Submitted to IROS, EPFL, Switzerland, 2002

Vision for Mobile Robot Localization in Urban Environments

Atanas Georgiev, Peter K. Allen,
Computer Science Department, Columbia University, New York, NY*

Abstract

This paper addresses the problem of mobile robot localization in urban environments. Typically, GPS is the preferred sensor for outdoor operation. However, using GPS-only localization methods leads to significant performance degradation in urban areas where tall nearby structures obstruct the clear view of the satellites. In our work, we use vision-based techniques to supplement GPS and odometry and provide accurate localization. The vision system identifies prominent linear features in the scene and matches them with a reduced model of nearby buildings, yielding improved pose estimation of the robot.

1 Introduction

The problem of accurate localization is fundamental to mobile robotics. A mobile robot's ability to correctly estimate its current pose is essential to its successful autonomous operation. Without a good sense of position and orientation, key navigation tasks, such as path planning and motion control, are impossible to perform and inevitably result in the robot getting lost. On a higher level, applications, such as environmental modeling, surveying, or transportation, will produce unusable or even undesired results.

A very popular way to address the problem of outdoor localization is by using GPS. GPS-based systems are attractive because they provide very accurate global location measurements and are becoming affordable. Using GPS in urban areas, however, poses a significant challenge. Tall buildings in the vicinity tend to obstruct the clear view of the sky. The signals of fewer satellites reach the receiver which results in unstable or even no estimates at all. The signal-to-noise ratio could be attenuated by trees or large structures standing in the way. Very difficult to deal with are signal reflections and the multipath phenomenon.

Our experience confirms the observations above. We built an urban site modeling robot, called *AVENUE*, which localizes itself by using GPS and odometry [2, 9]. Our tests showed that while it performed well in open areas, GPS failed to provide accurate positioning at many locations, such as between tall buildings. The conclusion was that, although GPS is very useful, it alone can not provide adequate coverage in a highly-urbanized area. Additional sensors are needed.

We have now expanded our system with vision. As we have seen, GPS performs well in open areas; it is around buildings where it fails. The knowledge of having buildings in the vicinity allows us to exploit their typical characteristics, such as horizontal and vertical principal directions and abundance of parallel lines. These features are easily captured by a camera and their linear nature facilitates the difficult and computationally expensive task of image processing.

In this paper, we address the limitations of a pure GPS-based localization system. Our focus here is on improving the overall performance in areas where GPS fails. The proposed method consists of the integration of GPS and odometry with vision, and the utilization of a simple and compact model of the working environment. After a brief discussion of the related work in the next section, our method is described in detail and experimental results are presented.

2 Related Work

GPS is typically used in combination with inertial sensors and proper filtering techniques. A good example of this strategy with a focus on fault detection has been shown by Sukkarieh et al [16]. Another typical example is the autonomous mower built by Aono et al [3] whose accuracy the authors estimate to be $0.2\,m$ based on accurate GPS data and simulating noise with standard deviation of $1\,m$.

Various methods for camera pose estimation have been adapted to robot localization. Some of them make assumptions about the environment that are not easily met outdoors (e.g. constant illumination).

*This work was supported in part by an ONR/DARPA MURI award ONR N00014-95-1-0601 and NSF grants EIA-97-29844 and ANI-0099184.

Appearance-based methods need extensive training sets and huge storage requirements [12, 17]. Others require closely following previously traversed paths [19]. A good systematic approach to recovering the relative poses of multiple cameras in urban environments can be found in [1].

Despite the strong interest in the use of GPS and cameras for mobile robot localization, there does not seem to be much work on the integration of these two sensors. Kotani et al built a system using GPS, vision, and a fiber optic gyro for localization [10], however, they use GPS to only establish the initial pose of the robot. Chen and Shibasaki have also observed the problems with using GPS in urban sites and have addressed them by supplementing GPS with additional sensors, including a camera [5]. In addition, their solution requires the availability of a comprehensive geodetic information system.

Various other approaches to mobile robot localization have been proposed and are being investigated. Among them are the idea of simultaneous localization and map building [4, 7, 11, 18], the probabilistic approaches [13, 18], and Monte Carlo localization [6].

An advantage of our approach is that it makes selective use of the camera and, thus, avoids wasting precious CPU power on image processing when GPS and odometry perform well on their own. Further, it does not require modifications of the working environment. It uses a simple environmental model that is already available by the site-modeling application it coexists with. This provides opportunities to actively seek the best portion of the environment to image and process.

