36. World Modeling

Wolfram Burgard, Martial Hebert

In this chapter we describe popular ways to represent the environment of a mobile robot. For indoor environments, which are often stored using two-dimensional representations, we discuss occupancy grids, line maps, topological maps, and landmark-based representations. Each of these techniques has its own advantages and disadvantages. Whilst occupancy grid maps allow for quick access and can efficiently be updated, line maps are more compact. Also landmark-based maps can efficiently be updated and maintained, however, they do not readily support navigation tasks such as path planning like topological representations do.

Additionally, we discuss approaches suited for outdoor terrain modeling. In outdoor environments, the flat-surface assumption underling many mapping techniques for indoor environments is no longer valid. A very popular approach in this context are elevation and variants maps, which store the surface of the terrain over a regularly spaced grid. Alternatives to such maps are point clouds, meshes, or three-dimensional grids, which provide a greater flexibility but have higher storage demands.

The construction of models of the environment is crucial to the development of several applications of mobile robot systems. It is through these environment models that the robot can adapt its decisions to the current state of the world. The models are constructed from sensor data as the robot discovers its environment. There are three challenges in constructing environment models from sensor data. First, the models must be compact so that they can be used efficiently by other components of the system, such as path planners. Second, the models must be adapted to the task and to the type of environment. For example, modeling the environment as a set of planes is not relevant to a robot operating in natural terrain. In particular, this implies that a *universal* representation for mobile robots in not possible and that we must instead choose from an array of different approaches. Third, the representation must accommodate the uncertainty inherent to both sensor data and to the robot's state estimation system. The latter point is particular important since environment models typically accumulate sensor readings over substantial distances into a common reference frame. Drift in position estimates are unavoidable and must be accounted for in the model representation and in its construction.

36.1 Historical Overview

Historically, work in this area concentrated first on robots operating in indoor environments. In that case, the models take advantage of the fact that the world can be represented as vertical structures on reference ground planes. This simplification can be used for representing the world as a two-dimensional (2-D) grid. Uncertainty in the measurements and in the robot's pose can then be modeled by using probabilities of occupancy in the grid rather than binary occupied/empty flags. Another feature of indoor environments is that they are highly structured in that they contain primarily linear structures such as lines and planes. This observations led to a second class of representations based on collections of points, lines, and planes to represent the environment. Here again, a lot of attention was paid to representing the uncertainty on the relative poses of these geometric elements, including a considerable amount of work in the 1980s on using Kalman filters and other probabilistic techniques.

With continued progress on sensing (e.g., longer-range image laser scanners and stereo vision), and on mechanical and controls aspects of mobile robots systems, it became possible to develop mobile robot systems for operations in unstructured, natural terrain, motivated in part by planetary exploration and military applications. In these scenarios, it is no longer appropriate to project the data in a 2-D grid, and environments cannot be described adequately by a small vocabulary of geometric elements. Since, in most cases, it is still appropriate to assume (at least locally) that there is a reference ground plane, a natural representation is a 2-1/2-D grid, in which each cell contains the elevation (and possibly other features) of the terrain at that location. Although they have been used extensively, the main challenges with such *elevation maps* are that they are not compact representations and that it is difficult to incorporate uncertainty in the representation. Accordingly, much of the research has focused on designing efficient data structures and algorithms for elevation maps, for example, hierarchical representations. Recently, the uncertainty challenge has been tackled by using probabilistic representation of the elevation maps.

While elevation maps provide a natural representation for many types of natural terrains, they cannot represent environments with vertical or overhanging structures. This limitation has become more evident in recent years as applications of mobile robots are increasingly demanding operation in urban environment (in which building walls and other structures cannot be represented by elevation maps) and the use of aerial data, which involves overhanging structures such as tree canopies. This has led to the development of representations that are truly three dimensional (3-D), such as point clouds, 3-D grids, and meshes. The two challenges described above apply here as well, except that the situation is more complicated because of the increased complexity introduced by the third dimension. Current research includes efficient computation over 3-D structures and probabilistic representations of 3-D data.

Because of the large volume of data involved in all of these representations, it is important to be able to group the data into larger chunks corresponding to semantically meaningful parts of the environment. This can be done at different levels, depending on the application the environment. At the lowest level, work in this area involves classifying the points into classes that are relevant to navigation tasks (e.g., distinguishing between vegetation and ground, extracting walls, tree surfaces, etc.). At an intermediate level of representation, this part of the work involves extracting parts of the environments that are considered landmarks of interest for the navigation task (e.g., roads). Finally, the highest representation level involves extracting and representing objects in the environments (e.g., natural obstacles, or specific objects such as cars for operation in urban environments).

All of these representations assume a static environment. In fact, many of the current applications of mobile robots call for operation in mixed environments in which the robot shares its environments with other moving agents, including other robots, people, and vehicles. Assuming that perception algorithms are able to detect and track individual moving objects in the environment, the challenge here is to insert this information into a representation that can be used by a planner. In this case, any one of the previous representations can be used as a foundation, but it needs to be extended to handle the temporal dimension by storing a history of the locations and states of the detected objects. Such representations include the detected trajectories of the detected objects and, in some cases, information on the future predicted trajectories of the objects.

36.2 World Models for Indoors and Structured Environments

36.2.1 Occupancy Grids

Occupancy grid maps, which were introduced in the 1980s by *Moravec* and *Elfes* [36.1], are a popular, probabilistic approach to represent the environment. They are an approximative technique in which we calculate for each cell of a discrete grid the posterior probability that the corresponding area in the environment is occupied by an obstacle. The advantage of occupancy grid maps lies in the fact that they do not rely on any predefined features. Additionally, they offer a constant-time access to grid cells and provide the ability to represent unknown (unobserved) areas, which can be important, for example, in exploration tasks. Their disadvantage lies in potential discretization errors and the high memory requirements.

Throughout this section we assume that the map m consists of a discrete, two-dimensional grid of L cells denoted as $m_1, \ldots, m_L$. Given the sensory input $z_{1:t}$ obtained by the robot at the corresponding positions $x_{1:t}$, the occupancy grid mapping approach calculates a posterior probability $p(m \mid x_{1:t}, z_{1:t})$.

To keep the calculations tractable, the overall approach assumes that the individual cells of the grid are independent, i. e., that the following equation holds:

$$p(m \mid x_{1:t}, z_{1:t}) = \prod_{l=1}^{L} p(m_l \mid x_{1:t}, z_{1:t}) \,. \tag{36.1}$$

Note that this assumption is rather strong. It basically states that any information about the occupancy of one cell does not tell us anything about its neighboring cells. In practice, we often find objects that are larger than the individual grid cells, like doors, cabinets, chairs etc. Thus, if we know that one cell is occupied, for each of its neighboring cells the probability raises that it is occupied. Despite this fact, occupancy grid maps have been successfully applied in numerous installations of mobile robots and have been proven to be a powerful tool that supports various navigation tasks such as localization and path planning.

Because of the independence assumption expressed by (36.1) we can concentrate on the estimation of the occupancy probability of the individual cells m_l in m. Under additional independence assumptions one can finally arrive at the following formula for calculating the occupancy probability $p(m_l \mid x_{1:t}, z_{1:t})$ that cell m_l is occupied given $p(m_l \mid x_{1:t-1}, z_{1:t-1})$ and the new observation z_t observed at position x_t

$$\begin{aligned} &p(m_l \mid x_{1:t}, z_{1:t}) \\ &= \left(1 + \frac{(1 - p(m_l \mid x_t, z_t))}{p(m_l \mid x_t, z_t)} \cdot \frac{p(m_l)}{(1 - p(m_l))} \right. \\ &\quad \left. \times \frac{1 - p(m_l \mid x_{1:t-1}, z_{1:t-1})}{p(m_l \mid x_{1:t-1}, z_{1:t-1})} \right)^{-1} . \end{aligned} \tag{36.2}$$

In practice, one often assumes that the prior $p(m_l)$ is 0.5 so that the second factor in the product becomes 1 and therefore vanishes from the equation.

