as hypothetical, tentative, or confirmed. Once a feature is confirmed, it is used for constructing the map as shown in Figure 8.8. Before the map can be updated, though, every new data point must be associated with either a plane, a corner, or an edge (and some variations of these features). Rencken devices a "hypothesis tree" which is a data structure that allows tracking of different hypotheses until a sufficient amount of data has been accumulated to make a final decision.

One further important aspect in making this decision is feature visibility. Based on internal models for different features, the robot's decisions are aided by a routine check on visibility. For example, the visibility of edges is smaller than that of corners. The visibility check further reduces the uncertainty and improves the robustness of the algorithm.

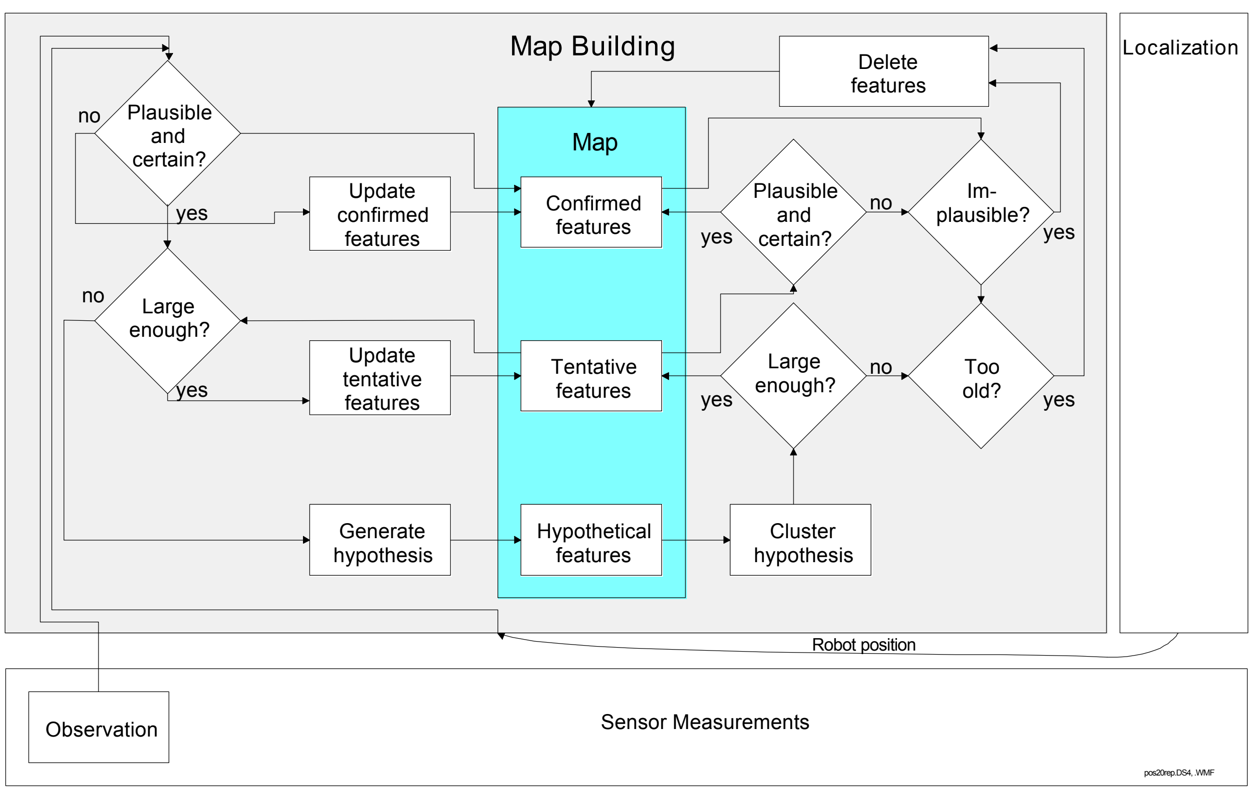


Figure 8.8: The basic map-building algorithm maintains a hypothesis tree for the three sensor reading categories: hypothetical, tentative, and confirmed. (Adapted from [Rencken, 1994].)

Based on the above methods, Rencken [1993] summarizes his method with the following procedure:

1. Predict the robot's position using odometry.
2. Predict the associated covariance of this position estimate.
3. Among the set of given features, test which feature is visible to which sensor and predict the measurement.
4. Compare the predicted measurements to the actual measurements.
5. Use the error between the validated and predicted measurements to estimate the robot's position.
6. The associated covariance of the new position estimate is also determined.

The algorithm was implemented on Siemens' experimental robot Roamer (see Fig. 8.9). In an endurance experiment, Roamer traveled through a highly cluttered office environment for approximately 20 minutes. During this time, the robot updated its internal position only by means of odometry and its map-building capabilities. At a relatively slow travel speed of 12 cm/s (4¾ in/s) Roamer's position accuracy was periodically recorded, as shown in Table 8.1.

Table 8.1: Position and orientation errors of Siemens' Roamer robot in an map-building "endurance test." (Adapted from [Rencken, 1994].)

