CHAPTER 9
VISION-BASED POSITIONING

A core problem in robotics is the determination of the position and orientation (often referred to as the pose) of a mobile robot in its environment. The basic principles of landmark-based and map-based positioning also apply to the vision-based positioning or localization which relies on optical sensors in contrast to ultrasound, dead-reckoning and inertial sensors. Common optical sensors include laser-based range finders and photometric cameras using CCD arrays.

Visual sensing provides a tremendous amount of information about a robot's environment, and it is potentially the most powerful source of information among all the sensors used on robots to date. Due to the wealth of information, however, extraction of visual features for positioning is not an easy task.The problem of localization by vision has received considerable attention and many techniques have been suggested. The basic components of the localization process are:

- representations of the environment,
- sensing models, and
- localization algorithms.

Most localization techniques provide absolute or relative position and/or the orientation of sensors. Techniques vary substantially, depending on the sensors, their geometric models, and the representation of the environment.

The geometric information about the environment can be given in the form of landmarks, object models and maps in two or three dimensions. A vision sensor or multiple vision sensors should capture image features or regions that match the landmarks or maps. On the other hand, landmarks, object models, and maps should provide necessary spatial information that is easy to be sensed. When landmarks or maps of an environment are not available, landmark selection and map building should be part of a localization method.

In this chapter, we review vision-based positioning methods which have not been explained in the previous chapters. In a wider sense, “positioning” means finding position and orientation of a sensor or a robot. Since the general framework of landmark-based and map-based positioning, as well as the methods using ultrasound and laser range sensors have been discussed, this chapter focuses on the approaches that use photometric vision sensors, i.e., cameras. We will begin with a brief introduction of a vision sensor model and describe the methods that use landmarks, object models and maps, and the methods for map building.

9.1 Camera Model and Localization

Geometric models of photometric cameras are of critical importance for finding geometric position and orientation of the sensors. The most common model for photometric cameras is the pin-hole camera with perspective projection as shown in Fig. 9.1. Photometric cameras using optical lens can be modeled as a pin-hole camera. The coordinate system (X, Y, Z) is a three-dimensional camera coordinate system, and (x, y) is a sensor (image) coordinate system. A three-dimensional feature in