Additionally, if we define

$$\mathrm{Odds}(x) = \frac{p(x)}{1 - p(x)} \,, \tag{36.3}$$

the incremental updates can be computed using the following equation:

$$\begin{aligned} &\mathrm{Odds}(m_l \mid x_{1:t}, z_{1:t}) \\ &= \mathrm{Odds}(m_l \mid x_t, z_t) \cdot \mathrm{Odds}(m_l)^{-1} \\ &\quad \times \mathrm{Odds}(m_l \mid x_{1:t-1}, z_{1:t-1}) \,. \end{aligned} \tag{36.4}$$

To recover the occupancy probability from the Odds representation given in (36.4), one can use the following formula, which can easily be derived using (36.3):

$$p(x) = \frac{\mathrm{Odds}(x)}{1 + \mathrm{Odds}(x)} \,. \tag{36.5}$$

It remains to describe how to compute the occupancy probability $p(m_l \mid x_t, z_t)$ of a grid cell given a *single* observation z_t and the corresponding pose x_t of the robot. This quantity strongly depends on the sensor of the robot and has to be defined individually for each type of sensor. Additionally, the parameters of these models have to be adapted according to the properties of each sensor. Let us assume that the function $dist(x_t, m_l)$ refers to the distance between the sensor at pose x_t and the center of the cell m_l. Let us first assume that we only need to consider the optical axis of the sensor cone as is, for example, the case for laser beams sent out by a laser range finder. Then, $p(m_l \mid x_t, z_t)$ can be formulated as

$$p(m_l \mid x_t, z_t) = \begin{cases} p_{\text{prior}}, & z_t \text{ is a maximum range reading} \\ p_{\text{prior}}, & m_l \text{ is not covered by } z_t \\ p_{\text{occ}}, & |z_t - \mathrm{dist}(x_t, m_l)| < r/2 \\ p_{\text{free}}, & z_t \geq \mathrm{dist}(x_t, m_l) \end{cases} , \tag{36.6}$$

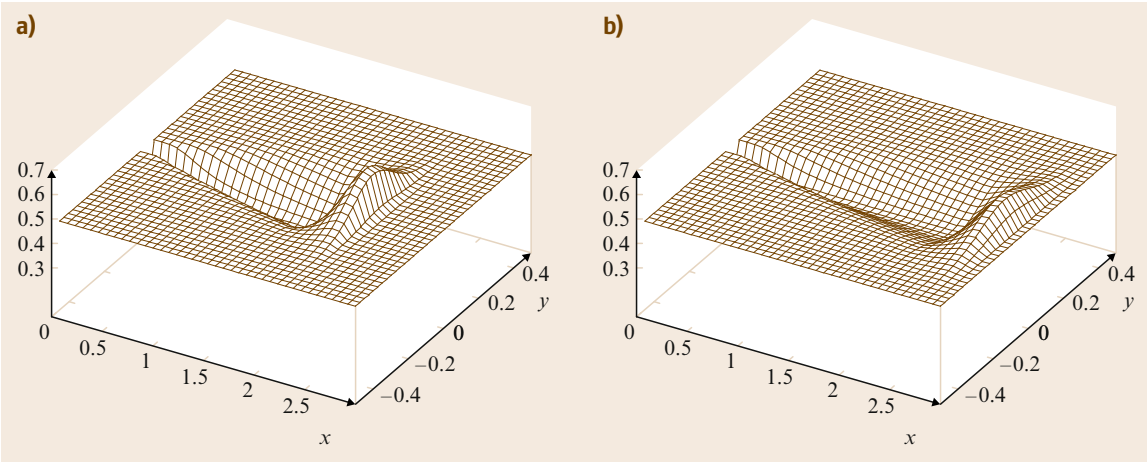

Fig. 36.1a,b Occupancy probability introduced by a single ultrasound measurement of **(a)** $z = 2.0\,\text{m}$ and **(b)** $z = 2.5\,\text{m}$ [36.2]

Fig. 36.2 Incremental mapping in a corridor environment. The *upper left* image shows the initial map and the lower one shows the resulting map. The maps in between are the local maps built from the individual ultrasound scans perceived by the robot [36.2] ◀

where r is the resolution of the grid map. Obviously, it must hold $0 \leq p_{\text{free}} < p_{\text{prior}} < p_{\text{occ}} \leq 1$.

If a sonar sensor is used, the sensor model is slightly more complicated, since the sensor is not a beam sensor and the observations are more noisy than the ones of a laser range finder. In practice, one typically uses a mixture of three functions to express the model. First, the influence of an observation (which is represented by the difference between p_{prior} and p_{occ} as well as between p_{prior} and p_{free}) decreases with the measured distance. Second, the proximity information of a sonar is substantially affected by noise. Therefore, one typically uses a piecewise-linear function to model a smooth transition from p_{free} to p_{occ}. Finally, the sonar sensor should not be modeled as a beam sensor, since it sends out a conic signal. The accuracy of an observation decreases with the angular distance between the cell under consideration and the optical axis of the observation. This is expressed by the derivation from the prior and is typically modeled using a Gaussian with zero mean. Therefore, it is maximal along the optical axis and decreases the bigger the angular distance form the optical axis is [36.2].

Two examples for a resulting model are depicted in Fig. 36.1, which shows two three-dimensional plots of the resulting occupancy probabilities for a meas-

Fig. 36.3 Occupancy grid map obtained from ultrasound data [36.2]

urement of 2 m (Fig. 36.1a) and 2.5 m (Fig. 36.1b). In this figure, the optical axis of the sensor cone was identical with the x-axis and the sensor was placed in the origin of the coordinate frame. As can be seen, the occupancy probability is high for cells whose distance to x_t is close to z_t. It decreases for cells with shorter distance than z_t as well as with increasing values of the angular distance.

Figure 36.2 depicts the mapping process for a sequence of observations recorded with an iRobot B21r robot. The first row shows how a map was built from a sequence of previous ultrasound scans. Afterwards the robot perceived a series of 18 ultrasound scans, each consisting of 24 measurements. The occupancy probabilities for these 18 scans are depicted in rows 2–7. The occupancy probability grid obtained by integrating the individual observations into the map is shown in the last row of this figure. As can be seen, the belief converges to a representation of the corridor structure in which the scans where recorded. A typical resulting map obtained for an indoor environment is depicted in Fig. 36.3.

36.2.2 Line Maps

The representation of the environment by line models is a popular alternative to the grid-based approximations described above. Line models have several advantages over these nonparametric representations. They require substantially less memory than grids and therefore scale better with the size of the environment. They furthermore are more accurate since they do not suffer from discretization problems. In this section, we consider the problem of calculating lines that provide the approximation of a set of range points by a line. If the data points are given by n pairs (x_i, y_i) of Cartesian coordinates, the line that minimizes the squared distances to all points can be calculated in closed form according to

$$\tan 2\phi = \frac{-2\sum_i (\bar{x} - x_i)(\bar{y} - y_i)}{\sum_i \left[(\bar{y} - y_i)^2 - (\bar{x} - x_i)^2\right]} , \tag{36.7}$$

$$r = \bar{x}\cos\phi + \bar{y}\sin\phi , \tag{36.8}$$

where $\bar{x} = \frac{1}{n}\sum_i x_i$ and $\bar{y} = \frac{1}{n}\sum_i y_i$. In these equations, r is the normal distance of the line from the origin and ϕ is angle of the normal.

Unfortunately, there is no closed-form solution for situations in which the data points are generated by multiple linear structures. In such a situation two problems arise. First one has to answer the question how many lines there are, and second one has to solve a data association problem, which is to find the assignment of the data points to the individual lines. Once the number of lines and the associations are known, we can apply (36.7) and (36.8) to calculate the individual line parameters.

There is a popular approach to solve this problem for range scans with multiple linear structures. This approach, which goes back to the work of *Douglas* and *Peucker* [36.3], is also known as the *split-and-merge* algorithm. Its key idea is to recursively subdivide the point set into subsets that can be more accurately approximated by a line. The approach starts with the line calculated from all data points and determines the data point with maximum distance from this line. If this distance is below a given threshold, the algorithm terminates and provides the fitted line as output. Otherwise,

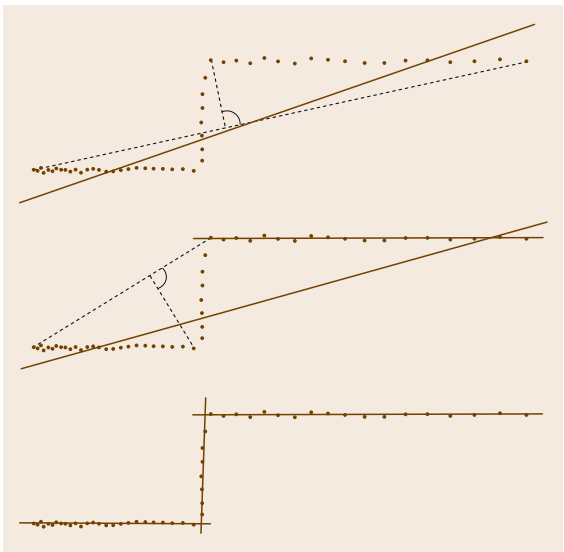

Fig. 36.4 Split-and-merge algorithm applied to a point set

it calculates the point with the largest distance from the line that connects the first and the last point. This point is the so-called splitting point. It then divides the point set into two subsets, one containing all points up to the first to the splitting point and one containing all points after the splitting points. The algorithm is then recursively applied to the two subsets.