Time [min:sec]	Pos. Error [cm] (in)	Orientation error [°]
5:28	5.8 (2-1/4)	-7.5
11:57	5.3 (2)	-6.2
14:53	5.8 (2-1/4)	0.1
18:06	4.0 (1-1/2)	-2.7
20:12	2.5 (1)	3.0

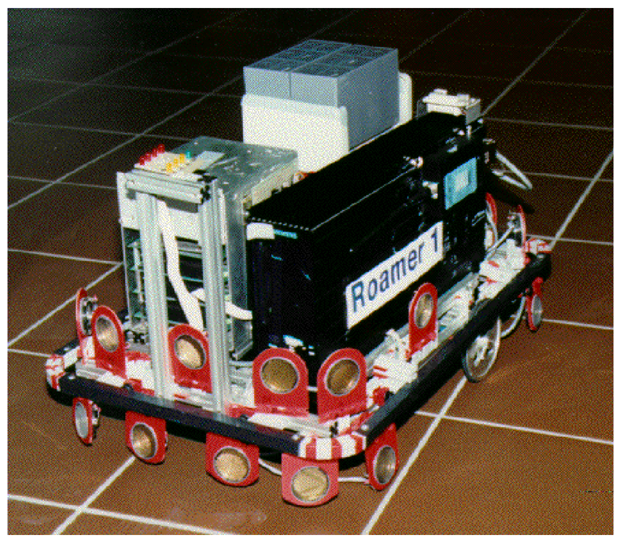


Figure 8.9: Siemens' Roamer robot is equipped with 24 ultrasonic sensors. (Courtesy of Siemens).

8.2.5 Bauer and Rencken: Path Planning for Feature-based Navigation

Bauer and Rencken [1995] at Siemens Corporate Research and Development Center in Munich, Germany are developing path planning methods that assist a robot in feature-based navigation. This work extends and supports Rencken's feature-based navigation method described in Section 8.2.4, above.

One problem with all feature-based positioning systems is that the uncertainty about the robot's position grows if there are no suitable features that can be used to update the robot's position. The problem becomes even more severe if the features are to be detected with ultrasonic sensors, which are known for their poor angular resolution. Readings from ultrasonic sensors are most useful when the sound waves are being reflected from a wall that is normal to the incident waves, or from distinct corners.

During operation the robot builds a list of expected sonar measurements, based on earlier measurements and based on the robot's change of location derived from dead-reckoning. If actual sonar readings match the expected ones, these readings are used to estimate the robot's actual position. Non-matching readings are used to define new hypothesis about surrounding features, called tentative features. Subsequent reading will either confirm tentative features or remove them. The existence of confirmed features is important to the system because each confirmed feature offers essentially the benefits of a navigation beacon. If further subsequent readings match confirmed features, then the robot can use this data to reduce its own growing position uncertainty. Bauer and Rencken show that the growing uncertainty in the robot's position (usually visualized by so-called “uncertainty ellipses”) is reduced in one or two directions, depending on whether a new reading matches a confirmed feature that is a line-type (see cases a. and b. in Fig. 8.10) or point-type (case c. in Fig. 8.10).

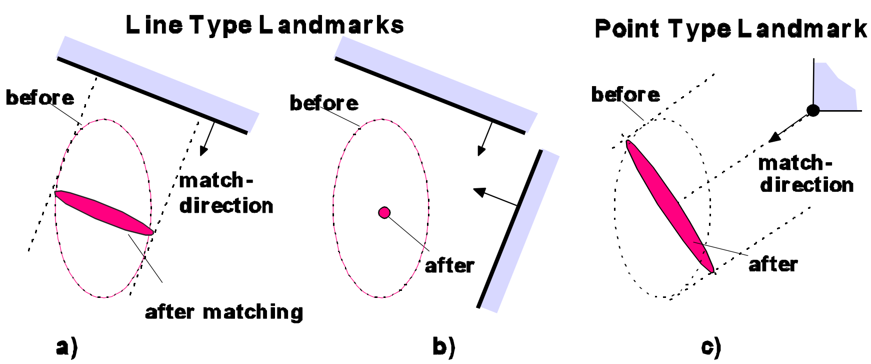


Figure 8.10: Different features can reduce the size of the robot's uncertainty ellipse in one or two directions.
a, c: Walls and corners reduce uncertainty in one direction
b.: Two adjacent walls at right angles reduce uncertainty in two directions.
(Courtesy of [Bauer and Rencken, 1995]).

One novel aspect of Bauer and Rencken's approach is a behavior that steers the robot in such a way that observed features stay in view longer and can thus serve as a navigation reference longer. Fig. 8.11 demonstrates this principle. In the vicinity of a confirmed feature "straight wall" (Fig. 8.11a), the robot will be steered alongside that wall; in the vicinity of a confirmed feature "corner" (Fig. 8.11b) the robot will be steered around that corner.

Experimental results with Bauer and Rencken's method are shown in Figures 8.12 and 8.13. In the first run (Fig. 8.12) the robot was programmed to explore its environment while moving from point A in the office in the upper left-hand corner to point E in the office in the lower right-hand corner. As the somewhat erratic trajectory shows, the robot backed up frequently in order to decrease its position uncertainty (by confirming more features). The actual position accuracy of the robot was mea-

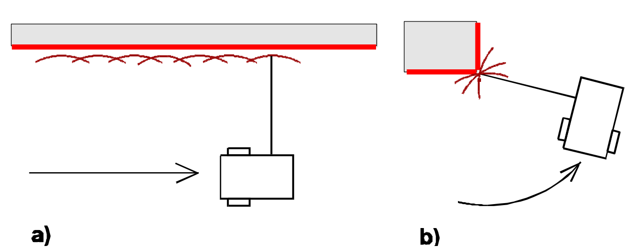


Figure 8.11: Behaviors designed to improve feature-based positioning
a. Near walls, the robot tries to stay parallel to the wall for as long as possible.
b. Near corners the robot tries to trun around the corner for as long as possible.
(Courtesy of [Bauer and Rencken, 1995]).

