

Probabilistic Self-Localization for Mobile Robots

Clark F. Olson, *Member, IEEE*

Abstract—Localization is a critical issue in mobile robotics. If a robot does not know where it is, it cannot effectively plan movements, locate objects, or reach goals. In this paper, we describe probabilistic self-localization techniques for mobile robots that are based on the principle of maximum-likelihood estimation. The basic method is to compare a map generated at the current robot position with a previously generated map of the environment in order to probabilistically maximize the agreement between the maps. This method is able to operate in both indoor and outdoor environments using either discrete features or an occupancy grid to represent the world map. The map may be generated using any method to detect features in the robot's surroundings, including vision, sonar, and laser range-finder. We perform an efficient global search of the pose space that guarantees that the best position is found according to the probabilistic map agreement measure in a discretized pose space. In addition, subpixel localization and uncertainty estimation are performed by fitting the likelihood function with a parameterized surface. We describe the application of these techniques in several experiments, including experimental localization results for the Sojourner Mars rover.

Index Terms—Maximum-likelihood estimation, mobile robotics, self-localization, uncertainty estimation.

I. INTRODUCTION

Mobile robots must have some method by which to determine their position with respect to known locations in the environment in order to navigate effectively and achieve goals. This is called the *localization problem*. The most common and basic method for performing localization is through dead-reckoning. This technique integrates the velocity history of the robot over time to determine the change in position from the starting location (see, for example, [1] and [2]). Unfortunately, pure dead-reckoning methods are prone to errors that grow without bound over time, so some additional method is necessary to periodically correct the robot position. It is common to combine the additional localization technique, such as triangulation from landmarks or map matching, with dead-reckoning using an extended Kalman filter to probabilistically update the robot position.

In this paper, we describe a technique that performs localization infrequently to update the position of the robot. In order to perform localization, we compare a map generated using the robot's sensors at the current position (the *local*

map) to a previously generated map of the environment (the *global map*), which may be constructed as the robot explores. The maps are compared according to a maximum-likelihood similarity measure. The best relative position between the maps according to the similarity measure (although not always the correct position) is found using a branch-and-bound search of the robot pose space. This method does not require an initial estimate of the robot position to yield good results, only bounds on the search space, which may be of any size. In addition, this technique is general; it can be used with either a discrete landmark-based map representation or an occupancy grid model of the environment. We have primarily explored the application of these techniques to three-dimensional occupancy grids in order to model unstructured outdoor terrain.

The measure that we use to compare the maps is derived from previous work on image matching using the Hausdorff distance [3]. We have reformulated this measure in terms of maximum-likelihood estimation. In this measure, the likelihood of each position is computed as the product of the likelihoods of the distances from the features in the local map to the closest features in the global map, with an additional term representing the prior probability of the position. This probabilistic measure avoids the drawbacks of the original matching measure, which include a sharp boundary between good and poor feature matches and the inability to incorporate probabilistic information, while retaining the advantages, which include robustness to outliers and a global search technique [4]. Our approach allows subpixel localization in discretized pose spaces and accurate estimation of the uncertainty in the localization by fitting the likelihood function with a parameterized surface. This combination of techniques yields a localization method that achieves accurate and robust global localization in unstructured terrain through the use of dense three-dimensional data (although we are not restricted to this data), since the map matching measure is very robust to outliers and distracting data.

These techniques can be viewed as a variant of the Markov localization method [5]–[10], since we formulate the problem with a likelihood function over the space of possible robot positions. A key contribution of our approach is the development of a probabilistic map matching measure that is robust to map errors, sufficiently general to apply to virtually any metric map representation and powerful enough to yield accurate localization in complex and unstructured environments. Additional contributions include a method to search the space of robot positions efficiently to locate the most likely position(s) and techniques to accurately estimate the error in the localization.