Whereas the split-and-merge algorithm is quite effective and efficient, it is not guaranteed that the resulting model is actually optimal, i. e., is the model that minimizes the squared distances of all data points. One approach that is able to find such a model is based on the expectation maximization (EM(algorithm, which in the application described here can be regarded as a variant of the fuzzy k-means clustering algorithm. Let us suppose that the number m of lines in the model θ is known. Let us furthermore suppose that the likelihood of a data point $z = (x, y)$ given the model θ consisting of the line set $\{\theta_1, \ldots, \theta_m\}$ is defined as

$$p(z \mid \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\frac{d(z,\theta_j)^2}{\sigma^2}\right), \tag{36.9}$$

where σ is the standard deviation of the measurement noise and θ_j is the line for which Euclidean distance $d(z, \theta_j)$ to z is minimal.

The goal of the EM algorithm is to generate an iterated sequence of models of increased likelihood. To achieve this, one introduces so-called correspondence variables $c_{ij} \in \{0, 1\}$ that specify to which linear component of the model each point belongs. Since the correct values of these assignment variables are unknown, one estimates a posterior about their values. Let θ_j be a component of the model and z_i be a measurement. Then the expectation about c_{ij}, i. e., that measurement i belongs to line j is computed in the E-step as

$$\begin{aligned} E[c_{ij} \mid \theta_j, z_i] &= p(c_{ij} \mid \theta_j, z_i) && (36.10)\\ &= \alpha\, p(z_i \mid c_{ij}, \theta_j) p(c_{ij} \mid \theta_j) && (36.11)\\ &= \alpha'\, p(z_i \mid \theta_j)\,. && (36.12) \end{aligned}$$

In the M-step, the algorithm then computes the parameters of the model by taking into account the expectations computed in the E-step:

$$\theta_j^* = \arg\min_{\theta_j'} \sum_i \sum_j E[c_{ij} \mid \theta_j, z_i] d^2(z_i, \theta_j')\,. \tag{36.13}$$

Given a fixed variance σ for all data points in z we can determine the most likely model in closed form according to the following equations, which are probabilistic variants of (36.7) and (36.8) that take into account the data association uncertainty given by the expectations calculated in the E-step:

$$\tan 2\phi_j = \frac{-2\sum_i E[c_{ij}|\theta_j, z_i](\bar{x} - x_i)(\bar{y} - y_i)}{\sum_i E[c_{ij}|\theta_j, z_i][(\bar{y} - y_i)^2 - (\bar{x} - x_i)^2]}\,, \tag{36.14}$$

$$r_j = \bar{x}\cos\phi_j + \bar{y}\sin\phi_j\,. \tag{36.15}$$

Here $\bar{x}$ and $\bar{y}$ are computed as

$$\bar{x} = \frac{\sum_i E[c_{ij}|\theta_j, z_i] x_i}{\sum_i E[c_{ij}|\theta_j, z_i]}\,, \quad \bar{y} = \frac{\sum_i E[c_{ij}|\theta_j, z_i] y_i}{\sum_i E[c_{ij}|\theta_j, z_i]}\,. \tag{36.16}$$

Figure 36.5 depicts a line map extracted from 311 823 data points using the EM-based approach. In this example, the model consists of 94 lines. One approach to determine the optimal line is to utilize the Bayesian information criterion [36.4].

Fig. 36.5 Line map consisting of 94 lines generated for 311 823 range points generated with the EM-based approach [36.4]

36.2.3 Topological Maps

In contrast to the previously discussed representations, which mainly focus on the geometric structure of the environment, topological representations have also received significant attraction. One of the pioneering

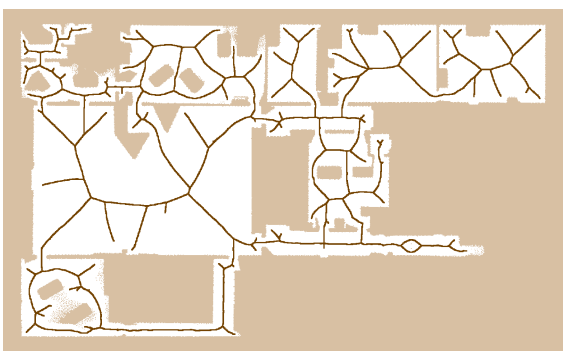

Fig. 36.6 Example of a generalized Voronoi graph [36.5]

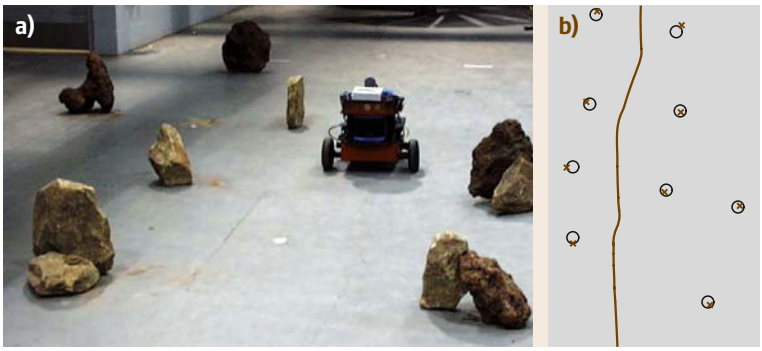

Fig. 36.7 **(a)** Mobile robot mapping a set of stone rocks. The right image **(b)** depicts the path and the estimated landmark positions. The manually determined positions of the rocks are marked by *circles* [36.8]

approaches to topological mapping have been presented in 1988 is the work by *Kuipers* and *Byun* [36.6]. In this approach, the environment is represented by a graph-like structure, in which the nodes are locally distinguishable places and the nodes are travel edges along which the robot can move between the places. Here, distinctive places are identified according to the distance to nearby objects. The distinctive places were been defined by *Choset* and colleagues [36.7] as the meet-points in the generalized Voronoi diagrams, i. e., the points with an out-degree of three or more. A generalized Voronoi diagram is the set of points equidistant from the closest two or more obstacle boundaries. Generalized Voronoi diagrams are a very popular representation as they can be considered as road-maps with a high correspondence to the topological structure of the environment. They have been used extensively for path planning. To plan a path from a starting position to a goal point in the environment, all the robot has to do is to first plan a path to the generalized Voronoi graph, than along the generalized Voronoi graph, and then from the generalized Voronoi graph to the goal point [36.7]. Figure 36.6 shows an example generalized Voronoi graph for an indoor environment. Note that in this figure only those parts of the graph that can be traversed by the robot without colliding with objects are displayed.

36.2.4 Landmark-Based Maps

For environments with locally distinguishable features, landmark-based maps have been extensively used. If we assume that the position of the robot is always known, it simply remains to maintain an estimate about the positions of the individual landmarks over time. In a planar environment, m is represented by K two-dimensional Gaussians with mean μ_k and covariance Σ_k, one for each landmark. If a linearized version of the perception problem is given, the individual Gaussians can be updated using the equations for the extended Kalman filter. Note that, compared to the use of the EKF for landmark-based simultaneous localization and mapping (SLAM), where we needed a $2K+3$-dimensional state vector (3 dimensions for the position of the robot and $2K$ dimensions for the positions of the individual landmarks), we only need K two-dimensional Gaussians to represent the entire map since the robot pose is known. This property, for example, has been utilized in the FastSLAM algorithm [36.8].

The left image of Fig. 36.7 shows a robot equipped with a SICK laser range finder that maps the position of rocks. The right image depicts the path of the vehicle as well as the manually determined and automatically estimated landmark positions.

36.3 World and Terrain Models for Natural Environments

From a survey of environment modeling [36.9] with an emphasis on probabilistic techniques for indoor environments, a taxonomy can be created along several axis: metric versus topological versus semantic, robot-centric versus world-centric, or application-based. We choose to first review purely geometric models (elevation grid, 3-D grid, meshes), then geometric models with low-level attributes (cost maps), and then models with richer semantic attributes, to finish with heterogeneous and hierarchical models.