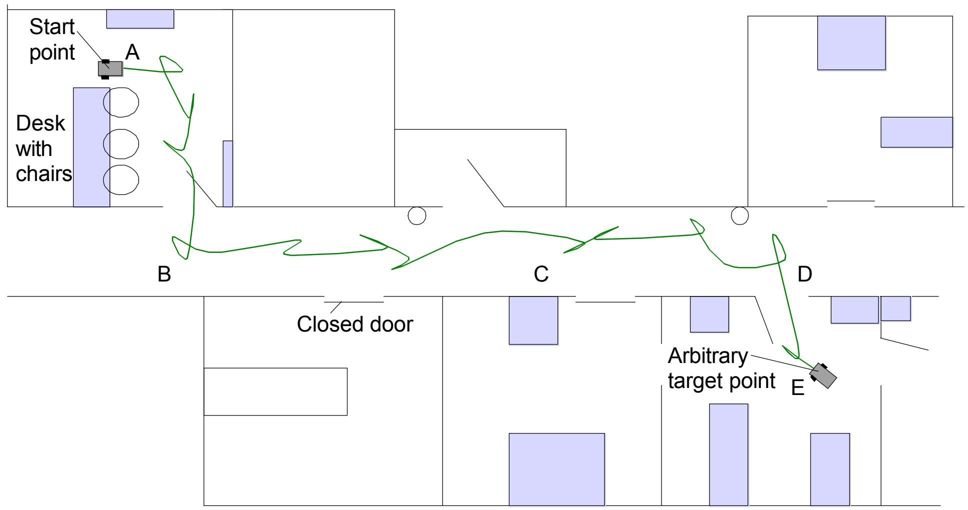


Figure 8.12: Actual office environment and robot's trajectory during the exploratory travel phase. (Courtesy of [Bauer and Rencken, 1995]).

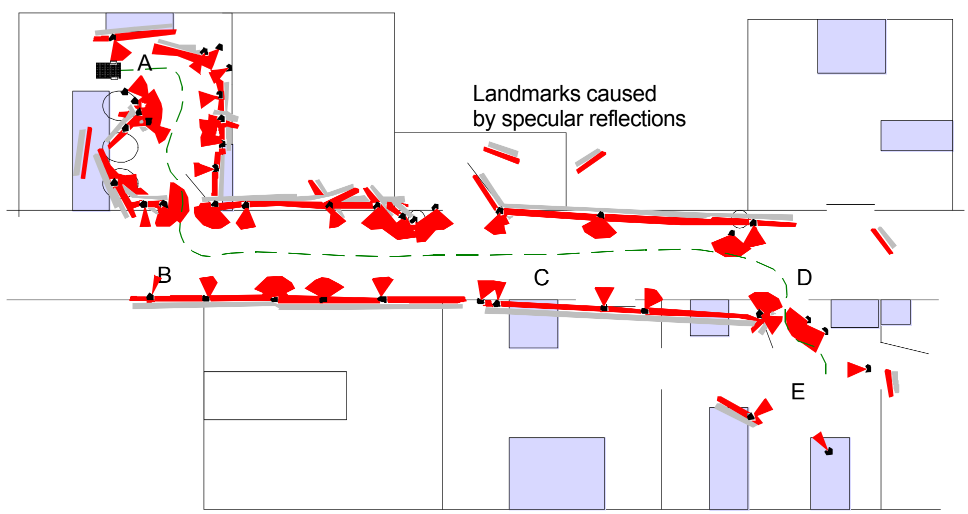


Figure 8.13: Gathered features and robot's return trajectory (Courtesy of [Bauer and Rencken, 1995]).

sured by hand at control points A through E, the results are listed in Table 8.2.

When the robot was programmed to return to its starting position, the resulting path looked much smoother. This is because of the many features that were stored during the outbound trip.

Table 8.2: Hand-measured position error of the robot at intermediate way-points during the exploration phase (Adapted from [Bauer and Rencken, 1995]).

Point	Absolute x,y-coordinates [cm]	Pos. Error [cm] (in)	Orient. Error [°]
A	(0,0)	2.3 (7/8)	0.7
B	(150, -500)	5.7 (2-1/4)	1.9
C	(1000, -500)	9.1 (3-1/2)	5.3
D	(1800,-500)	55.8 (22)	5.9
E	(1800,-800)	63.2 (25)	6.8

8.3 Geometric and Topological Maps

In map-based positioning there are two common representations: geometric and topological maps. A geometric map represents objects according to their absolute geometric relationships. It can be a grid map, or a more abstracted map, such as a line map or a polygon map. In map-based positioning, sensor-derived geometric maps must be matched against a global map of a large area. This is often a formidable difficulty because of the robot's position error. By contrast, the topological approach is based on recording the geometric relationships between the observed features rather than their absolute position with respect to an arbitrary coordinate frame of reference. The resulting presentation takes the form of a graph where the nodes represent the observed features and the edges represent the relationships between the features. Unlike geometric maps, topological maps can be built and maintained without any estimates for the position of the robot. This means that the errors in this representation will be independent of any errors in the estimates for the robot position [Taylor, 1991]. This approach allows one to integrate large area maps without suffering from the accumulated odometry position error since all connections between nodes are relative, rather than absolute. After the map has been established, the positioning process is essentially the process of matching a local map to the appropriate location on the stored map.