3 Overview of the method

The work presented here is a part of *AVENUE*[1] — a large project to produce an automated system for 3D geometric and photometric modeling of urban sites [2]. Our hardware platform is an ATRV-2 robot equipped with a number of sensors, including a real-time kinematic GPS, a color CCD camera mounted on a pan-tilt unit, and a laser range finder (Figure 1).

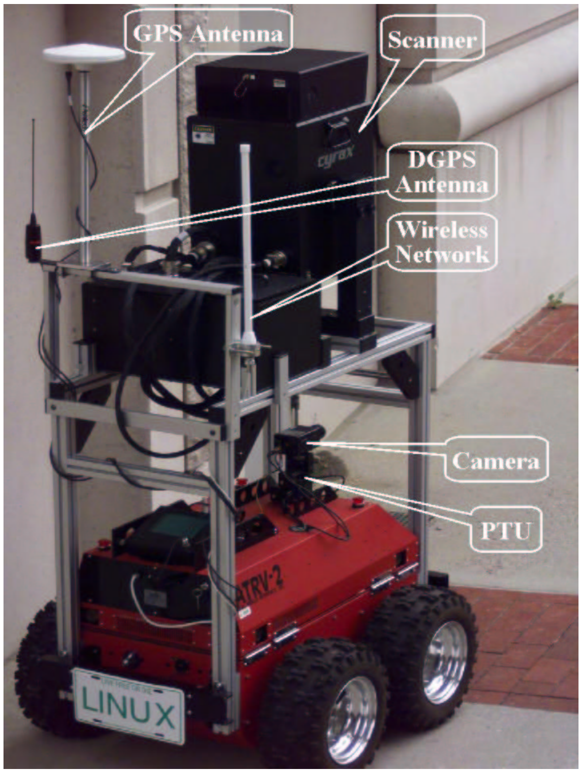

Figure 1: Mobile robot for automated site modeling

The robot's task is to go to desired locations and acquire 3D range scans and images of selected buildings. The locations are determined by our view planning system and are used by the path planning system to compute a good trajectory which the robot then follows [2]. When the robot arrives at a chosen location, it acquires the requested scans and images and hands them over to the 3D modeling system which registers them and incorporates them into the model of the site [14, 15]. After that, the view planning system determines what portions of the site are not yet modeled and decides upon the next best data acquisition location. The process starts from a certain location and gradually expands the area it has covered until a complete model of the site is obtained.

In order to follow the desired trajectories and position itself accurately at the target locations, the robot needs to have a precise estimate of its pose at all times. As stated in the introduction, the combination of odometry and high-accuracy GPS works sufficiently well in open areas. Thus, we need to employ image-based pose estimation only in proximity of buildings. This also means that we can use the buildings — specifically, their abundance of linear features — as cues to our visual localization. Further, our camera is better suited for localization than our scanner because of the scanner's slow acquisition speed (15–20 min), the amount of data it returns (1 milion points), and the lack of control due to a closed proprietary interface.

Given the above considerations, the robot uses GPS and odometry most of the time since they both provide

[1] AVENUE stands for Autonomous Vehicle for Exploration and Navigation in Urban Environments

frequent updates and require minimal computational power. Their estimates are tagged with a confidence factor of their accuracy based on the discrepancies between their readings and the plant model of the robot's motion [9]. If this confidence is sufficiently high, we accept the result and bypass the image processing step, thus saving time and computational resources. Otherwise, we are likely close to a building that is causing the degraded GPS accuracy and we attempt our image-based pose estimation algorithm.

4 Visual Pose Estimation

Our visual pose estimation is based on matching an image of a building taken by the camera with a model of that building. The model consists of linear segments which are both abundant in a typical urban landscape and easy to detect and process using 2-D image operators. We use a separate model for each building's facade and store all models in a data base. A view of the models of our test area is shown in Figure 4.

We should point out that the models we use for localization are not the detailed full-featured models that are ultimately built by AVENUE but ones that are of very low complexity and are easy to create from relatively few key measurements, even manuallly. In our case, we are also able to obtain reduced-complexity models from available full-featured ones.

When visual pose estimation is attempted, we still have a rough estimate of the robot's location from recent accurate GPS data and odometry. We use this rough estimate to search our data base for the best model to use. Models outside of the working range ($10-30\,m$) or viewed at a very low angle (< 30 deg) are eliminated from consideration. The rest are sorted by their euclidian distance and the closest one is picked. Then, we turn the camera toward that building's facade and take a snapshot.