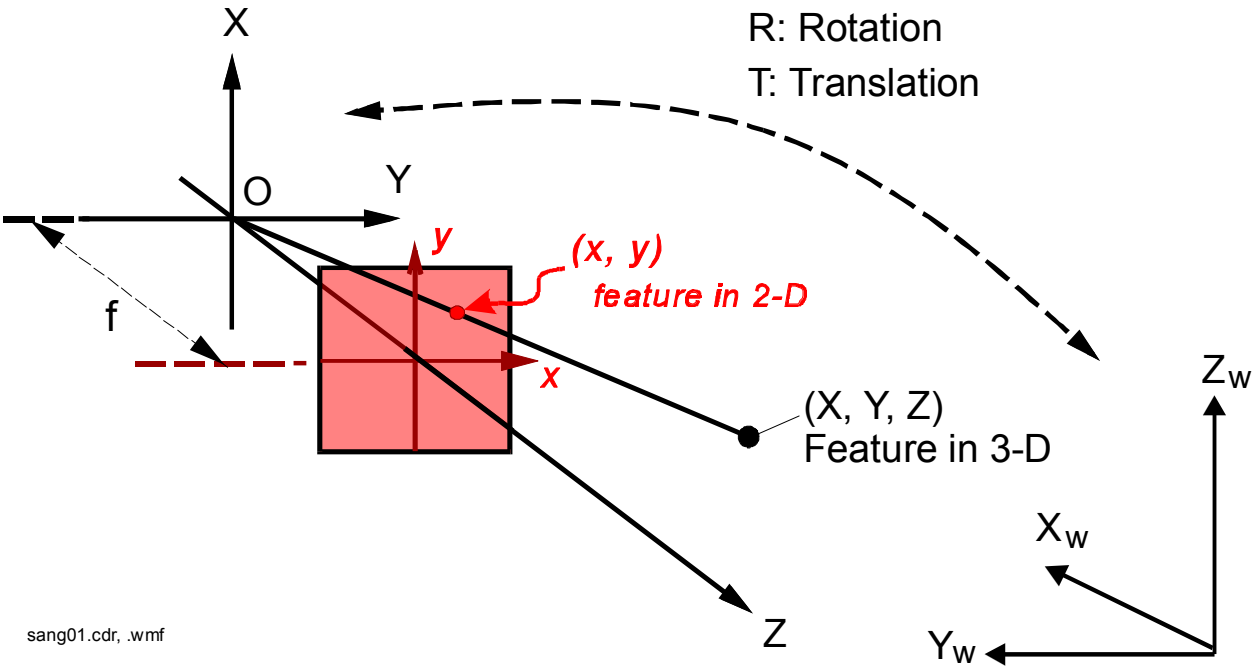

Figure 9.1: Perspective camera model.

an object is projected onto the image plane (x, y). The relationship for this perspective projection is given by

$$x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z} \tag{9.1}$$

Although the range information is collapsed in this projection, the angle or orientation of the object point can be obtained if the focal length f is known and there is no distortion of rays due to lens distortion. The internal parameters of the camera are called intrinsic camera parameters and they include the effective focal length f, the radial lens distortion factor, and the image scanning parameters, which are used for estimating the physical size of the image plane. The orientation and position of the camera coordinate system (X, Y, Z) can be described by six parameters, three for orientation and three for position, and they are called extrinsic camera parameters. They represent the relationship between the camera coordinates (X, Y, Z) and the world or object coordinates (X_W, Y_W, Z_W). Landmarks and maps are usually represented in the world coordinate system.

The problem of localization is to determine the position and orientation of a sensor (or a mobile robot) by matching the sensed visual features in one or more image(s) to the object features provided by landmarks or maps. Obviously a single feature would not provide enough information for position and orientation, so multiple features are required. Depending on the sensors, the sensing schemes, and the representations of the environment, localization techniques vary significantly.

9.2 Landmark-Based Positioning

The representation of the environment can be given in the form of very simple features such as points and lines, more complex patterns, or three-dimensional models of objects and environment. In this section, the approaches based on simple landmark features are discussed.

9.2.1 Two-Dimensional Positioning Using a Single Camera

If a camera is mounted on a mobile robot with its optical axis parallel to the floor and vertical edges of an environment provide landmarks, then the positioning problem becomes two-dimensional. In this case, the vertical edges provide point features and two-dimensional positioning requires identification of three unique features. If the features are uniquely identifiable and their positions are known, then the position and orientation of the pin-hole camera can be uniquely determined as illustrated in Fig. 9.2a. However, it is not always possible to uniquely identify simple features such as points and lines in an image. Vertical lines are not usually identifiable unless a strong constraint is imposed. This is illustrated in Fig. 9.2b.