8.3.1 Geometric Maps for Navigation

There are different ways for representing geometric map data. Perhaps the simplest way is an occupancy grid-based map. The first such map (in conjunction with mobile robots) was the Certainty Grid developed by Moravec and Elfes, [1985]. In the Certainty Grid approach, sensor readings are placed into the grid by using probability profiles that describe the algorithm's certainty about the existence of objects at individual grid cells. Based on the Certainty Grid approach, Borenstein and Koren [1991] refined the method with the Histogram Grid, which derives a pseudo-probability distribution out of the motion of the robot [Borenstein and Koren, 1991]. The Histogram Grid method is now widely used in many mobile robots (see for example [Buchberger et al., 1993; Congdon et al., 1993; Courtney and Jain, 1994; Stuck et al., 1994; Wienkop et al., 1994].)

A measure of the goodness of the match between two maps and a trial displacement and rotation is found by computing the sum of products of corresponding cells in the two maps [Elfes, 1987]. Range measurements from multiple points of view are symmetrically integrated into the map. Overlapping empty volumes reinforce each other and serve to condense the range of the occupied volumes. The map definition improves as more readings are added. The method deals effectively with clutter and can be used for motion planning and extended landmark recognition.

The advantages of occupancy grid-based maps are that they:
- allow higher density than stereo maps,
- require less computation and can be built more quickly,
- allow for easy integration of data from different sensors, and
- can be used to express statistically the confidence in the correctness of the data [Raschke and Borenstein, 1990].

The disadvantages of occupancy grid-based maps are that they:

- have large uncertainty areas associated with the features detected,
- have difficulties associated with active sensing [Talluri and Aggarwal, 1993],
- have difficulties associated with modeling of dynamic obstacles, and
- require a more complex estimation process for the robot vehicle [Schiele and Crowley, 1994].

In the following sections we discuss some specific examples for occupancy grid-based map matching.

8.3.1.1 Cox [1991]

One typical grid-map system was implemented on the mobile robot Blanche [Cox, 1991]. This positioning system is based on matching a local grid map to a global line segment map. Blanche is designed to operate autonomously within a structured office or factory environment without active or passive beacons. Blanche's positioning system consists of :

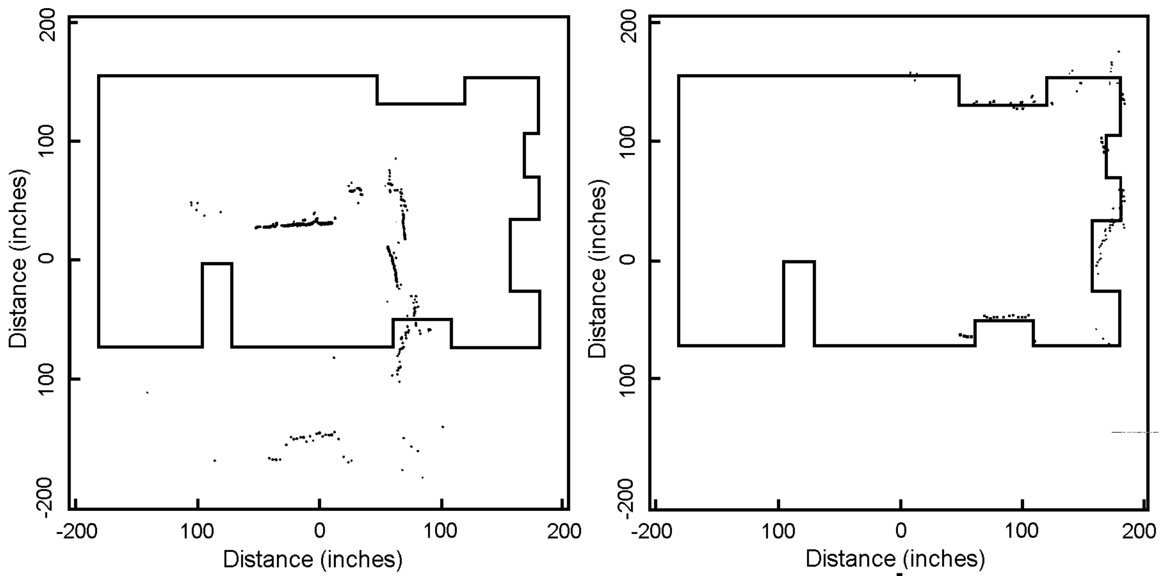
- an a *priori* map of its environment, represented as a collection of discrete line segments in the plane,
- a combination of odometry and a rotating optical range sensor to sense the environment,
- an algorithm for matching the sensory data to the map, where matching is constrained by assuming that the robot position is roughly known from odometry, and
- an algorithm to estimate the precision of the corresponding match/correction that allows the correction to be combined optimally (in a maximum likelihood sense) with the current odometric position to provide an improved estimate of the vehicle's position.

The operation of Cox's map-matching algorithm (item 2, above) is quite simple. Assuming that the sensor hits are near the actual objects (or rather, the lines that represent the objects), the distance between a hit and the closest line is computed. This is done for each point, according to the procedure in Table 8.3 (from [Cox, 1991]).

Table 8.3: Procedure for implementing Cox's [1991] map-matching algorithm .

1. For each point in the image, find the line segment in the model that is nearest to the point. Call this the target.
2. Find the congruence that minimizes the total squared distance between the image points and their target lines.
3. Move the points by the congruence found in step 2.
4. Repeat steps 1 to 3 until the procedure converges.

Figure 8.14 shows how the algorithm works on a set of real data. Figure 8.14a shows the line model of the contours of the office environment (solid lines). The dots show hits by the range sensor. This scan was taken while the robot's position estimate was offset from its true position by 2.75 meters (9 ft) in the x-direction and 2.44 meters (8 ft) in the y-direction. A very small orientation error was also present. After running the map-matching procedure in Table 8.3, the robot corrected its internal position, resulting in the very good match between sensor data and line model, shown in Figure 8.14b. In a longer run through corridors and junctions Blanche traveled at various slow speeds, on the order of 5 cm/s (2 in/s). The maximal deviation of its computed position from the actual position was said to be 15 centimeters (6 in).



a. b.