36.3.1 Elevation Grids

Assuming that the terrain can be represented as a function $h = f(x, y)$, where x and y are the coordinates on a reference place and h is the corresponding eleva-

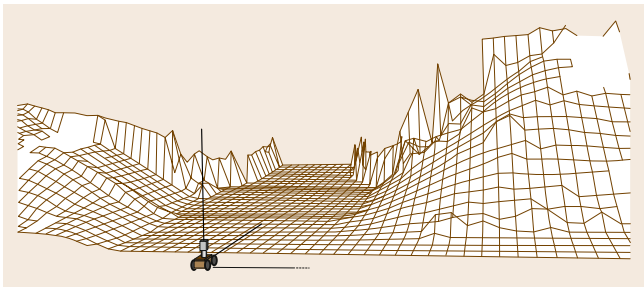

Fig. 36.8 Example elevation map built by accumulating 3-D data from a range sensor

tion. A natural representation is a digital elevation map which stores the value of h at discrete locations (x_i, y_i). Because they are simple data structures and can be generated from sensor data in a relatively straightforward manner, elevation maps have been used extensively for mobile robots operating in natural environments with no vertical surfaces or overhangs (e.g., for planetary exploration scenarios [36.10]). Several issues need to be addressed when using elevation maps for mobile robots.

A regularly sampled grid is appropriate when the sensor data is roughly uniformly distributed on the reference plane. This is the case, for example, with aerial data. For ground robots, however, the distribution of the data on the reference ground plane varies dramatically because of the small incidence angles with the reference plane. This can be addressed by using variable-size cells instead of regularly sampled cells. In that case, the distribution of the cells on the reference plane is designed to approximate the distribution of points that would be measured by the sensor on this reference plane. Such nonuniform representations are useful primarily in cases in which the map is referenced to the current position of the robot [36.11] (Fig. 36.9). Frequent resampling is needed if the map is expressed in a global reference frame.

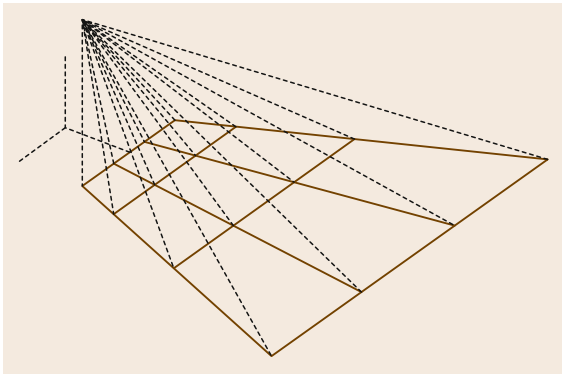

Fig. 36.9 Example of a robot-centric map with variable size (after [36.11])

Irrespective of the sampling scheme used to construct the reference grid, the density of data in the grid varies due to local self-occlusions of the terrain surface and to mismatches between the resolution of the projected sensor data and the resolution of the grid. This can be problematic because, from the point of view of a planner, the elevation map would appear as a scattering of cells with elevation data and cells with no evidence. Various schemes have been proposed to remedy this problem. The basic idea is to estimate the elevation values of the empty cells by interpolating across the cells with know elevation values. This has to be done with great care to avoid filling up regions that correspond to parts of the environment occluded by the terrain (*range shadows*) in which elevation data should not be inferred. Such mistakes can be catastrophic since a planner may generate paths through areas that are entirely unknown. One general approach to this problem is to use surface interpolation techniques including a term that allows for discontinuities in the resulting surfaces [36.12]. An alternative approach is to use the visibility constraints induced by the known geometry of the sensor in order to estimate plausible values of elevation at each cell [36.13, 14].

A difficulty in interpolating techniques is to explicitly take into account the sensor uncertainty. In particular, it is difficult to account for the varying resolution of the sensor a function of range in a fixed-resolution grid. An alternative is to use multiple grids at different resolutions. The appropriate resolution to be used at a particular point (x, y) can be decided based on the range from (x, y) to the closest sensor location to retrieve the value at (x, y) from the appropriate map for that resolution. Generally, the longer the range, the coarser maps that are used. This approach eliminates the gaps in the map due to undersampling at long range from the sensor by using the optimal resolution based on sensor geometry [36.15].

Uncertainty in the sensor measurement is expressed most naturally with respect to the sensor frame, for example, by expressing the uncertainty in the direction of measurement. As a result, converting such uncertainty models to elevation maps is difficult because a distribution in the direction of the measurement maps to a distribution in the reference plane, not a distribution on the elevation value at a particular grid point. When updating a cell based on sensory input, one has to take into account that the uncertainty in a measurement increases

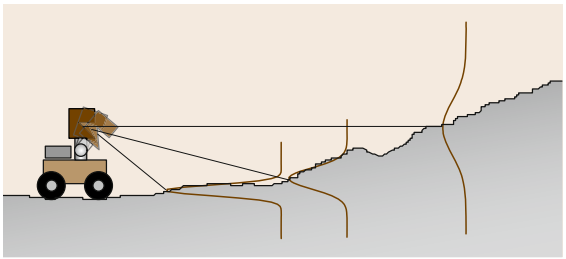

Fig. 36.10 The standard deviation of a height measurements can be modeled to depend linearly on the distance to the grid cell

with the distance measured due to errors in the tilting angle. A popular approach to estimate the height h at location (x, y) is to use a Kalman filter. If we assume that σ is the standard deviation of the current measurement h in the vertical direction in (x, y) and σ_{t-1} is the standard deviation of the current estimate or h_{t-1}, we can apply the following equations to obtain the new estimate h_t with standard deviation σ_t,

$$h_t = \frac{\sigma^2 h_{t-1} + \sigma_{t-1}^2 h}{\sigma_{t-1}^2 + \sigma^2} , \tag{36.17}$$

$$\sigma_t^2 = \frac{\sigma_{t-1}^2 \sigma^2}{\sigma_{t-1}^2 + \sigma^2} . \tag{36.18}$$

One possible solution to deal with varying uncertainties is to use a model in which the standard deviation of the height of a measurement increases linearly with the length of the range measurement, as indicated in Fig. 36.10.

Vegetation cover is another source of error in the recovery of terrain elevation by totally or partially occluding the ground surface from the vehicle sensors. Online learning techniques can address this issue by predicting ground elevation ahead of the vehicle based on past appearance observations of similar terrain and vehicle state over the terrain [36.17].

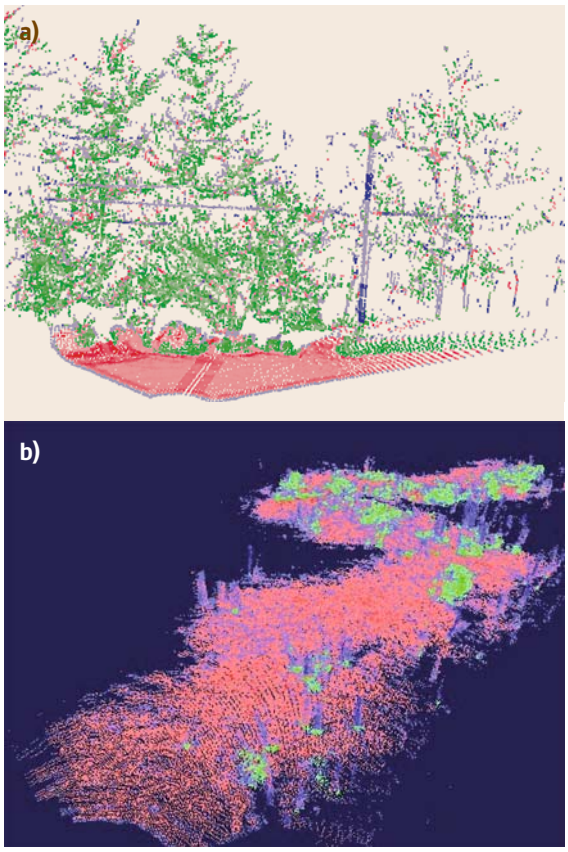

Fig. 36.12a,b Example of mapping and classification of 3-D point sets: **(a)** 3-D data and classification result (green = vegetation, red = surfaces, blue = lines; the saturation of the color is proportional to the confidence in the classification result); **(b)** map accumulated over a large number of scans

Intermediate representations between 2-D elevation maps and full 3-D representations, addressed below, are extended elevation maps [36.18] and so-called multilevel surface maps [36.16]. Such approaches are

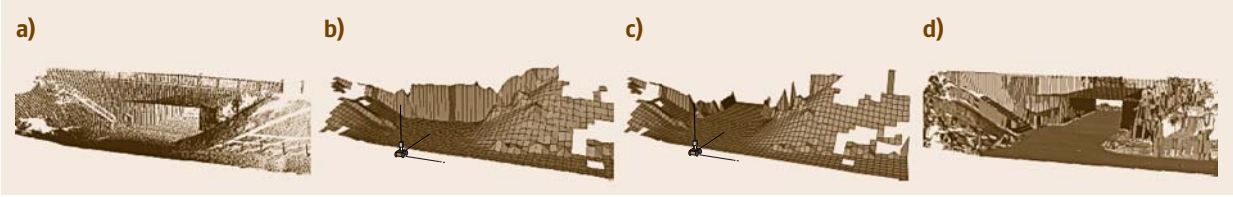

Fig. 36.11a–d Different versions of digital elevation maps [36.16]: **(a)** scan (point set) of a bridge, **(b)** a standard elevation map computed from this data set, **(c)** an extended elevation map, which correctly represents the underpass under the bridge, and **(d)** a multilevel surface map that correctly represents the height of the vertical objects

specifically useful in structured terrain with vertical objects or overhangs such as bridges (Fig. 36.11).