The strategy that we use to locate the best position is a hierarchical divide-and-conquer algorithm over the space of possible robot positions (the *pose space*) that has been recently used for matching image edge maps [11]–[13]. We first test the position

Manuscript received January 20, 1999; revised July 19, 1999. This paper was recommended for publication by Associate Editor G. Hager and Editor A. De Luca upon evaluation of the reviewers' comments. This work was supported in part by NASA Space Telerobotics Program, through the JPL Robotics and Mars Exploration Technology Office and by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

The author is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109-8099 USA (e-mail: clark.olson@jpl.nasa.gov).

Publisher Item Identifier S 1042-296X(00)03023-8.

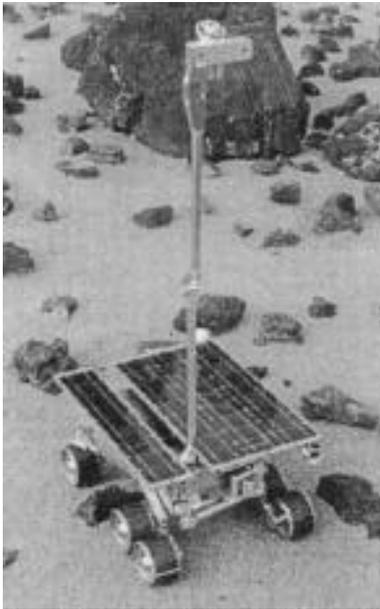


Fig. 1. Rocky 7 Mars rover prototype in the JPL Mars Yard with mast deployed.

given by dead-reckoning so that we have an initial position to compare against. The pose space is then divided into rectilinear cells. For each cell in the space, we attempt to prove that the cell cannot contain a position that is superior to the best one that has been found so far using an efficient bounding mechanism. For any cell that cannot be pruned, the cell is divided into smaller cells and the process is repeated recursively. The process stops dividing the cells when they have become small enough to represent valid hypotheses or by some other robust stopping criterion.

Our motivation for this research is the long-range science rover project at JPL, which has developed the Rocky 7 Mars rover prototype [14]. Mars rovers require increased self-localization ability in order to perform with greater autonomy from both operators on Earth and from the lander bringing the rover to Mars. For example, the Sojourner Mars rover was limited to moving short distances during each downlink cycle due to positional uncertainty and it could not venture far from the lander. The method by which dead-reckoning errors were corrected for Sojourner was through a human operator overlaying a model of the rover on stereo range data that was computed from downlinked imagery of the rover taken by the lander [15].

The techniques described here are effective whenever a map can be generated in the robot's local coordinate frame and we have a map of the same terrain in the frame of reference in which we wish to localize the robot. We can thus use rover imagery, either from the close-to-the-ground body-mounted cameras or from a rover mast such as the one on Rocky 7 (see Fig. 1) to generate the local map. The global map might also be created from the rover imagery, but it could also be generated using imagery from the lander (including imagery taken during the lander's descent to the surface), and it is possible that orbital imagery could be used, although we will not have orbital imagery of Mars with sufficient resolution to use for rover localization with submeter precision in the near future [16].

The localization techniques described here are very useful in the context of a Mars mission. While operating in a small area containing several science targets (such as the area around the lander that Sojourner operated in), we may perform localization using the panoramic imagery generated at the center of the area as our global map. While this is not crucial when the lander can see the rover, the next-generation Mars rover will venture away from the lander and it will be equipped with a mast with stereo cameras that will allow it to generate panoramic imagery of the terrain. This allows localization to be performed by matching the panoramic terrain maps generated using the mast imagery to maps generated from either the body-mounted cameras, if possible, or by using the mast to image interesting terrain, if necessary. Our approach can also be used on traverses between sites by performing localization at some interval in order to update the position of the rover [17].