Figure 8.14: Map and range data a. before registration and b. after registration. (Reproduced and adapted from [Cox, 1991], © 1991 IEEE.)

Discussion

With the grid-map system used in Blanche, generality has been sacrificed for robustness and speed. The algorithm is intrinsically robust against incompleteness of the image. Incompleteness of the model is dealt with by deleting any points whose distance to their target segments exceed a certain limit. In Cox's approach, a reasonable heuristic used for determining correspondence is the minimum Euclidean distance between the model and sensed data. Gonzalez et al. [1992] comment that this assumption is valid only as long as the displacement between the sensed data and the model is sufficiently small. However, this minimization problem is inherently non-linear but is linearized assuming that the rotation angle is small. To compensate for the error introduced due to linearization, the computed position correction is applied to the data points, and the process is repeated until no significant improvement can be obtained [Jenkin et al., 1993].

8.3.1.2 Crowley [1989]

Crowley's [1989] system is based on matching a local line segment map to a global line segment map. Crowley develops a model for the uncertainty inherent in ultrasonic range sensors, and he describes a method for the projection of range measurements onto external Cartesian coordinates. Crowley develops a process for extracting line segments from adjacent collinear range measurements, and he presents a fast algorithm for matching these line segments to a model of the geometric limits for the free-space of the robot. A side effect of matching sensor-based observations to the model is a correction to the estimated position of the robot at the time that the observation was made. The projection of a segment into the external coordinate system is based on the estimate of the position of the vehicle. Any uncertainty in the vehicle's estimated position must be included in the uncertainty of the segment before matching can proceed. This uncertainty affects both the position and orientation of the line segment. As each segment is obtained from the sonar data, it is matched to the composite model. Matching is a process of comparing each of the segments in the composite local model against the observed segment, to allow detection of similarity in orientation, collinearity, and overlap. Each of these tests is made by comparing one of the parameters in the segment representation:

a. Orientation The square of the difference in orientation of the two candidates must be smaller than the sum of the variances.

b. Alignment The square of the difference of the distance from the origin to the two candidates must be smaller than the sum of the corresponding variance.

c. Overlap The difference of the distance between centerpoints to the sum of the half lengths must be smaller than a threshold.

The longest segment in the composite local model which passes all three tests is selected as the matching segment. The segment is then used to correct the estimated position of the robot and to update the model. An explicit model of uncertainty using covariance and Kalman filtering provides a tool for integrating noisy and imprecise sensor observations into the model of the geometric limits for the free space of a vehicle. Such a model provides a means for a vehicle to maintain an estimate of its position as it travels, even in the case where the environment is unknown.

Figure 8.15 shows the model of the ultrasonic range sensor and its uncertainties (shown as the hatched area A). The length of A is given by the uncertainties in robot orientation σ_w and the width is given by the uncertainty in depth σ_D. This area is approximated by an ellipse with the major and minor axis given by σ_w and σ_D.

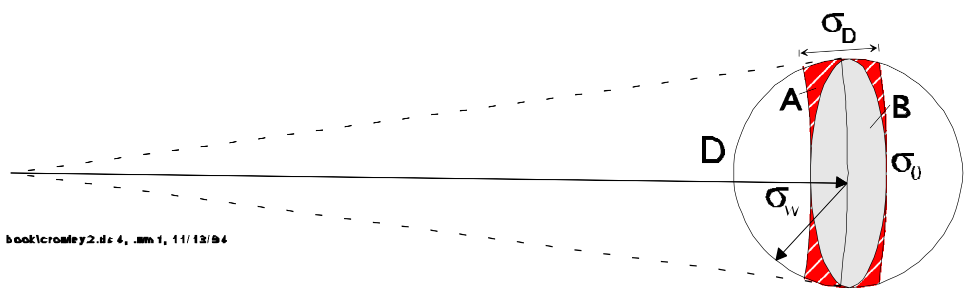


Figure 8.15: Model of the ultrasonic range sensor and its uncertainties. (Adapted from [Crowley, 1989].)

Figure 8.16 shows a vehicle with a circular uncertainty in position of 40 centimeters (16 in) detecting a line segment. The ultrasonic readings are illustrated as circles with a radius determined by its uncertainty as defined in Figure 8.15. The detected line segment is illustrated by a pair of parallel lines. (The actual line segment can fall anywhere between the two lines. Only uncertainties associated with sonar readings are considered here.)

Figure8.16b shows the segment after the uncertainty in the robot's position has been added to the segment uncertainties. Figure8.16c shows the uncertainty in position after correction by matching a model segment. The position uncertainty of the vehicle is reduced to an ellipse with a minor axis of approximately 8 centimeters (3.15 in).