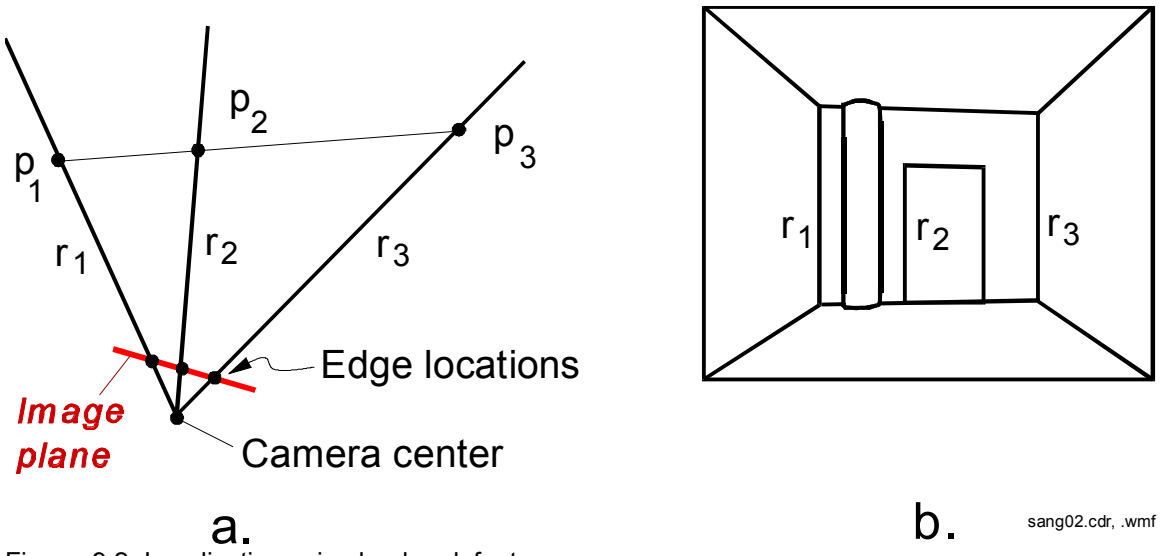

Figure 9.2: Localization using landmark features.

Sugihara [1988] considered two cases of point location problems. In one case the vertical edges are distinguishable from each other, but the exact directions in which the edges are seen are not given. In this case, the order in which the edges appear is given. If there are only two landmark points, the measurement of angles between the corresponding rays restricts the possible camera position to part of a circle as shown in Fig. 9.3a. Three landmark points uniquely determine the camera position which is one of the intersections of the two circles determined by the three mark points as depicted in Fig. 9.3b. The point location algorithm first establishes a correspondence between the three landmark points in the environment and three observed features in an image. Then, the algorithm measures the angles between the rays. To measure the correct angles, the camera should be calibrated for its intrinsic parameters. If there are more than three pairs of rays and landmarks, only the first three pairs are used for localization, while the remaining pairs of rays and landmarks can be used for verification.

In the second case, in which k vertical edges are indistinguishable from each other, the location algorithm finds all the solutions by investigating all the possibilities of correspondences. The algorithm first chooses any four rays, say r_1, r_2, r_3, and r_4. For any ordered quadruplet (p_i, p_j, p_l, p_m) out of n mark points $p_1,...,p_n$, it solves for the position based on the assumption that r_1, r_2, r_3, and r_4 correspond to p_i, p_j, p_l, and p_m, respectively. For n(n-1)(n-2)(n-3) different quadruples, the algorithm can solve for the position in $O(n^4)$ time. Sugihara also proposed an algorithm that runs in $O(n^3 \log n)$ time with O(n) space or in $O(n^3)$ time with $O(n^2)$ space. In the second part of the paper, he considers the case where the marks are distinguishable but the directions of rays are inaccurate. In this case, an estimated position falls in a region instead of a point.

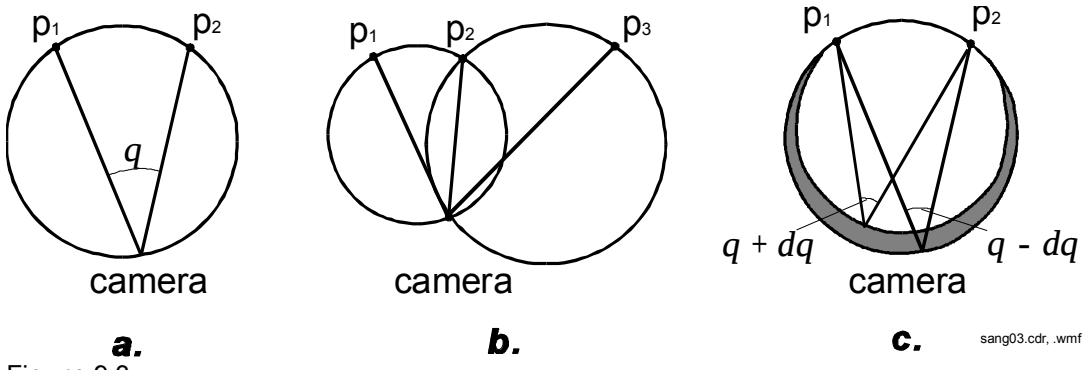

Figure 9.3:
a. Possible camera locations (circular arc) determined by two rays and corresponding mark points.
b. Unique camera position determined by three rays and corresponding mark points.
c. Possible camera locations (shaded region) determined by two noisy rays and corresponding mark points.