We have tested our method using real and synthetic data. The synthetic experiments model a case where the robot performs localization using a discrete set of known landmarks in the environment. These experiments indicate that accurate localization can be achieved by searching a discretized pose space through the use of subpixel estimation, and that the uncertainty in the localization can be accurately estimated by fitting the surface of the likelihood function. Our application of this method to real data creates an occupancy grid representation of the terrain using stereo vision [18], since we are concerned primarily with performing localization in natural terrain. Experiments using the terrain maps generated from stereo vision have been performed with both terrestrial data, acquired in the JPL Mars Yard using the Rocky 7 research rover [14], and imagery of Mars acquired by the Mars Pathfinder lander and Sojourner rover [15]. The experiments using Mars imagery validate the use of these techniques to perform autonomous localization for Mars rovers without the need to downlink information to Earth.

In Section II, we review previous work on robot localization, focusing on techniques that perform map matching in order to localize the robot. Section III describes the probabilistic map similarity measure that we use to determine which positions of the robot are the most likely to be correct. Section IV gives an algorithm for searching the space of possible robot positions to locate the position that maximizes this map similarity measure. Section IV also discusses the application of this method to robot localization from discrete landmarks and by comparing occupancy. The techniques by which we perform subpixel estimation in discretized pose spaces and estimate the uncertainty in the localization process are given in Section V. The experimental results that we have achieved with real and synthetic data are described in Section VI. Section VII discusses the strengths and weaknesses of the algorithm and compares the method to other localization methods. Finally, Section VIII summarizes the paper and gives some concluding remarks. We note that portions of this research have been presented at recent conferences [19]–[21].

II. PREVIOUS WORK

Mobile robot localization is typically performed by combining the results of dead-reckoning with some periodic sensor-based localization technique using, for example, an extended Kalman filter. Many techniques have been used to

provide the periodic sensor-based localization. Often these techniques operate by determining correspondences between a set of sensed features (such as landmarks) and a known map of the environment. The known positions of the features in the map, together with the sensed positions relative to the robot, allow the robot's current position to be determined. We give a brief review of several such techniques here.

Some sensor-based techniques operate frequently, so that the robot moves a small amount between localization steps. This simplifies the problem, since the determination of the feature correspondences can be treated as a tracking problem, rather than searching the map for the features. The drawback to this formulation is that the techniques must operate frequently enough to prevent the tracker from losing track of the features. If the tracker makes a mistake by specifying an incorrect correspondence, it may have a drastic effect on the localization result. Examples of sensor-based techniques that operate frequently and combine results using the Kalman filter include [22]–[26].

Many other methods have been proposed that do not require frequent sensor measurements. One such method is to locate nearby landmarks and to perform a triangulation procedure to determine the position of the robot. Sugihara [27] addressed the problem where the relative directions of the landmarks can be sensed, but not the distance to the landmarks. He developed an algorithm for performing localization from this data in $O(n^3)$ time, where n is the number of landmarks. Sugihara's extensions of this method yielded an $O(n^2 \log n)$ algorithm for a robot with a compass and an $O(n^2)$ algorithm for the case where the landmarks are distinguishable. Betke and Gurvits [28] further consider the case where the landmarks are distinguishable. By representing the landmark positions as complex numbers, they obtained a linear time algorithm with a least-squares error criterion.

Another localization technique uses a search tree [29], [30] to perform matching between the features or landmarks detected by a sensor and the known map. Drumheller [31] used this technique to perform localization using walls detected by sonar. He incorporated a *sonar barrier test* to check for inconsistencies based on the constraints of sonar data. Simsarian *et al.* [32] described a variation of this technique where the map is decomposed into *view-invariant regions*, which are used to guide the tree search and reduce the cost of feature matching. Talluri and Aggarwal [33] similarly match line segments in the plane. They have used a world model in which the regions of the pose space from which the same set of obstacle boundaries are visible are computed. To compute the robot pose, they used a Hough transform variation to limit the number of regions that must be examined. A feature correspondence search was then performed for each of the possible regions to determine the best match.