In another experiment, the robot was placed inside the simple environment shown in Figure 8.17. Segment 0 corresponds to a wall covered with a textured wall-paper. Segment 1 corresponds to a metal cabinet with a sliding plastic door. Segment 2 corresponds to a set of chairs pushed up against

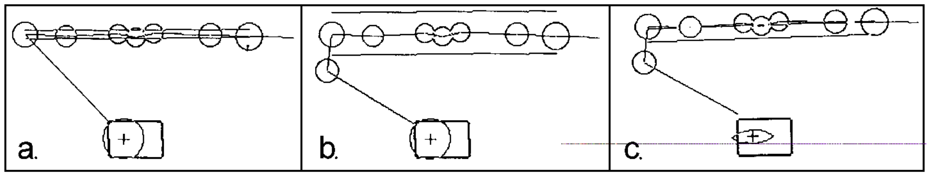


Figure 8.16:
a. A vehicle with a position uncertainty of 40 cm (15.7 in), as shown by the circle around the centerpoint (cross), is detecting a line segment.
b. The boundaries for the line segment grow after adding the uncertainty for the robot's position.
c. After correction by matching the segment boundaries with a stored map segment, the uncertainty of the robot's position is reduced to about 8 cm (3.15 in) as shown by the squat ellipse around the robot's center (cross).
Courtesy of [Crowley, 1989].

two tables. The robot system has no a priori knowledge of its environment. The location and orientation at which the system was started were taken as the origin and x-axis of the world coordinate system. After the robot has run three cycles of ultrasonic acquisition, both the estimated position and orientation of the vehicle were set to false values. Instead of the correct position ($x = 0$, $y = 0$, and $\theta = 0$), the position was set to $x = 0.10$ m, $y = 0.10$ m and the orientation was set to 5 degrees. The uncertainty was set to a standard deviation of 0.2 meters in x and y, with a uncertainty in orientation of 10 degrees. The system was then allowed to detect the "wall" segments around it. The resulting estimated position and covariance is listed in Table 8.4].

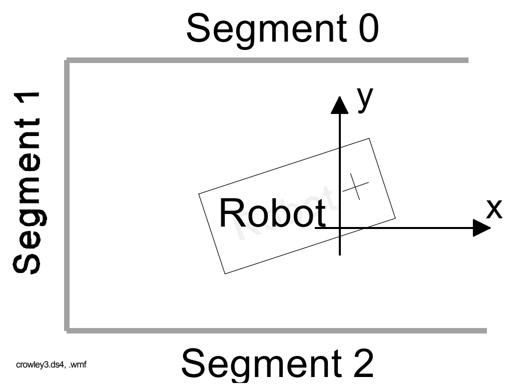


Figure 8.17: Experimental setup for testing Crowley's map-matching method. Initially, the robot is intentionally set-off from the correct starting position.

Table 8.3: Experimental results with Crowley's map-matching method. Although initially placed in an incorrect position, the robot corrects its position error with every additional wall segment scanned.

Initial estimated position (with deliberate initial error)	x,y,θ = (0.100, 0.100, 5.0)
Covariance	0.040 0.000 0.000 0.000 0.040 0.000 0.000 0.000 100.0
After match with segment 0 estimated position:	x,y,θ = (0.102, 0.019, 1.3)
Covariance	0.039 0.000 0.000 0.000 0.010 0.000 0.000 0.000 26.28
After match with segment 1 estimated position:	x,y,θ = (0.033, 0.017, 0.20)
Covariance	0.010 0.000 0.000 0.000 0.010 0.000 0.000 0.000 17.10

8.3.1.3 Adams and von Flüe

The work by Adams and von Flüe follows the work by Leonard and Durrant-Whyte [1990] in using an approach to mobile robot navigation that unifies the problems of obstacle detection, position estimation, and map building in a common multi-target tracking framework. In this approach a mobile robot continuously tracks naturally occurring indoor targets that are subsequently treated as "beacons." Predicted targets (i.e., those found from the known environmental map) are tracked in order to update the position of the vehicle. Newly observed targets (i.e., those that were not predicted) are caused by unknown environmental features or obstacles from which new tracks are initiated, classified, and eventually integrated into the map.

Adams and von Flüe implemented the above technique using real sonar data. The authors note that a good sensor model is crucial for this work. For this reason, and in order to predict the expected observations from the sonar data, they use the sonar model presented by Kuc and Siegel [1987].

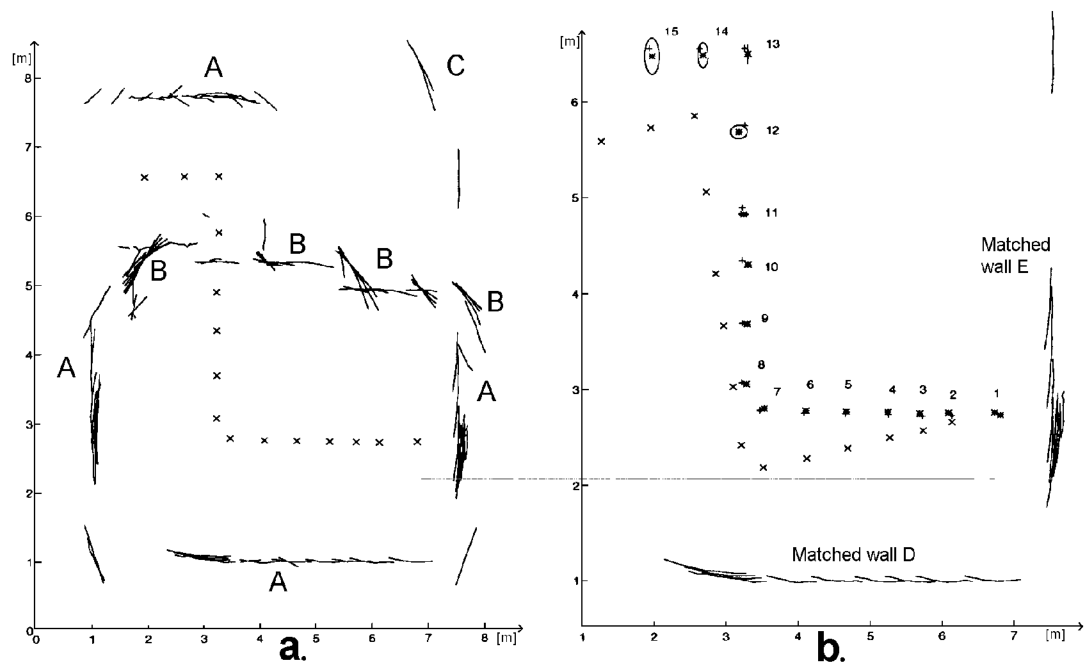


Figure 8.18: a. Regions of constant depth (RCD's) extracted from 15 sonar range scans.
b. True (x), odometric (+), and estimated (*) positions of the mobile robot using two planar (wall) "beacons" for localization. (Courtesy of Adams and von Flüe.)