(Adapted from [Sugihara 1988; Krotkov 1989]).

Krotkov [1989] followed the approach of Sugihara and formulated the positioning problem as a search in a tree of interpretation (pairing of landmark directions and landmark points). He developed an algorithm to search the tree efficiently and to determine the solution positions, taking into account errors in the landmark direction angle. According to his analysis, if the error in angle measurement is at most $\delta\theta$, then the possible camera location lies not on an arc of a circle, but in the shaded region shown in Fig. 3c. This region is bounded by two circular arcs.

Krotkov presented simulation results and analyses for the worst-case errors and probabilistic errors in ray angle measurements. The conclusions from the simulation results are:

- the number of solution positions computed by his algorithm depends significantly on the number of angular observations and the observation uncertainty $\delta\theta$.
- The distribution of solution errors is approximately a Gaussian whose variance is a function of $\delta\theta$ for all the angular observation errors he used: a. uniform, b. normal, and c. the worst-case distribution.

Betke and Gurvits [1994] proposed an algorithm for robot positioning based on ray angle measurements using a single camera. Chenavier and Crowley [1992] added an odometric sensor to landmark-based ray measurements and used an extended Kalman filter for combining vision and odometric information.

9.2.2 Two-Dimensional Positioning Using Stereo Cameras

Hager and Atiya [1993] developed a method that uses a stereo pair of cameras to determine correspondence between observed landmarks and a pre-loaded map, and to estimate the two-dimensional location of the sensor from the correspondence. Landmarks are derived from vertical edges. By using two cameras for stereo range imaging the algorithm can determine the two-dimensional locations of observed points — in contrast to the ray angles used by single-camera approaches.

Hager and Atiya's algorithm performs localization by recognizing ambiguous sets of correspondences between all the possible triplets of map points p_i, p_j, p_k and those of observed points o_a, o_b, o_c. It achieves this by transforming both observed data and stored map points into a representation that is invariant to translation and rotation, and directly comparing observed and stored entities. The permissible range of triangle parameters due to sensor distortion and noise is computed and taken into account.

For n map points and m observed points, the off-line initialization stage consumes $O(n^3 \log n)$ time to compute and sort all triangle parameters from the map points. At run time, the worst case complexity is $O(m^3 (n^3 + \log n))$. However, an efficient strategy of marking and scanning reduces the search space and real-time performance (half a second) is demonstrated for five observed and 40 stored landmarks.

9.3 Camera-Calibration Approaches

The camera-calibration approaches are more complex than the two-dimensional localization algorithms discussed earlier. This is because calibration procedures compute the intrinsic and extrinsic camera parameters from a set of multiple features provided by landmarks. Their aim is to establish the three-dimensional position and orientation of a camera with respect to a reference coordinate system. The intrinsic camera parameters include the effective focal length, the lens distortion parameters, and the parameters for image sensor size. The computed extrinsic parameters provide three-dimensional position and orientation information of a camera coordinate system relative to the object or world coordinate system where the features are represented.

The camera calibration is a complex problem because of these difficulties:

- All the intrinsic and extrinsic parameters should be computed from the two-dimensional projections of a limited number of feature points,
- the parameters are inter-related, and
- the formulation is non-linear due to the perspectivity of the pin-hole camera model.

The relationship between the three-dimensional camera coordinate system (see Fig. 1)

$$\underline{X} = [X, Y, Z]^T \tag{9.2}$$

and the object coordinate system

$$\underline{X}_W = [X_W, Y_W, Z_W]^T \tag{9.3}$$

is given by a rigid body transformation

$$\underline{X} = \underline{R}\underline{X}_W + \underline{T} \tag{9.4}$$

where

$$\underline{R} = \begin{bmatrix} r_{XX} & r_{XY} & r_{XZ} \\ r_{YX} & r_{YY} & r_{YZ} \\ r_{ZX} & r_{ZY} & r_{ZZ} \end{bmatrix}, \quad \underline{T} = \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix} \tag{9.5}$$

are the rotation and translation matrices, respectively.