At this stage, we have an image of the facade and a model of it and we need to determine the pose of the robot. Since the pose of the camera is tracked by a pan-tilt unit rigidly affixed to the robot, if we find the pose of the camera, we can easily derive the pose of the robot. Thus our focus in this section is the computation of the camera pose.

We do this by using matching features, specifically, linear segments in the image and the model. The 3D linear features are explicitly represented in the model so the first step is to find their 2D counterparts. We apply a Canny edge detector to locate edge pixels and then use the incremental line fitting technique to connect them in straight line segments. Only the longest few of the line segments are retained.

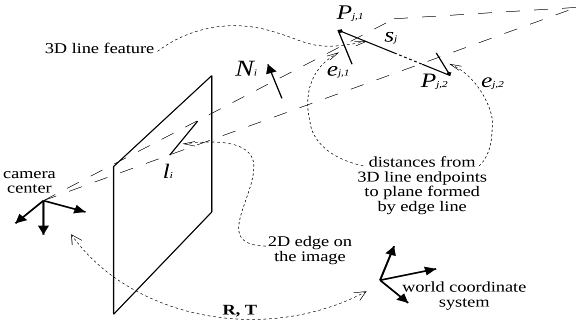

Figure 2: Error metric used for pose estimation

The difficulty in the next step comes from the well-known data association problem. We need to correctly match a subset of the edge segments from the image with the 3D line segments that we have in our model. A brute-force approach is not feasible because of its extreme computational requirements. Instead, we have adapted the RANSAC paradigm which has proven very efficient in solving matching problems [8]. The basic idea is to solve the pose estimation problem a number of times using randomly chosen matches between a minimum number of 2D and 3D line segments. In our case we pick four pairs and compute an estimate for the camera pose. This is done by minimizing an error function that quantifies the displacement of a 3D line segment from the plane passing through the center of projection of the camera and its matching 2D edge segment [14]. Specifically, if N_i is the normal of the plane formed by the i-th edge segment and the camera center of projection, and R and T are the rotation and translation that align the world coordinate system with the one of the camera, then

$$d_{i,j} = (N_i \cdot (R(P_{j,1}) + T))^2 + (N_i \cdot (R(P_{j,2}) + T))^2 \quad (1)$$

gives us the sum of squared distances of the end points $P_{j,1}$ and $P_{j,2}$ of the j-th 3D line segment to that plane (Figure 2). The error function that we minimize is the sum of $d_{i,j}$ for the four matching pairs.

Next, we need to determine the consensus set, i.e. all matching pairs of 2D edge segments from the image and 3D line segments from the model that agree with the computed pose. To do this, we need a proximity measure that tells us how "close" a 3D line segment and a 2D edge segment are from the perspective of the current camera pose. Using the metric in equation 1 here is not a good idea because it only measures how well the 2D line and its 3D match are aligned. Two line segments can be perfectly aligned (collinear) but

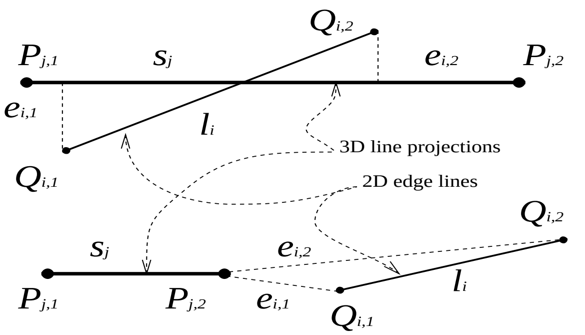

Figure 3: Distance metric used for matching.

still far apart in the direction of their orientation. We use the computed R and T to project all 3D lines on the image and perform the matching in 2D space. The metric that we use is the sum of squared distances from each end of the projected 2D edge to the closest point on the 3D line segment projection (as opposed to the infinite line). That is, if we have an edge line l_i with end points $Q_{i,1}$ and $Q_{i,2}$ and the projection of a 3D line segment, s_j, the metric is

$$d_{i,j} = dist(Q_{i,1}, s_j)^2 + dist(Q_{i,2}, s_j)^2 \qquad (2)$$

where $dist(Q, s)$ is the distance from the point Q to the closest point P on the line segment (Figure 3).