Figure 8.18a shows regions of constant depth (RCDs) [Kuc and Siegel, 1987] that were extracted from 15 sonar scans recorded from each of the locations marked "×."

The model from Kuc and Siegel's work suggests that RCDs such as those recorded at the positions marked A in Figure 8.18a correspond to planar surfaces; RCDs marked B rotate about a point corresponding to a 90 degree corner and RCDs such as C, which cannot be matched, correspond to multiple reflections of the ultrasonic wave.

Figure 8.18b shows the same mobile robot run as Figure 8.18a, but here the robot computes its position from two sensed "beacons," namely the wall at D and the wall at E in the right-hand scan in Figure 8.18b. It can be seen that the algorithm is capable of producing accurate positional estimates of the robot, while simultaneously building a map of its sensed environment as the robot becomes more confident of the nature of the features.

8.3.2 Topological Maps for Navigation

Topological maps are based on recording the geometric relationships between the observed features rather than their absolute position with respect to an arbitrary coordinate frame of reference. Kortenkamp and Weymouth [1994] defined the two basic functions of a topological map:

a. Place Recognition With this function, the current location of the robot in the environment is determined. In general, a description of the place, or node in the map, is stored with the place. This description can be abstract or it can be a local sensory map. At each node, matching takes place between the sensed data and the node description.

b. Route Selection With this function, a path from the current location to the goal location is found.

The following are brief descriptions of specific research efforts related to topological maps.

8.3.2.1 Taylor [1991]

Taylor, working with stereo vision, observed that each local stereo map may provide good estimates for the relationships between the observed features. However, because of errors in the estimates for the robot's position, local stereo maps don't necessarily provide good estimates for the coordinates of these features with respect to the base frame of reference. The recognition problem in a topological map can be reformulated as a graph-matching problem where the objective is to find a set of features in the relational map such that the relationships between these features match the relationships between the features on the object being sought. Reconstructing Cartesian maps from relational maps involves minimizing a non-linear objective function with multiple local minima.

8.3.2.2 Courtney and Jain [1994]

A typical example of a topological map-based approach is given by Courtney and Jain [1994]. In this work the coarse position of the robot is determined by classifying the map description. Such classification allows the recognition of the workspace region that a given map represents. Using data collected from 10 different rooms and 10 different doorways in a building (see Fig. 8.19), Courtney and Jain estimated a 94 percent recognition rate of the rooms and a 98 percent recognition rate of the doorways. Courtney and Jain concluded that coarse position estimation, or place recognition, in indoor domains is possible through classification of grid-based maps. They developed a paradigm wherein pattern classification techniques are applied to the task of mobile robot localization. With this paradigm the robot's workspace is represented as a set of grid-based maps interconnected via topological relations. This representation scheme was chosen over a single global map in order to avoid inaccuracies due to cumulative dead-reckoning error. Each region is represented by a set of multi-sensory grid maps, and feature-level sensor fusion is accomplished through extracting spatial descriptions from these maps. In the navigation phase, the robot localizes itself by comparing features extracted from its map of the current locale with representative features of known locales in the environment. The goal is to recognize the current locale and thus determine the workspace region in which the robot is present.

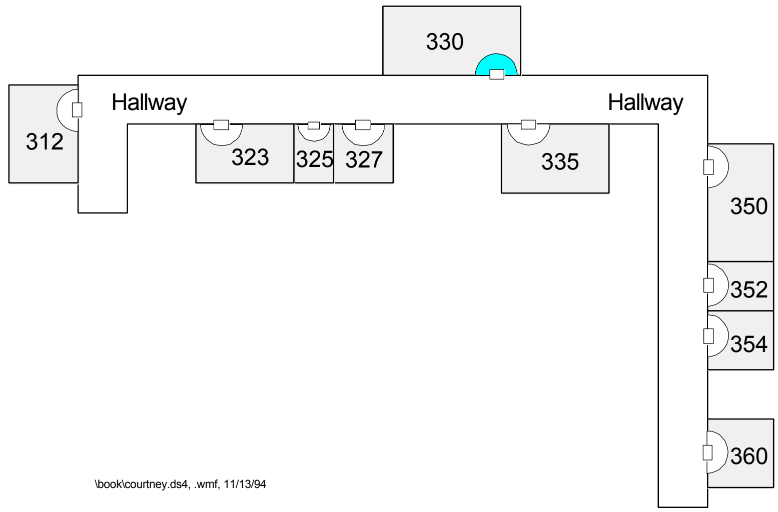


Figure 8.19: Based on datasets collected from 10 different rooms and 10 different doorways in a building, Courtney and Jain estimate a 94 percent recognition rate of the rooms and a 98 percent recognition rate of the doorways. (Adapted from [Courtney and Jain, 1994].)