Determination of camera position and orientation from many image features has been a classic problem of photogrammetry and has been investigated extensively [Slama 1980; Wolf 1983]. Some photogrammetry methods (as described in [Wolf 1983]) solved for the translation and rotation parameters by nonlinear least-squares techniques. Early work in computer vision includes that by Fischler and Bolles [1981] and Ganapathy [1984]. Fischler and Bolles found the solution by first computing the length of rays between the camera center (point O in Fig. 9.1) and the feature projections on the image plane (x, y). They also established results on the number of solutions for various number of feature points. According to their analysis, at least six points are required to get a unique solution. Ganapathy [1984] showed similar results and presented somewhat simplified algorithms.

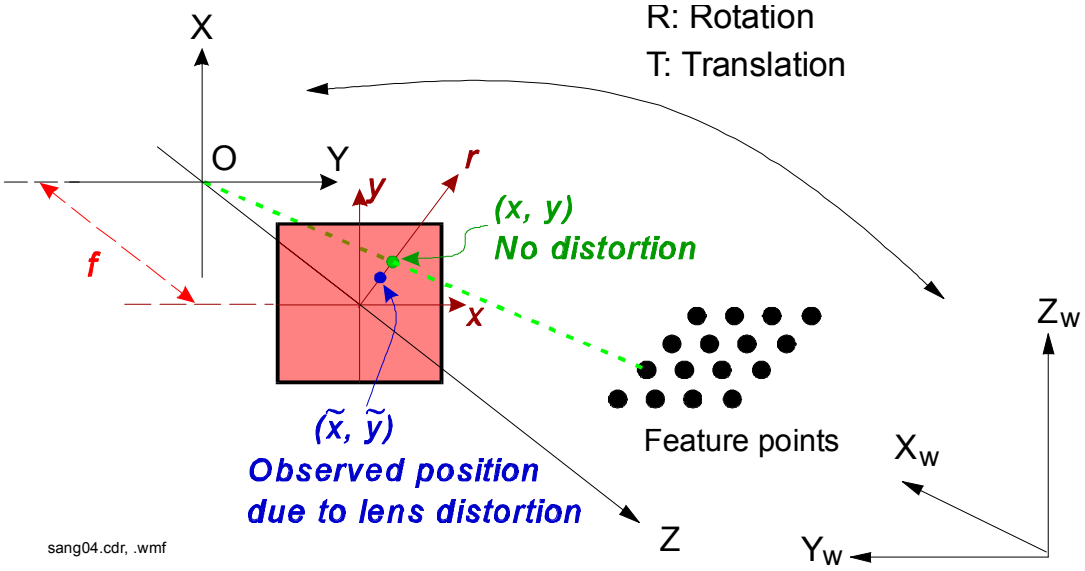

Figure 9.4: Camera calibration using multiple features and a radial alignment constraint.

More recently several newer methods were proposed for solving for camera position and orientation parameters. The calibration technique proposed by Tsai [1986] is probably the most complete and best known method, and many versions of implementation code are available in the public domain. The Tsai's algorithm decomposes the solution for 12 parameters (nine for rotation and three for translation) into multiple stages by introducing a constraint. The radial alignment constraint

assumes that the lens distortion occurs only in the radial direction from the optical axis Z of the camera. Using this constraint, six parameters r_{XX}, r_{XY}, r_{YX}, r_{YY}, t_X, and t_Y are computed first, and the constraint of the rigid body transformation $\underline{R}\underline{R}^T = \underline{I}$ is used to compute r_{XZ}, r_{YZ}, r_{ZX}, r_{ZY}, and r_{ZZ}. Among the remaining parameters, the effective focal length f and t_Z are first computed neglecting the radial lens distortion parameter κ, and then used for estimating κ by a nonlinear optimization procedure. The values of f and t_Z are also updated as a result of the optimization. Further work on camera calibration has been done by Lenz and Tsai [1988].