When the 2D line edge does not extend much past the 3D line's projection (Figure 3, top), this metric is the same as the "alignment" metric above. However, when the two lines are mostly collinear but far from each other (Figure 3, bottom), the metric will return a reasonably high distance.

For each 3D line segment on the model, we search in a neighborhood of its projection on the image for 2D edges and compute their distance according to this metric. The 2D edge with the smallest distance is taken to be the match, if that distance is less than a threshold. If no such 2D edge is found, then the 3D line segment is assumed to have no match.

The consensus set consists of all matches found. If it contains all but very few lines from the model (which might be occluded or simply not detected) and the total error is less than a threshold, we have found a good pose candidate. A sanity check is done whether this new pose is within the expected error from the estimate of the other sensors and, if it is, the new pose is accepted and the random sampling process is terminated. Otherwise, we continue with the next sample until a good match is found or a certain number of iterations are performed.

In a typical RANSAC implementation, a certain probability of success is decided upon and then the number of required iterations is computed to guarantee success with that probability. However, this can only be done if the probability for a given match being correct is known. In our case, this is extremely difficult to estimate, especially when anomalies like occlusions and misdetections need to be considered. Instead of relying on an imprecise heuristic for the number of iterations, we run the process for an allotted amount of time and if no good solution is found within this period, the robot repeats the process a little farther along its route with a new set of images.

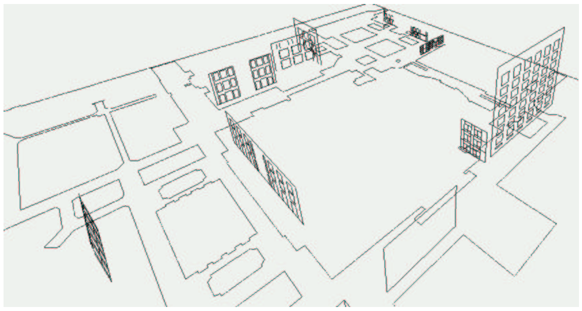

Figure 4: Our 3-D models used for localization shown on a 2-D map of the test area.

5 Experiments

To test the accuracy of our method, we performed two kinds of tests: one that compares the result for each test location with ground truth data, and another, that compares the two results the algorithm produced on two different images taken from the same location.

In both kinds of tests, we wanted to measure the quality of the location estimation alone and minimize the interference from inaccuracies in the model. Thus we took care to create accurate models of the buildings we imaged by scanning their prominent features with a high-quality electronic theodolite with nominal accuracy of 2 *mm*. The features we modeled were windows, ledges and decorations — all commonly found and abundant in urban structures and easy to find using 2D image operators (Fig. 4).

We drove the robot to a number of locations in our test area and at each location we took an image with the robot's camera of a modeled face of a nearby building. We chose locations at which we have previously had problems receiving stable GPS data. A sketch of the test area with the test locations and directions in which the images were taken is shown in Figure 5.

The first test consisted of 6 images taken at locations 1 through 5 (two images were taken at location 5). The input and the output of the localization system for

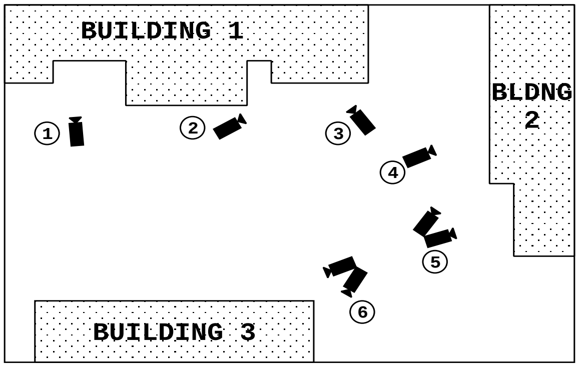

Figure 5: A map of the area where the experiments were conducted showing camera locations and orientations.

each run are illustrated in Figures 6 and 7 (top). The left image in each pair shows the model used projected onto the image using the initial inaccurate estimate of the camera pose (that comes from GPS, odometry, or as a guess). The image to the right shows the model projected on the image after the camera pose was computed. In all cases the alignment of the model and the image is very accurate. The resulting errors in translation for these six runs were 0.306, 0.148, 0.369, 0.186, 0.147, and 0.211 m respectively. The distance to the buildings were $10-30\,m$. These errors are comparable to what many accurate GPS systems provide in practice and are acceptable for our kind of outdoor mobile robot application.