8.3.2.3 Kortenkamp and Weymouth [1993]

Kortenkamp and Weymouth implemented a cognitive map that is based on a topological map. In their topological map, instead of looking for places that are locally distinguishable from other places and then storing the distinguishing features of the place in the route map, their algorithm looks for places that mark the transition between one space in the environment and another space (gateways). In this algorithm sonar and vision sensing is combined to perform place recognition for better accuracy in recognition, greater resilience to sensor errors, and the ability to resolve ambiguous places. Experimental results show excellent recognition rate in a well-structured environment. In a test of seven gateways, using either sonar or vision only, the system correctly recognized only four out of seven places. However, when sonar and vision were combined, all seven places were correctly recognized. Figure 8.20 shows the experimental space for place recognition. Key locations are marked in capital letters. Table 8.5a and Table 8.5b show the probability for each place using only vision and sonar, respectively. Table 8.5c shows the combined probabilities (vision and sonar) for each place. In spite of the good results evident from Table 8.5c, Kortenkamp and Weymouth pointed out several drawbacks of their system:

The robot requires several initial, guided traversals of a route in order to acquire a stable set of location cues so that it can navigate autonomously.

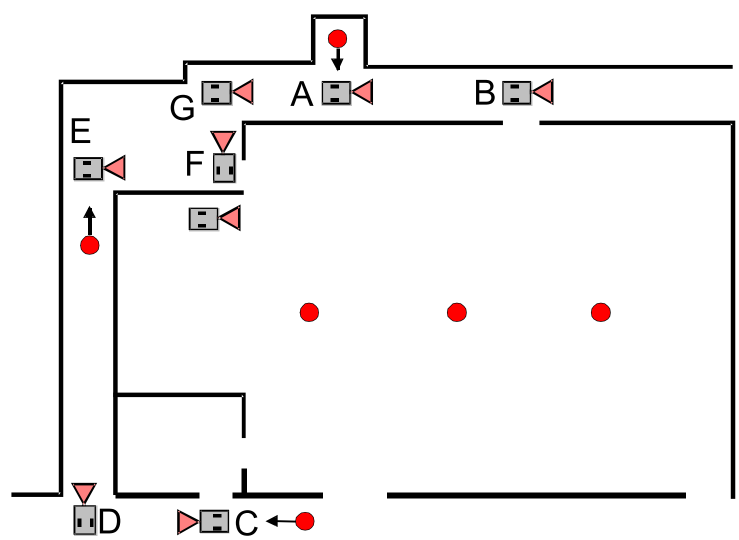


Figure 8.20: An experiment to determine if the robot can detect the same place upon return at a later time. In this case, multiple paths through the place can be "linked" together to form a network. (Adapted from [Kortenkamp and Weymouth, 1994].)

- Acquiring, storing, and matching visual scenes is very expensive, both in computation time and storage space.
- The algorithm is restricted to highly structured, orthogonal environments.

Table 8.5a: Probabilities for each place using only vision.

	Stored Places						
	A	B	C	D	E	F	G
A	0.43	0.09	0.22	0.05	0.05	0.1	0.06
B	0.05	0.52	0.21	0.06	0.05	0.05	0.05
C	0.10	0.12	0.36	0.2	0.04	0.13	0.04
D	0.14	0.05	0.24	0.43	0.05	0.04	0.05
E	0.14	0.14	0.14	0.14	0.14	0.14	0.14
F	0.14	0.14	0.14	0.16	0.14	0.14	0.14
G	0.14	0.14	0.14	0.14	0.14	0.14	0.14

Table 8.5b: Probabilities for each place using only sonar.

	Stored Places						
	A	B	C	D	E	F	G
A	0.82	0.04	0.04	0.04	0.04	0	0
B	0.02	0.31	0.31	0.31	0.06	0	0
C	0.02	0.31	0.31	0.31	0.06	0	0
D	0.02	0.31	0.31	0.31	0.61	0	0
E	0.04	0.12	0.12	0.12	0.61	0	0
F	0	0	0	0	0	0.90	0.10
G	0	0	0	0	0	0.10	0.90

Table 8.5c: Combined probabilities (vision and sonar) for each place.

	Stored Places						
	A	B	C	D	E	F	G
A	0.95	0.01	0.02	0.01	0.01	0	0
B	0	0.65	0.26	0.07	0.01	0	0
C	0	0.17	0.52	0.29	0.01	0	0
D	0.01	0.07	0.33	0.58	0.01	0	0
E	0.04	0.12	0.12	0.12	0.61	0	0
F	0	0	0	0	0	0.90	0.1
G	0	0	0	0	0	0.09	0.91

8.4 Summary

Map-based positioning is still in the research stage. Currently, this technique is limited to laboratory settings and good results have been obtained only in well-structured environments. It is difficult to judge how the performance of a laboratory robot scales up to a real world application. Kortenkamp and Weymouth [1994] noted that very few systems tested on real robots are tested under realistic conditions with more than a handful of places.

We summarize relevant characteristics of map-based navigation systems as follows:

Map-based navigation systems:
- are still in the research stage and are limited to laboratory settings,
- have not been tested extensively in real-world environments,
- require a significant amount of processing and sensing capability,
- need extensive processing, depending on the algorithms and resolution used,
- require initial position estimates from odometry in order to limit the initial search for features to a smaller area.

There are several critical issues that need to be developed further:
- Sensor selection and sensor fusion for specific applications and environments.
- Accurate and reliable algorithms for matching local maps to the stored map.
- Good error models of sensors and robot motion.
- Good algorithms for integrating local maps into a global map.