36.3.2 3-D Grids and Point Sets

Elevation maps as described above assume a reference direction. In many cases, this assumption is violated. An alternative is to represent the data directly in 3-D without projecting it onto a reference 2-D plane. The advantage of doing this is that all the sensor data can be preserved in its original distribution and that there is no restriction on the geometry of the environment. The problem is to be able to manipulate efficiently very large sets of 3-D points. Data structures based on dynamic 3-D grids can be used for this purpose [36.21]. An interesting characteristic of these representations is that they enable the computation of cost evaluated from the true local 3-D distribution of the data (as opposed to costs computed from a local surface. This is important, for example, in environments with vegetation scatter, which cannot be modeled as a surface.

In the past, octree-based 3-D representations have been successfully used to map an underwater cave from several sonars mounted on an underwater vehicle [36.19]. The data structure is designed to maintain efficiently, in terms of memory and access time, hundreds of maps to store particles for localization. However, dense 3-D occupancy grids have also been maintained to model natural environments using a radar [36.20]. An example of each approach is presented in Fig. 36.13.

In the approaches mentioned above, the data sets are accumulated, sometimes probabilistically, into discrete environment volumes. A different approach is to represent the environment with discrete sensor samples integrated probabilistically into a metric map [36.22]. Such an approach is different from occupancy grids in several aspects: the storage requirement and resolution is adaptive instead of fixed and the potential for extensibility and scalability is higher.

36.3.3 Meshes

As described above, elevation maps are compact and easy to implement but they are restricted to a particular class of terrain; at the other extreme, using 3-D representations directly is more general but it is also more expensive computationally, and it does not represent explicitly surface continuity. A compromise is to represent the map as a mesh. This approach is attractive because it can, in principle, represent any combination of surfaces. It is also a compact representation in that, even though the size of the mesh may be initially very large, efficient mesh simplification algorithms exist in the literature [36.23] and can be used to reduce the map to a small number of vertices.

The key issue with meshes is that extraction of the correct surfaces from raw data can be difficult in complex environments. In particular, the data is corrupted by sensor noise and by random clutter from other sources, such as vegetation, which cannot be represented as a continuous surface. In addition, it is necessary to detect discontinuities in the data precisely so that disconnected pieces of surface do not become accidentally linked in the mesh formation process [36.18]. Figure 36.14 shows typical mesh representations of an urban environment.

In some applications, the objective is to produce terrain models not to support autonomous navigation but to produce a model to be visualized to a human. City planning is a typical application. Examples

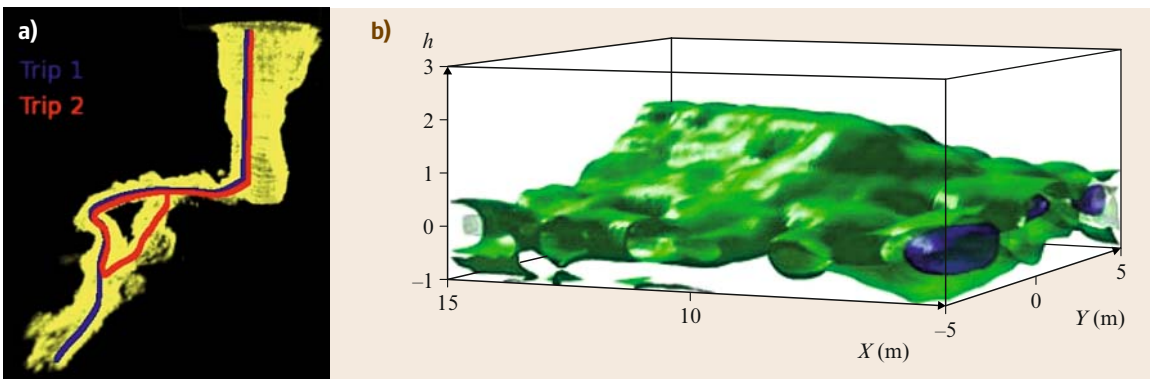

Fig. 36.13a,b Examples of volumetric maps for a natural environment. Whereas **(a)** shows a map of an underwater cave [36.19], **(b)** depicts a terrain map obtained from a radar [36.20]

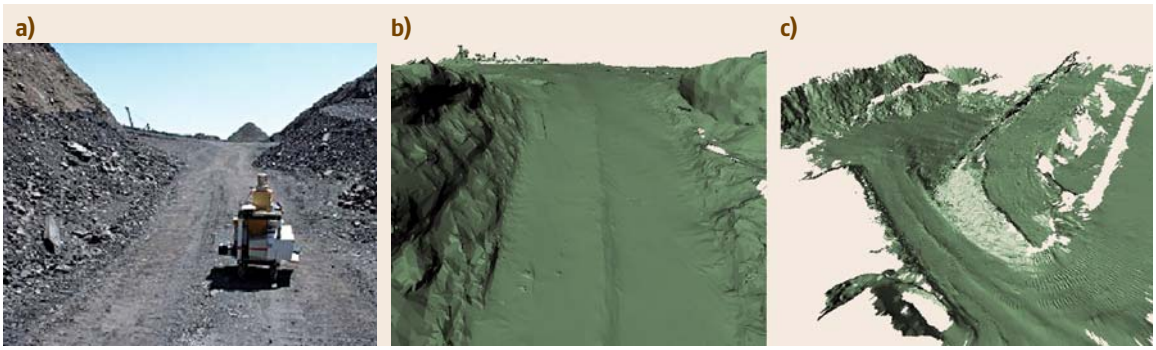

Fig. 36.14a–c Two views **(b–c)** of the mesh representation of a map **(a)** obtained by scanning an environment. The meshes are reduced by a factor of 10 from the initial data sampling

Fig. 36.15a–c Examples of urban terrain modeled as a textured mesh (**(a)** from [36.24], **(b)** from [36.26], and **(c)** from [36.25])

include producing 3-D texture models of urban environments using videos from a camera mounted on a moving vehicle equipped with an inertial navigation system [36.24], using two laser scanners in push-broom mode, one for mapping and the other for localization [36.25], with a camera coregistered with the mapping laser, and collecting high-resolution laser data and imagery to produce geometric and photometric correct 3-D models of buildings [36.26]. Figure 36.15 depicts examples of textured meshes representing urban terrain.

36.3.4 Cost Maps

The most direct use of elevation maps is to compute traversability costs at each cell of the grid. The costs

are computed by comparing the local terrain shape with a kinematic model of the robot. Several approaches have been proposed for computing the costs depending on the exact vehicle model; for example, a cost can be computed by considering the local slope and 3-D texture of the terrain [36.27]. More involved models take into account a detailed model of the robot dynamics. In that case, different costs are generated for different possible robot speeds and path curvatures. The cost grid is used in a minimum-cost planner. Since the terrain map is updated continuously as the robot traverses the environment, it is also necessary to update the costs continuously as new data arrives. Consequently, it is important that the planner is able to support changing costs without needing to process the entire grid every time new data is inserted.

An example of such a combination of grid representation and dynamic planner is the D* system, which uses a version of A* that fully supports dynamic grid updates [36.29–31]. Whenever a grid cell (or a group of grid cells) is updated, the planner makes minimal updates to its internal representation so that the optimal path can be updated quickly.