The purpose of the second test was to confirm that the algorithm does not generate contradictory results when used on different facades from the same location. We took a pair of images of two faces of the same building at locations 5 and 6 by simply panning and tilting the camera. We processed both pairs of images with their corresponding models (Figure 7) and the errors in translation were 0.064 and 0.290 m — again within reasonable expectations for mobile robot navigation.

In these tests, we focused primarily on the accuracy of the location estimates and not so much on the orientation. This is partially because it is difficult to obtain reliable ground truth for the orientation. It is obvious, however, from the resulting alignment of model and image shown in the figures, that the camera orientation was recovered correctly and more than adequately for robot navigation.

6 Summary and Future Work

We have described our approach to mobile robot localization in outdoor urban environments. This method is a part of a larger project aiming the automation of the process of building accurate photo-realistic and geometrically correct models of urban sites. The system depends on GPS and computer vision to compensate for the long-term unreliability of the robot odometry. Our image-based solution makes use of data and assumptions that are already present in the context for which the entire mobile robot application is designed. No environmental modifications are necessary. A simple database of models of building faces is available that allows us to actively decide where to point the camera when taking an image.

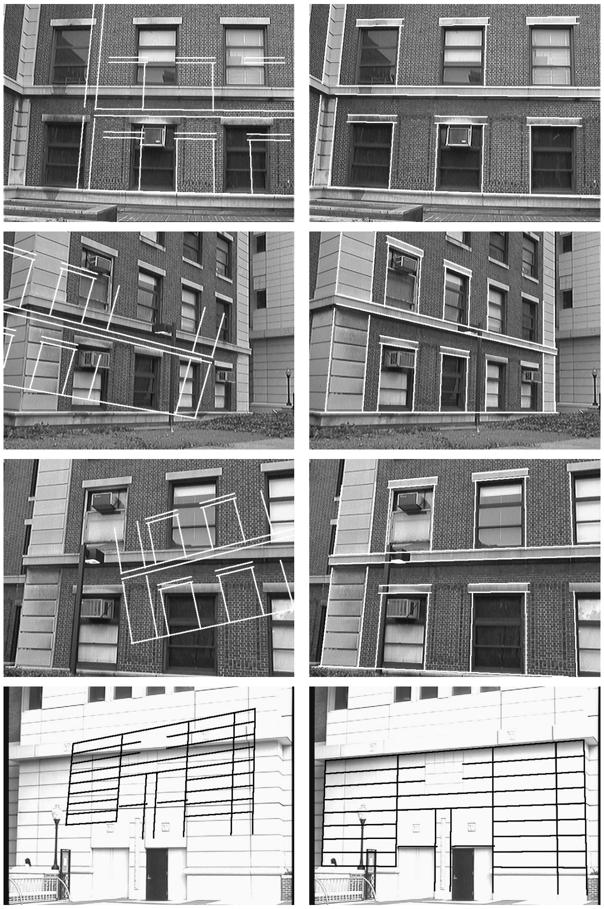

Figure 6: Pose estimation at locations 1 through 4: Each pair shows the model projected on the image using the initial pose of the camera (left), and the resulting pose of the camera (right) for the corresponding test location.

Currently, our system works with a single “best” building’s face even if a number of alternatives exist. We are extending it to use all visible modeled buildings in the vicinity in a single pose estimation step. The matching part extends trivially, except that we need to take care to match lines in an image with ones in the corresponding model only. The consensus set will con-