Cox [34] also performed matching between line segments in the plane using a laser range-finder to detect the line segments corresponding to building walls. However, Cox assumed that the robot would have rough knowledge of its location and thus used an iterative least-squares fitting procedure to improve the position estimation. Lu and Milios [35] apply a similar technique for map matching to the case where the local and global maps are represented by the set of sensed points, rather than extracted line

segments. An alternative method for localization with such data was explored by Crowley *et al.* [36]. They extract an eigenspace from a large set of range scans of the environment from various robot positions. Localization is performed by matching the maps in the extracted subspace.

Elfes [37] used an occupancy grid representation of the environment. Each cell in the grid was given a score between -1 and 1 , where -1 represents unoccupied, 1 represents occupied, and values in between represent varying levels of certainty. Localization was performed by locating the position between a local and a global occupancy grid that maximized the product of the values at the corresponding cells in the grids.

Atiya and Hager [38] addressed the problem where the landmarks are two-dimensional points in a plane. Correspondences were determined by matching triples of sensed landmarks to triples of map landmarks, since such triples yield lengths and angles that are invariant to the robot position. Uncertainties in the localization estimate were computed by intersecting the uncertainty regions of the landmarks, which were approximated by rectilinear cells.

A technique that has been used for coarse localization in a large environment is to examine the features present on the horizon and to use some strategy to match them to a known elevation map of the terrain. Talluri and Aggarwal [39] use the shape of the horizon line to search for the position of a robot in a digital elevation map. They first perform pruning using geometrical constraints to eliminate many positions in a discretized space of possible robot positions. For positions that pass the first stage, a refinement step is used that performs curve matching between the visible horizon and the estimated horizon line computed from the elevation map. The best match is taken to be the most likely robot position. Stein and Medioni [40] approximate the horizon line by a polygonal chain and index a table storing subsections of the horizon as it would be seen from each position in a discretized pose space on the map. A verification step for the indexed matches uses geometric constraints to select the best match. Thompson *et al.* [41] extract and match features on the horizon and other visible hills and ridges. Matches between configurations of features are then searched for in a map that has been preprocessed. The hypothesized locations are then refined and evaluated. Cozman and Krotkov [42] also detect mountain peaks on the horizon. They perform the search in a discretized space of positions using table look-up in order to maximize the posterior probability of finding the correct position.

Several probabilistic localization methods have recently been explored that maintain a probability distribution over the possible robot positions. The Markov localization paradigm, which is used by several researchers [5]–[10], computes a probability distribution over the space of possible robot positions. When the robot moves, the probability distribution is updated to take into account the additional uncertainty in the robot position that is induced by dead-reckoning errors. When the robot senses the environment, the distribution is updated to take into account the new data, thus reducing the uncertainty in the robot position. Methods based on the extended Kalman filter [22]–[26] can be considered a special case of this method, where the probability distribution is constrained to be a normal distribution.

Nourbakhsh *et al.* [5] use a partially observable Markov model to perform localization without metric information. Simmons and Koenig [6] combine a partially observable Markov model with an evidence grid in order to perform localization with both topological and metric information. Thrun *et al.* [10] use an expectation-maximization algorithm to perform simultaneous localization and map building according to a maximum-likelihood measure. Fox *et al.* [9] give an active localization method using the Markov localization technique.

The final technique that we mention is to perform localization by matching a three-dimensional map of the terrain near the robot to a previously generated map. This is the approach that Kweon and Kanade [43] take in order to generate a terrain map by fusing multiple local maps. They first generate a terrain map from stereo vision using the *locus method*. They then perform matching between the maps in a two-stage procedure. First, an estimate for the relative position is generated by extracting and matching map features (high curvature points). The estimate is then refined using an iterative optimization procedure. Szeliski [44] and Zhang [45] also describe techniques that can be used for matching 3-D terrain maps. Szeliski interpolates a surface from sparse range data and determines the transformation that makes it the most likely that a new set of points arise from the same surface. The optimal transformation is determined using gradient descent search. Zhang's technique uses an initial estimate of the relative position between two sets of points to iteratively improve the estimated position. At each iteration, the technique determines the closest match for each point and updates the estimated position based on a least-squares metric, with some modifications to increase robustness.