Defining the exact relation between the costs and the elevation values stored in the grid can be quite difficult. In fact, beyond the limiting cases of terrain with large elevation gradient or large slope, there is little guidance as to why a particular part of the terrain may be more traversable than another as it depends critically on the exact configuration of the robot. For this reason, recent work has focused on deriving cost maps directly from observations, rather than by using handcrafted algorithms. One approach involves learning the best weights to combine a set of predefined costs. Another approach uses costs computed from other sources for inferring how to determine the costs on the terrain currently seen by the robot; for example, data from overhead imagery can be used to predict the traversability costs that should be used for a ground robot [36.28] or for learning to map local elevation distributions to costs values by analyzing actual robot trajectories through the terrain. Similar online learning approaches are used to predict terrain roughness [36.32], terrain slippage [36.33], or traversability in general [36.34, 35] to be used in a cost map.

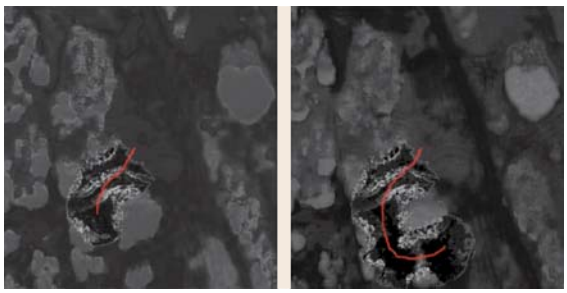

Fig. 36.16 Example of online learning of cost [36.28]. The vehicle trajectory is represented in *red*. As the vehicle progresses, note the change in cost over all the map. The *darker* the appearance, the lower the cost

36.3.5 Semantic Attributes

The representations described above are concerned only with storing the data in a way that is compact and that can be used to estimate drivability costs. Often it is necessary to reason with high-level knowledge about the environment, e.g., the location and types of objects around the robot or the type of terrain (vegetation, mud, wall) in the environment. For convenience we will refer to this type of information as *semantic attributes* attached to different parts of the environment. There are several different ways to approach the problem of generating and representing semantic attributes. One possible direction is to approach the problem as one of extracting *landmarks* from the environment. As for their indoor counterparts, outdoor environments can also be modeled using landmarks. Landmarks are defined broadly as scene elements easy to detect, salient in their surroundings, and easy to recognize. They can be specific objects, such as rocks [36.36], tree trunks in forest [36.37], or locations with a distinct appearance signature in 2-D or 3-D, for urban terrain [36.38, 39] or natural environments [36.40]. Landmarks can be used to produce topological representations of the environment or used in conjunction with metric maps. Recent efforts involve learning a compact representation of environments without defining a priori what a landmark should be [36.41,42] or using statistical learning techniques [36.43] to extract new features from images. Nonlinear dimension-reduction techniques are used to obtain the representation and the results were demonstrated onboard ground and aerial vehicles.

Another view of the problem is as a *classification and grouping* problem. Indeed, the map representations described in the earlier sections are low-level representations in that they do not attempt to group the data points into larger structures that are coherent with respect to geometry, terrain type, or semantic content. In practice, one would like to abstract the data into larger units that can be used by a planner or transmitted to another robot or an operator; for example, in an urban environment, one would like to group the parts of the data that belong

to planes corresponding to pieces of walls. One such example is presented in Fig. 36.18.

One approach is to group 3-D points into components based on low-level classification and feature detection [36.45]. The shape of these component is analyzed further to discriminate between various natural object components (tree trunks, branches) and manmade obstacle (wires). Geometric primitives are then fitted to the components (mesh for the ground surface, ball tree for the vegetation, and cylinder to branches) to produce a compact high-level geometric description of the terrain.

Local terrain classification can be achieved by computing local features from the map and classifying each element of the map in different classes of terrain. This can be done from an elevation map, in which case the *elements* are the cells of the elevation map, or from a point cloud representation in which case the elements are 3-D locations. Given a location in the map x, a feature vector is $V(x)$ is computed in a neighborhood $N(x)$ of x and a classifier $f(V)$ returns the type of terrain at x. Features that have been used in the past are statistics of the distribution of the map data around each element, such as the slope and distribution of elevation [36.46], and the second-order moment of the distribution of the 3-D points in a neighborhood [36.45]. The terrain classes depend on the application. The most direct classification approach uses a binary classifier that separates the terrain into obstacle regions and traversable regions. More involved classifiers segment the map into more classes such as vegetation, solid surfaces, and linear structures [36.13, 45, 47]; an example is shown in Fig. 36.12. In some cases, it is possible to store, with each data element x in the map, the direction in which a measurement was taken for this element. In this case, it is possible to refine the classification by reasoning about the intersection of the measurement ray $d(x)$ with the rest of map; for example, this type of geometric reasoning [36.48] has been exploited to recover the load-bearing surface obscured by vegetation [36.40, 49, 50] and for extracting *negative* obstacles (such as ditches) [36.51]. In all of these cases, the classification information cannot be recovered directly from local statistics and must be inferred from longer-range geometric reasoning.

This class of approaches to local feature classification is similar to approaches for extracting features from images and it suffers from similar limitations. Specifically, these representations are sensitive to the choice of neighborhood used for computing the features. If it is too large, information over a large area is averaged out, leading to poor classification performance. If it is too small, there is not enough information in the neighborhood for reliable classification. The situation is complicated by the fact that the resolution of the map, or more precisely the density of data points in the map, may vary drastically as the distance from the sensor changes. This problem is addressed by using different neighbor-

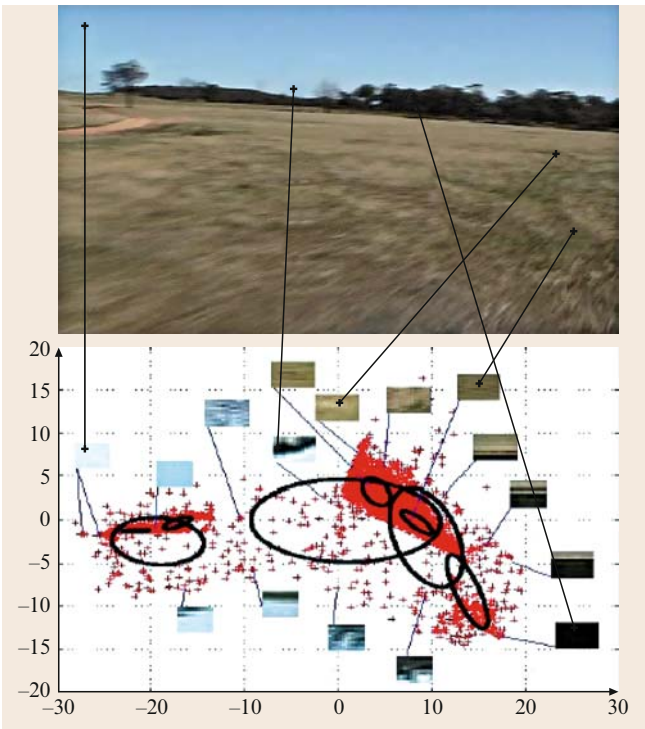

Fig. 36.17 Sample image and low-dimensional embedding of randomly sampled high-dimensional image patches [36.42]

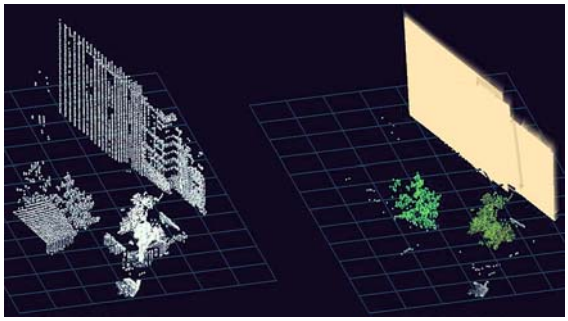

Fig. 36.18 Terrain classification and extraction of geometric features.Whereas the 3-D data is shown on the *left*, the feature map with extracted planes and vegetation regions is shown on the *right* [36.44]

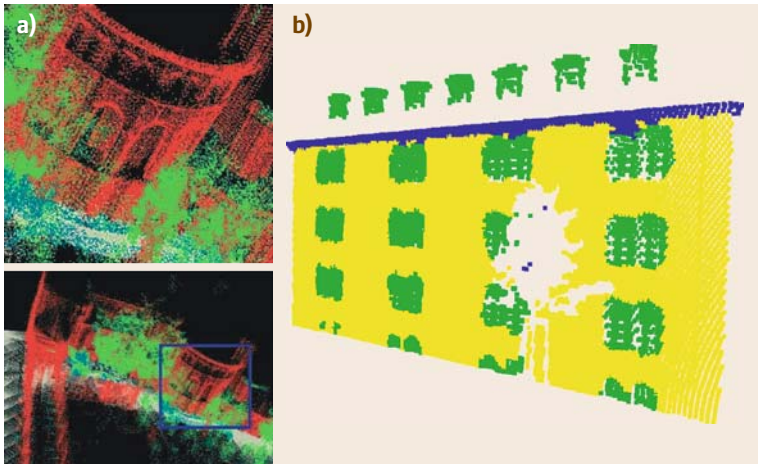

Fig. 36.19 (a) Structure learning for terrain classification [36.54] and (b) building features extraction [36.55]

hood sizes at different locations in the map, or by tuning the classifier differently depending on the map location, typically based on distance from the sensor [36.15, 52]. A second problem is to generate the classifier f. This can be done by using a physical model that predicts the statistics of the local data distribution in the map assuming different terrain types [36.53]. This is generally difficult and a preferred approach is to train the classifier on training data.