Liu et al. [1990] first suggested the use of straight lines and points as features for estimating extrinsic camera parameters. Line features are usually abundant in indoor and some outdoor environments and less sensitive to noise than point features. The constraint used for the algorithms is that a three-dimensional line in the camera coordinate system (X, Y, Z) should lie in the plane formed by the projected two-dimensional line in the image plane and the optical center O in Fig 9.1. This constraint is used for computing nine rotation parameters separately from three translation parameters. They present linear and nonlinear algorithms for solutions. According to Liu et al.'s analysis, eight-line or six-point correspondences are required for the linear method, and three-line or three-point correspondences are required for the nonlinear method. A separate linear method for translation parameters requires three-line or two-point correspondences.

Haralick et al. [1989] reported their comprehensive investigation for position estimation from two-dimensional and three-dimensional model features and two-dimensional and three-dimensional sensed features. Other approaches based on different formulations and solutions include Kumar [1988], Yuan [1989], and Chen [1991].

9.4 Model-Based Approaches

A priori information about an environment can be given in more comprehensive form than features such as two-dimensional or three-dimensional models of environment structure and digital elevation maps (DEM). The geometric models often include three-dimensional models of buildings, indoor structure and floor maps. For localization, the two-dimensional visual observations should capture the features of the environment that can be matched to the preloaded model with minimum uncertainty. Figure 5 illustrates the match between models and image features. The problem is that the two-dimensional observations and the three-dimensional world models are in different forms. This is basically the problem of object recognition in computer vision: (1) identifying objects and (2) estimating pose from the identified objects.

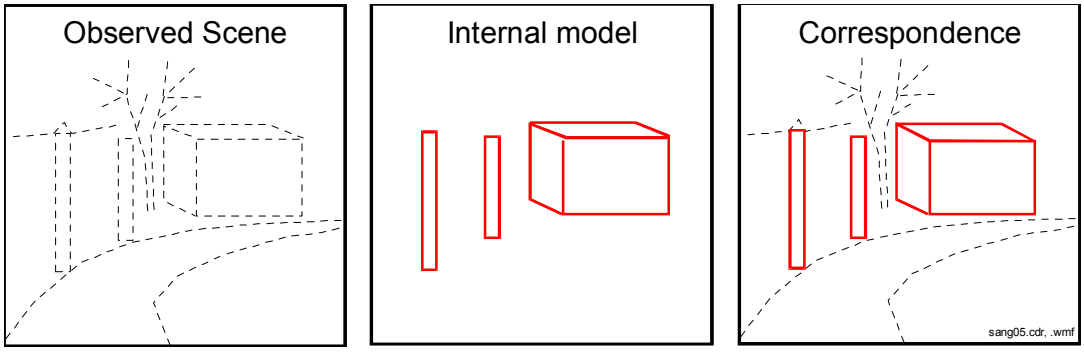

Figure 9.5: Finding correspondence between an internal model and an observed scene.

9.4.1 Three-Dimensional Geometric Model-Based Positioning

Fennema et al. [1990] outlined a system for navigating a robot in a partially modeled environment. The system is able to predict the results of its actions by an internal model of its environment and models of its actions. Sensing is used to correct the model's predictions about current location or to progress towards some goal. Motions are composed in a hierarchical planner that sketches overall paths and details the short term path. Control of the robot is broken down into the action level, the plan level, and the goal level. Landmarks are chosen to measure progress in the plan. The system must receive perceptual confirmation that a step in the plan has been completed before it will move to the next part of the plan. Later steps in a plan expand in detail as earlier steps are completed. The environment is modeled in a graph structure of connected nodes called locales. Locales may exist at a variety of scales in different hierarchies of the map. Other information is kept in the system associated with each locale to provide more detail. Using these models the robot operates in a plan- and monitor-cycle, confirming and refining plans to achieve overall goals.

The algorithm by Fennema et al. [1990] matches images to the map by first matching the two-dimensional projection of landmarks to lines extracted from the image. The best fit minimizes the difference between the model and the lines in the data. Once the correspondence between model and two-dimensional image is found, the relation of the robot to the world coordinate system must be found. This relation is expressed as the rotation and translation that will match the robot- and world-systems. Matching is done by considering all possible sets of three landmarks. Once a close correspondence is found between data and map, the new data is used to find a new estimate for the actual pose.