III. MAP SIMILARITY MEASURE

We perform localization by matching a map generated at the current robot position (the *local map*) to a previously generated map of the environment (the *global map*), which may be generated by combining previous local maps. The optimal position of the robot with respect to the global map is located using a maximum-likelihood similarity measure for comparing images and maps [4]. This similarity measure (which is described below in more detail) yields a score for each possible position of the local map with respect to the global map by computing a function of the distance from each of the features in the local map to the closest feature in the global map. When an appropriate function is used, such that sensor uncertainty and the possibility of missing a feature is modeled, the measure is robust to outliers, noise, and occluded locations. In addition, it can be applied to either maps consisting of sparse landmarks or to a dense occupancy map representation.

In order to formulate the map matching problem in terms of maximum likelihood estimation, we must have some set of measurements that are a function of the robot position. We use the distances from the visible features at the current robot location to the closest features in the global representation of the environment. The method by which these distances are computed is problem dependent. We have used the Euclidean distance for both landmarks and occupancy maps, but more complex distance functions, such as the Mahalanobis distance can be used

given the requisite covariance information. Since we search for the best relative position between these maps, these distances are variables.

Let us say that our local map \mathcal{L} consists of n features $\{l_1, \dots, l_n\}$ and that our global map \mathcal{G} consists of m features $\{g_1, \dots, g_m\}$. These features may represent discrete landmarks or they may represent cells in an occupancy grid. The distance between a feature l_i in the local map and a feature g_j in the global map, when the local map is at position X with respect to the global map, is denoted $d_{ij}^X = \text{dist}(X(l_i), g_j)$, where the position, X , can be thought of as a function that transforms features in the local map into their corresponding position in the global map. The distance from a feature in the local map to the closest feature in the global map (at some relative position X between the maps) is called D_i^X

$$D_i^X = \min_{1 \leq j \leq m} d_{ij}^X. \quad (1)$$

While these distances are not pairwise independent, we have found that modeling them as such yields good results. Recent work on determining the probability of a false positive for matching sparse features (such as landmarks) [46], [47] and for matching dense features (such as edge maps and occupancy grids) [12], [48] has also achieved accurate results when treating the features independently. We thus formulate the likelihood function for the robot position X as the product of prior probability of the position with the probability distributions of these distances

$$L(X) = p(X) \prod_{i=1}^n p(D_i^X). \quad (2)$$

Note that normalizing the likelihood function such that it integrates to unity over the space of possible positions is not necessary for maximum-likelihood estimation, but a scale factor is necessary if we wish for $L(X)$ to be a probability distribution. For convenience, we work in the $\ln L(X)$ domain, since this does not change the relative ordering of the positions

$$\ln L(X) = \ln p(X) + \sum_{i=1}^n \ln p(D_i^X). \quad (3)$$

In our implementation, we take the position yielding the maximum likelihood to be the position of the robot. However, in environments where multiple positions appear to be similar, it is advisable to retain more than one position or even some representation of the entire likelihood surface. The prior probability distribution of robot positions and the probability density function (PDF) that is used for each feature $p(D_i^X)$ together determine the matching measure that is used between the maps. If nothing is known about the prior distribution of model positions, then it can be modeled by a constant and removed from the measure. On the other hand, if we are tracking the robot position over time (e.g., with an extended Kalman filter), we will have some known prior (a normal distribution in the case of the extended Kalman filter) and this will affect the computed position of the robot.