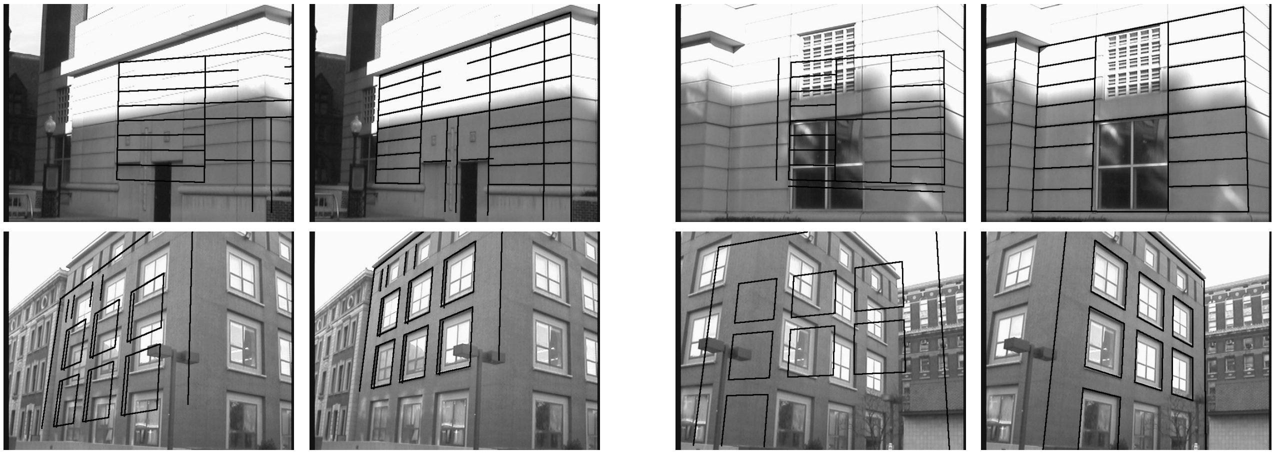

Figure 7: Initial and final alignments in the pose estimation tests with a pair of images taken from the same location.

sist of all matching segments across all image-model pairs. Optimizing across multiple images/models will result in higher accuracies and improved reliability.

An interesting research direction that we would like to pursue is an improved integration of GPS, odometry, and vision. Pose estimates from vision are typically not isotropic and it can be beneficial to utilize whatever approximate estimates are available from other sensors. This is very important in situations when GPS produces inaccurate but usable data.

References

[1] The MIT City Scanning Project. http://city.lcs.mit.edu.

[2] P. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer. AVENUE: Automated site modeling in urban environments. In *3rd Int. Conf. on Digital Imaging and Modeling, Quebec City*, pages 357–364, May 2001.

[3] T. Aono, K. Fujii, S. Hatsumoto, and T. Kamiya. Positioning of vehicle on undulating ground using GPS and dead reckoning. In *IEEE ICRA*, pages 3443–3448, 1998.

[4] J. A. Castellanos, J. M. Martinez, J. Neira, and J. D. Tardos. Simultaneous map building and localization for mobile robots: A multisensor fusion approach. In *IEEE ICRA*, pages 1244–1249, 1998.

[5] T. Chen and R. Shibasaki. High precision navigation for 3-D mobile GIS in urban area by integrating GPS, gyro and image sequence analysis. In *Proceedings of the International Workshop on Urban 3D/Multi-media Mapping, Tokyo, Japan*, pages 147–156, 1999.

[6] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *IEEE ICRA*, pages 1322–1328, 1999.

[7] H. Durrant-Whyte, M. Dissanayake, and P. Gibbens. Toward deployment of large scale simultaneous localization and map building (SLAM) systems. In *Proc. of Int. Simp. on Robotics Research*, pages 121–127, 1999.

[8] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In *DARPA*, pages 71–88, 1980.

[9] A. Gueorguiev, P. K. Allen, E. Gold, and P. Blaer. Design, control and architecture of a mobile site-modeling robot. In *IEEE ICRA*, pages 3266–3271, 2000.

[10] S. Kotani, K. Kaneko, T. Shinoda, and H. Mori. Mobile robot navigation based on vision and DGPS information. In *IEEE ICRA*, pages 2524–2529, 1998.

[11] J. Leonard and H. J. S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *Proc. of Int. Simp. on Robotics Research*, pages 128–135, 1999.

[12] R. Sim and G. Dudek. Mobile robot localization from learned landmarks. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, October 1998.

[13] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *IJCAI*, pages 1080–1087, 1995.

[14] I. Stamos and P. K. Allen. Integration of range and image sensing for photorealistic 3D modeling. In *IEEE ICRA*, pages II:1435–1440, 2000.

[15] I. Stamos and P. K. Allen. Registration of 3D and 2D imagery in urban environments. In *International Conference on Computer Vision*, Vancouver, Canada, 2001.

[16] S. Sukkarieh, E. Nebot, and H. F. Durrant-Whyte. Achieving integrity in an INS/GPS navigation loop for autonomous land vehicle applications. In *IEEE ICRA*, pages 3437–3442, 1998.

[17] S. Thrun. Finding landmarks for mobile robot navigation. In *IEEE ICRA*, pages 958–963, 1998.

[18] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5:253–271, 1998.

[19] J. Y. Zheng and S. Tsuji. Panoramic representation for route recognition by a mobile robot. *International Journal of Computer Vision*, 9(1):55–76, 1992.