This level of classification provides information about the local type of terrain, which can be used in planning, but it does not extract the extended geometric structures that may exist in the environment (Fig. 36.18). Geometric structures such as planar patches can in principle be extracted by using techniques similar to the ones described in the context of indoor environments, such as EM, with the added difficulty that there is a larger amount of clutter that complicates the extraction of the patches. Robust techniques that can handle this level of clutter are typically used for extracting the planes [36.56, 57].

Classification based on local attributes can be improved significantly by taking into account contextual information; for example, hidden Markov models have been successfully used for laser data analysis to determine terrain traversability [36.58]. Another structure learning approach based on Markov random fields in conjunction with margin-maximization criterion has been demonstrated for a wide class of applications, including 3-D terrain classification and object segmentation [36.54] or building structures extraction [36.55] (Fig. 36.19). Finally there are approaches to detect, select, model, and recognize natural landmarks automatically for robot localization and environment mapping [36.36].

36.3.6 Heterogeneous and Hierarchical Models

For long-range navigation, an autonomous vehicle must perform numerous tasks, including absolute and relative localization, path planning, and reactive obstacle avoidance. In addition, it must perform some tasks to fulfill the mission requirements, detecting and modeling objects, for example. To achieve this, a hierarchical framework has been presented that accounts for the different scales and granularities of the representation needed [36.46]. Furthermore, models build from heterogeneous imagery sources (overhead, descent, and ground) have been used to produce 3-D multiresolution terrain models that support Mars exploration [36.59]. The hybrid metric map [36.60], enhances feature maps with metric maps to provide a dense but compact environment representation. Another representation extracts landmarks based on laser/imagery appearance as well as a metric map [36.61].

36.4 Dynamic Environments

The majority of techniques for mapping have been developed for static environments. Certain approaches like occupancy grids or elevation maps can in principle deal with dynamic environments in which objects

Fig. 36.20 (a) A mobile robot acquiring a three-dimensional scan of an urban scene. (b) The people in the scene cause spurious data points in the resulting meshes. (c) The same scene after filtering people

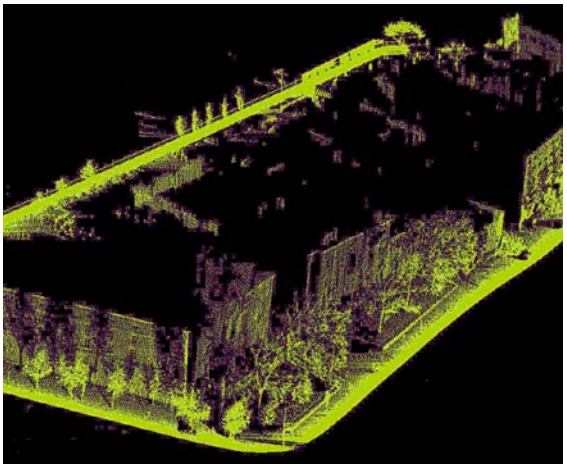

Fig. 36.21 Complex three-dimensional scene acquired with a mobile robot after filtering dynamic objects 36.63

move. Their drawback lies in the fact that the time to unlearn that a cell is free or that the elevation has changed can require as many observations as the robot received with the same area being occupied or with a different elevation. To resolve this problem, several alternative approaches have been proposed in the past. One very popular technique is to track moving objects using feature-based tracking algorithms [36.62, 63]. Such approaches are especially useful when the type of the dynamic objects is known in advance. They have been successfully applied in the context of learning three-dimensional city maps from range data. Figures 36.20 and 36.21 show applications in which dynamic objects were removed from the 3-D data acquired with mobile robots in urban scenes. Alternative approaches to such tracking techniques include those that learn maps on different time scales [36.64], that explicitly learn different states of dynamic environments [36.65], or that only map the static aspects [36.66].

36.4.1 Conclusion and Further Reading

Further reading about typical representations and how they can be utilized for mobile robot navigation can be found in recent textbooks on mobile robotics [36.2, 67, 68]. A comprehensive survey of the fundamentals of spatial data structures and their applications is available in *Samet* [36.69].

References

36.1 H.P. Moravec, A.E. Elfes: High resolution maps from wide angle sonar, Proc. IEEE Int. Conf. Robot. Autom. (ICRA) (1985)

36.2 H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun: *Principles of Robot Motion: Theory, Algorithms and Implementation* (MIT Press, Cambridge 2005)

36.3 D.H. Douglas, T.K. Peucker: Algorithms for the reduction of the number of points required to represent a line or its caricature, Cdn. Cartogr. **10**(2), 112–122 (1973)

36.4 D. Sack, W. Burgard: A comparison of methods for line extraction from range data, Proc. IVAC Symp. Intell. Auton. Vehicles (IAV) (2004)

36.5 P. Beeson, N.K. Jong, B. Kuipers: Towards autonomous topological place detection using the extended Voronoi graph, IEEE Int. Conf. Robot. Autom. (ICRA) (2005)

36.6 B.J. Kuipers, Y.-T. Byun: A robust qualitative method for spatial learning in unknown environments, Proc. Nat. Conf. Artif. Intell. (AAAI) (1988)

36.7 H. Choset, K. Nagatani: Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization, IEEE Trans. Robot. Autom. **17**(2), 125–137 (2001)

36.8 M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit: FastSLAM: a factored solution to the simultaneous localization and mapping problem, Proc. Nat. Conf. Artif. Intell. (AAAI) (2002)

36.9 S. Thrun: Robotic mapping: a survey. In: *Exploring Artificial Intelligence in the New Millenium*, ed. by G. Lakemeyer, B. Nebel (Morgan Kaufmann, New York 2002)

36.10 M. Maimone, P. Leger, J. Biesiadecki: Overview of the Mars exploration rovers' autonomous mobility and vision capabilities, IEEE Int. Conf. Robot. Autom. (2007)

36.11 S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, R. Chatila: Autonomous rover navigation on unknown terrains: functions and integration, Int. J. Robot. Res. **21**(10–11), 917–942 (2002)

36.12 R. Olea: *Geostatistics for Engineers and Earth Scientists* (Kluwer Adacemic, Dordrecht 1999)

36.13 A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarki, P. Rander, S. Thayer, N. Vallidi, R. Warner: Toward reliable off road autonomous vehicles operating in challenging environments, Int. J. Robot. Res. **25**(5-6), 449–483 (2006)

36.14 I.S. Kweon, T. Kanade: High-resolution terrain map from multiple sensor data, IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 278–292 (1992)

36.15 M. Montemerlo, S. Thrun: A multi-resolution pyramid for outdoor robot terrain perception, Proc. AAAI Nat. Conf. Artif. Intell. (San Jose 2004)

36.16 R. Triebel, P. Pfaff, W. Burgard: Multi-level surface maps for outdoor terrain mapping and loop closing, IEEE/RSJ Int. Conf. Intell. Robot. Syst. (2006)

36.17 C. Wellington, A. Courville, A. Stentz: A generative model of terrain for autonomous navigation in vegetation, Int. J. Robot. Res. **25**(12), 1287–1304 (2006)

36.18 P. Pfaff, R. Triebel, W. Burgard: An efficient extension to elevation maps for outdoor terrain mapping and loop closing, Int. J. Robot. Res. **26**(2), 217–230 (2007)

36.19 N. Fairfield, G. Kantor, D. Wettergreen: Real-time SLAM with octree evidence grids for exploration in underwater tunnels, J. Field Robot. **24**(1), 3–21 (2007)

36.20 A. Foessel: Scene Modeling from Motion-Free Radar Sensing. Ph.D. Thesis (Carnegie Mellon University, Pittsburgh 2002)

36.21 J.-F. Lalonde, N. Vandapel, M. Hebert: Data structure for efficient processing in 3-D, Proc. Robot. Sci. Syst. I (2005) p. 48