Kak et al. [1990] used their robot's encoders to estimate its position and heading. The approximate position is used to generate a two-dimensional scene from their three-dimensional world model and the features in the generated scene are matched against those extracted from the observed image. This method of image matching provides higher accuracy in position estimation.

Talluri and Aggarwal [1991; 1992] reported their extensive work on model-based positioning. They use three-dimensional building models as a world model and a tree search is used to establish a set of consistent correspondences. Talluri and Aggarwal [1993] wrote a good summary of their algorithms as well as an extensive survey of some other vision-based positioning algorithms.

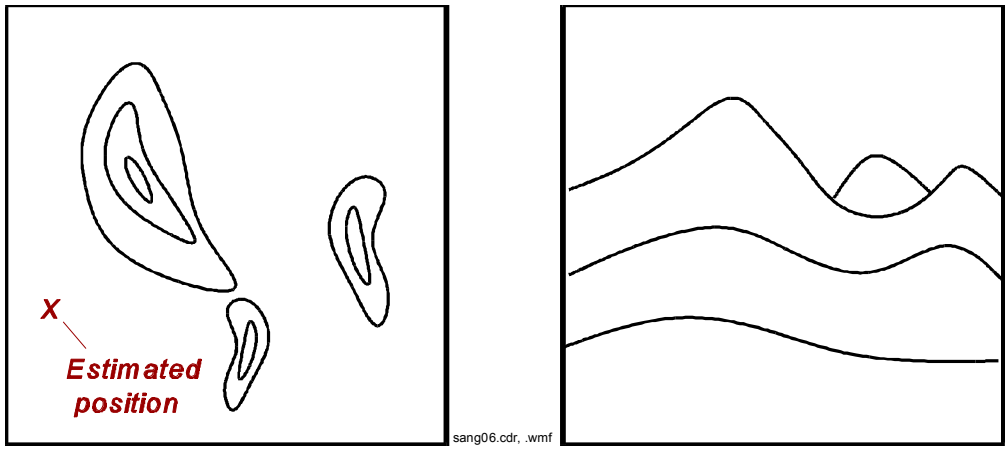

Figure 9.6: Finding a location on a digital elevation maps (DEM) that matches a visual scene observed from a point. The 'x' marks a possible location in the DEM that could generate the observed visual scene to the right.

9.4.2 Digital Elevation Map-Based Localization

For outdoor positioning, Thompson et al. [1993] developed a hierarchical system that compares features extracted from a visual scene to features extracted from a digital elevation maps (DEM). A number of identifiable features such as peaks, saddles, junctions, and endpoints are extracted from the observed scene. Similarly, features like contours and ridges are extracted from the DEM. The objective of the system is to match the features from the scene onto a location in the map. The feature matching module interacts with each feature extractor as well as with a geometric inference module. Each module may request information and respond to the others. Hypotheses are generated and tested by the interaction of these feature extractors, geometric inference, and feature matching modules.

In order to make matching more tractable, configurations of distinctive and easily identified features are matched first. Using a group of features cuts down dramatically on the number of possible comparisons. Using rare and easily spotted features is obviously advantageous to making an efficient match. A number of inference strategies that express viewpoint constraints are consulted in the geometric inference module. These viewpoint constraints are intersected as more features are considered, narrowing the regions of possible robot location.

Sutherland [1993] presented work on identifying particular landmarks for good localization. A function weighs configurations of landmarks for how useful they will be. It considers the resulting area of uncertainty for projected points as well as relative elevation. Sutherland showed that a careful choice of landmarks usually leads to improved localization.

Talluri and Aggarwal [1990] formulated position estimation using DEM as a constrained search problem. They determined an expected image based on a hypothetical location and compared that to the actual observed image. Possible correspondences are eliminated based on geometric constraints between world model features and their projected images. A summary of their work is given in [Talluri and Aggarwal, 1993].