We must also estimate the PDF of the feature distances. Accurate localization results can be achieved through the use of a PDF that models the sensor uncertainty [4]. The feature localization errors can often be accurately modeled by a normal distribution. However, this does not allow for outliers in the local feature map, which have no corresponding features in the global map. The use of a normal distribution with a constant additive term yields an accurate model for cases with outliers [21]

$$p(D_i^X) = k_1 + \frac{k_2}{\sigma\sqrt{2\pi}} e^{-(D_i^X)^2/2\sigma^2}. \quad (4)$$

In this PDF, σ is the standard deviation of the feature uncertainty and k_1 and k_2 are constants that vary with the frequency of outliers, the density of the maps, and the probability of missing a feature. The robot localization is insensitive to the settings of these constants, but a discussion of the values these constants should take can be found in [21]. It should be noted that (4) is not a probability distribution, since it does not integrate to unity. This is unavoidable if we wish to use a robust measure that does not become arbitrarily close to zero for large values of D_i^X . The use of a function that does not integrate to unity does not affect the accuracy of our results in any significant way.

IV. FINDING THE MOST LIKELY POSITION

Now that a similarity measure between the maps has been defined, we must discuss how the position that optimizes the similarity measure is determined. A simple hill-climbing technique could be used, but such a method would require a good initial estimate of the position of the robot, which is not always available, particularly if we exercise the localization techniques infrequently. We describe a method to search a bounded pose space using a variation of branch-and-bound search that guarantees that we locate the optimal position (according to the similarity measure) in a discretized version of the search space. A subsequent subpixel localization step is performed to gain precision. Following the general discussion of the search strategy, we discuss the application of this search strategy to maps consisting of landmarks and occupancy grids in more detail.

A. Search Strategy

We locate the most likely robot position by adapting a multiresolution search strategy that has been applied to image matching using the Hausdorff distance [11]–[13]. We first test the nominal position of the robot given by dead-reckoning (or any other position, if no initial estimate is available) so that we have an initial position and likelihood to compare against. Next, the pose space is divided into rectilinear cells. Each cell is tested using a conservative test to determine whether it could contain a position that is better than the best position found so far (or any threshold, in general). Cells that cannot be pruned are divided into smaller cells, which are examined recursively (see Fig. 2). When cells of a certain (small) size are reached, the cells are tested explicitly. For example, when we compare occupancy grids under translation, there is a natural discretization of the pose space such that neighboring positions move the maps by one grid cell with respect to each other.

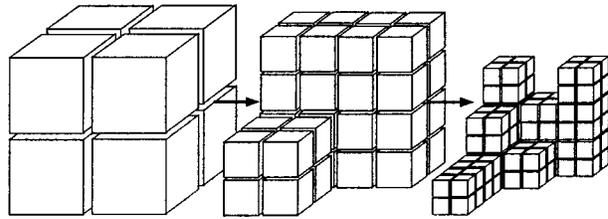


Fig. 2. A search strategy is used that recursively divides and prunes cells of the search space.

For this case, we stop dividing the cells when they contain a single position in the discretization and we then test this position explicitly. For more complex examples, we may set some threshold on the minimum cell size and test the center of the cell when the cell size is below the threshold. Subpixel localization estimates are useful in increasing the precision of the localization in both cases (see Section V).

The key to this strategy is a quick method to test the cells. A cell is allowed to pass the test if it does not contain a good pose, but it should never prune a cell that could contain a good pose, since this could result in the best position being missed. To determine whether a particular cell C could contain a pose that is superior to the best one found so far, we examine the pose c at the center of the cell. In order to place a bound on the best position within the cell, we compute the maximum distance between the locations to which a feature in the local map is transformed into the global map by c and by any other pose in the cell. Denote this distance Δ_C . This allows us to determine the quality of the robot position represented by the center of the cell and then compute a larger value using Δ_C to place an upper bound on the quality of any position in the entire cell.

If we treat robot poses as functions that transform positions in the local map into positions in the global map, then Δ_C can be written

$$\Delta_C = \max_{p \in C} \max_{l \in \mathcal{L}} \|p(l) - c(l)\|. \quad (5)$$

For the space of translations, Δ_C is simply the distance from the center of the cell to any corner of the cell, since the difference in the translated location of any feature in the local map for any two translations is simply the difference between the translations. When rotations are considered, Δ_C is also a function of the local map. In this case, Δ_C can be computed as a function of maximum orientation change between the center of the cell and the corners of the cell. While we concentrate on translations of the robot, since the robot orientation can be determined through other sensors, further discussion of techniques to handle rotations in such a branch-and-bound search strategy can be found elsewhere [12], [13].