36.22 J. Leal: Stochastic Environment Representation. Ph.D. Thesis (The University of Sydney, Sydney 2003)

36.23 P. Heckbert, M. Garland: Optimal triangulation and quadric-based surface simplification, J. Comput. Geom. Theory Appl. **14**(1-3), 49–65 (1999)

36.24 A. Akbarzadeh: Towards urban 3d reconstruction from video, Int. Symp. 3D Data Proc. Visualization Transmission (2006)

36.25 C. Frueh, S. Jain, A. Zakhor: Data processing algorithms for generating textured 3d building facade meshes from laser scans and camera images, Int. J. Comput. Vis. **61**(2), 159–184 (2005)

36.26 I. Stamos, P. Allen: Geometry and texture recovery of scenes of large scales, Comput. Vis. Image Underst. **88**, 94–118 (2002)

36.27 D. Gennery: Traversability analysis and path planning for a planetary rover, Auton. Robot. **6**, 131–146 (1999)

36.28 B. Sofman, E. Lin, J. Bagnell, J. Cole, N. Vandapel, A. Stentz: Improving robot navigation through self-supervised online learning, J. Field Robot. **23**(12), 1059–1075 (2006)

36.29 D. Ferguson, A. Stentz: The delayed D* algorithm for efficient path replanning, Proc. IEEE Int. Conf. Robot. Autom. (2005)

36.30 D. Ferguson, A. Stentz: Field D*: An interpolation-based path planner and replanner, Proc. Int. Symp. Robot. Res. (ISRR) (2005)

36.31 M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, S. Thrun: Anytime dynamic a*: An anytime, replanning algorithm, Proc. Int. Conf. Autom. Planning Scheduling (ICAPS) (2005)

36.32 D. Stavens, S. Thrun: A self-supervised terrain roughness estimator for off-road autonomous driving, Uncertainty Artif. Intell. (Boston 2006)

36.33 A. Angelova, L. Matthies, D. Helmick, P. Perona: Slip prediction using visual information, Proc. Robot. Sci. Syst. (Philadelphia 2006)

36.34 D. Kim, J. Sun, S. Oh, J. Rehg, A. Bobick: Traversability classification using unsupervised on-line visual learning for outdoor robot navigation, IEEE Int. Conf. Robot. Autom. (2006)

36.35 S. Thrun, M. Montemerlo, A. Aron: Probabilistic terrain analysis for high-speed desert driving, Robotics Science and System Conference (2005)

36.36 R. Murrieta-Cid, C. Parra, M. Devy: Visual navigation in natural environments: from range and color data to a landmark-based model, Auton. Robot. **13**(2), 143–168 (2002)

36.37 D. Asmar, J. Zelek, S. Abdallah: Tree trunks as landmarks for outdoor vision SLAM, Proc. Conf. Comp. Vision Pattern Recognition Workshop (2006)

36.38 I. Posner, D. Schroeter, P. Newman: Using scene similarity for place labelling, Int. Symp. Exp. Robot. (2006)

36.39 A. Torralba, K.P. Murphy, W.T. Freeman, M.A. Rubin: Context-based vision system for place and object recognition, IEEE Int. Conf. Comput. Vis. (ICCV) (2003)

36.40 D. Bradley, S. Thayer, A. Stentz, P. Rander: Vegetation detection for mobile robot navigation, Tech. Rep. **CMU-RI-TR-04-12**, Robotics Institute (Carnegie Mellon University, Pittsburgh 2004)

36.41 S. Kumar, J. Guivant, H. Durrant-Whyte: Informative representations of unstructured environments, Proc. IEEE Int. Conf. Robot. Autom. (ICRA) (2004)

36.42 S. Kumar, F. Ramos, B. Douillard, M. Ridley, H. Durrant-Whyte: A novel visual perception framework, Proc. 9th Int. Conf. Contr. Autom. Robot. Vision (2006)

36.43 F. Ramos, S. Kumar, B. Upcroft, H. Durrant-Whyte: Representing natural objects in unstructured environments, Neural Inf. Proc. Syst. (NIPS) (2005)

36.44 C. Pantofaru, R. Unnikrishnan, M. Hebert: Toward generating labeled maps from color and range data for robot navigation, Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (2003)

36.45 J.F. Lalonde, N. Vandapel, D. Huber, M. Hebert: Natural terrain classification using three-dimensional ladar data for ground robot mobility, J. Field Robot. **23**(10), 839–861 (2006)

36.46 M. Devy, R. Chatila, P. Fillatreau, S. Lacroix, F. Nashashibi: On autonomous navigation in a natural environment, Robot. Auton. Syst. **16**(1), 5–16 (1995)

36.47 R. Manduchi, A. Castano, A. Talukder, L. Matthies: Obstacle detection and terrain classification for autonomous off-road navigation, Auton. Robot. **18**(1), 81–102 (2005)

36.48 D. Huber, M. Hebert: 3d modeling using a statistical sensor model and stochastic search, Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR) (2003) pp. 858–865

36.49 S. Balakirsky, A. Lacaze: World modeling and behavior generation for autonomous ground vehicles, IEEE Int. Conf. Robot. Autom. (2000)

36.50 A. Lacaze, K. Murphy, M. Delgiorno: Autonomous mobility for the demo III experimental unmanned vehicles, Proc. AUVSI (2002)

36.51 P. Bellutta, R. Manduchi, L. Matthies, K. Owens, A. Rankin: Terrain perception for demo III, Proc. Intell. Vehicles Symp. (2000)

36.52 J.F. Lalonde, R. Unnikrishnan, N. Vandapel, M. Hebert: Scale selection for classification of point-sampled 3-d surfaces, 5th Int. Conf. 3-D Digital Imaging Modeling (3DIM 2005) (2005)

36.53 J. Macedo, R. Manduchi, L. Matthies: Ladar-based discrimination of grass from obstacles for autonomous navigation, Proc. 7th Int. Symp. Exp. Robot. (ISER) (2000)

36.54 D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, A. Ng: Discriminative learning of Markov random fields for segmentation of 3-d scan data, Proc. Conf. Comp. Vision Pattern Recognition (2005)

36.55 R. Triebel, K. Kersting, W. Burgard: Robust 3d scan point classification using associative Markov networks, IEEE Int. Conf. Robot. Autom. (2006)

36.56 H. Chen, P. Meer, D. Tyler: Robust regression for data with multiple structures, IEEE Int. Conf. Comput. Vision Pattern Recognition (2001)

36.57 R. Unnikrishnan, M. Hebert: Robust extraction of multiple structures from non-uniformly sampled data, Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (2003)

36.58 D. Wolf, G. Sukhatme, D. Fox, W. Burgard: Autonomous terrain mapping and classification using hidden Markov models, Proc. IEEE Int. Conf. Robot. Autom. (ICRA) (2005)

36.59 C. Olson, L. Matthies, J. Wright, R. Li, K. Di: Visual terrain mapping for Mars exploration, Comput. Vis. Understand. **105**, 73–85 (2007)

36.60 J. Nieto, J. Guivant, E. Nebot: The hybrid metric maps (hymms): a novel map representation for denseSLAM, IEEE Int. Conf. Robot. Autom. (2004)

36.61 F. Ramos, J. Nieto, H. Durrant-Whyte: Recognising and modelling landmarks to close loops in outdoor SLAM, IEEE Int. Conf. Robot. Autom. (2007)

36.62 D. Hähnel, D. Schulz, W. Burgard: Mobile robot mapping in populated environments, Adv. Robot. **17**(7), 579–598 (2003)

36.63 C.-C. Wang, C. Thorpe, S. Thrun: Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas, Proc. IEEE Int. Conf. Robot. Autom. (ICRA) (2003)

36.64 P. Biber, T. Duckett: Dynamic maps for long-term operation of mobile service robots, Proc. Robot. Sci. Syst. (RSS) (2005)

36.65 C. Stachniss, W. Burgard: Mobile robot mapping and localization in non-static environments, Proc. Nat. Conf. Artif. Intell. (Pittsburgh 2005)

36.66 D. Hähnel, R. Triebel, W. Burgard, S. Thrun: Map building with mobile robots in dynamic environments, Proc. IEEE Int. Conf. Robot. Autom. (ICRA) (2003)

36.67 R. Siegwart, I. Nourbakhsh: *Introduction to Autonomous Mobile Robots* (MIT-Press, Cambridge 2001)

36.68 S. Thrun, W. Burgard, D. Fox: *Probabilistic Robotics* (MIT Press, Cambridge 2005)

36.69 H. Samet: *Foundations of Multidimensional and Metric Data Structures* (Elsevier, Amsterdam 2006)