9.5 Feature-Based Visual Map Building

The positioning methods described above use a priori information about the environment in the form of landmarks, object models or maps. Sometimes pre-loaded maps and absolute references for positions can be impractical since the robot's navigation is restricted to known structured environments. When there is no a priori information, a robot can rely only on the information obtained by its sensors.

The general framework for map-building has been discussed in the previous chapter. For constructing the environment model, vision systems usually use image features detected at one or more robot positions. According to the computer vision theory of structure from motion and stereo vision, correct correspondences of image features detected in several locations can provide information about the motion of the sensor (both translation and rotation), as well as of the three-dimensional structure of the environment at the feature locations. The trajectory of the sensor can be obtained by visual dead-reckoning, i.e., the integration of the estimated incremental motion. This is illustrated in Fig. 9.7.

The object features detected in a sensor location become the relative reference for the subsequent sensor locations. When correspondences are correctly established, vision methods can provide higher

accuracy in position estimation than odometry or inertial navigation systems. On the other hand, odometry and inertial sensors provide reliable position information up to certain degree and this can assist the establishment of correspondence by limiting the search space for feature matching. A visual map based on object features is a sparse description of environment structure.

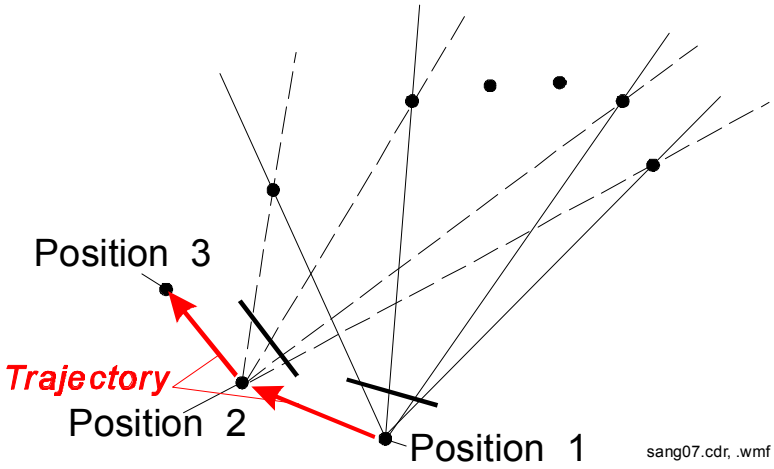

Figure 9.7: Illustration of map building and trajectory integration.

Moravec [1981] used stereo cameras with variable baseline for obtaining environment structure in the form of feature locations and estimating position of the sensors. A feature selection method was suggested and coarse-to-fine correlation feature matching was used. The suggested error measure is that the uncertainty in feature location is proportional to the distance from the sensor.

Matthies and Shafer [1987] proposed a more systematic and effective error measure using a three-dimensional Gaussian distribution. A Kalman filter was used for updating robot positions based on the Gaussian error distribution of detected features.

Ayache and Faugeras [1987] used trinocular stereo and three-dimensional line features for building, registering and fusing noise visual maps. They used an extended Kalman filter for combining measurements obtained at different locations.

9.6 Summary and Discussion

We reviewed some of the localization methods based only on photometric camera sensors. These methods use:

- landmarks
- object models
- maps
- feature-based map-building

Most of the work discussed suggests methodologies that relate detected image features to object features in an environment. Although the vision-based techniques can be combined with the methods using dead-reckoning, inertial sensors, ultrasonic and laser-based sensors through sensor fusion, tested methods under realistic conditions are still scarce.

Similar to the landmark-based and map-based methods that were introduced in the previous chapters, vision-based positioning is still in the stage of active research. It is directly related to most of the computer vision methods, especially object recognition which involves identification of object class and pose estimation from the identified object. As the research in many areas of computer vision and image processing progresses, the results can be applied to vision-based positioning. In addition to object recognition, relevant areas include structure from stereo, motion and contour, vision sensor modeling, and low-level image processing. There are many vision techniques that are potentially useful but have not been specifically applied to mobile robot positioning problems and tested under realistic conditions.