To place a bound on the quality of any position within the cell, we bound each of the distances that can be achieved by features in the local map over the cell. This is done by subtracting the maximum change in distance over the cell (Δ_C) from the distance achieved at the center of the cell D_i^c

$$D_i^C = \max(D_i^c - \Delta_C, 0). \quad (6)$$

The values obtained are then propagated through the likelihood function to bound the score that can be achieved by any position in the cell

$$P_i^C = \ln p(D_i^C). \quad (7)$$

P_i^C is the now maximum score that the i th feature of the local map can contribute to the likelihood for any position in the cell.¹

A bound on the best overall likelihood that can be found at a position in the cell is given by

$$\max_{X \in C} L(X) \leq \sum_{i=1}^n P_i^C. \quad (8)$$

If this likelihood does not surpass the best that we have found so far, then we can prune the entire cell from the search. Otherwise, the cell is divided into two cells of the same size by slicing it along the longest axis and the process is repeated recursively until all of the cells have been exhausted. Since we place conservative bounds on the maximum likelihood that can be achieved by any cell that is pruned, this method is guaranteed to locate the position with the largest likelihood in a discretized pose space. It is likely that the discretization of the pose space will cause the computed robot position to be slightly suboptimal with respect to the full continuous space. However, we can examine a fine discretization without adding much computation due to the pruning techniques, and we use subpixel localization methods to further improve the localization estimate.

Our method for pruning cells does not provide tight bounds on the likelihood that can be achieved by each cell and thus it is possible that more pruning could be performed with additional computation at each cell. Our strategy has been to make the processing of each cell as fast as possible, rather than to optimize the number of cells that are examined. Researchers on similar problems in computer vision have taken an approach where tighter bounds are sought at the expense of additional overhead [3], [13].

We note that for cases where more than one qualitative position in the robot's pose space has a significant likelihood, the search strategy can easily be modified to detect all positions meeting some minimum likelihood. In environments where many locations look similar (for example, in an office building), we could store a representation of the entire distribution until the position has been disambiguated. For example, a quadtree (or higher dimensional extension) could be used such that the cells are represented at the resolution at which they can be pruned by the search.

B. Occupancy Grids

The search strategy described above is well suited to matching maps that are represented by occupancy grids, since these are inherently discretized. The space of translations of the robot can be discretized at the same resolution as the maps and this yields a natural resolution of the search space at which to end the recursive division of the cells. We consider the case of binary occupancy grids. This allows for a fast implementation

¹This assumes that the PDF is monotonically nonincreasing, which is true for any reasonable PDF, since we desire closer matches to yield higher scores.

of the search, since each D_i^C can be computed efficiently over the entire global map by computing the distance transform of the map. Extensions of our method can be applied to nonbinary occupancy grids.

In order to implement this procedure efficiently, we first compute the distance transform of the global occupancy map. The distance transform measures the distance from each cell in a discretized map to the closest occupied cell [49], and can be computed efficiently using an algorithm that is linear in the size of the map² [51], [52]. We next compute a relative index into the distance transform for each occupied cell in the local map. The pose-space cell hierarchy is searched using a depth-first search strategy. For each cell that is examined, we loop through the pre-computed indexes into the distance transform (which must be offset by the position of the center of the cell). For each index, we get a distance in the global map. We then use (6)–(8) to determine whether the cell can be pruned.

C. Landmarks

This approach can also be applied to matching maps consisting of geometric landmarks. For example, in indoor environments, we may be able to detect and locate vertical edges, or we may use the peaks of rocks or other landmarks in outdoor terrain. In this case, we can use efficient nearest-neighbor searching techniques to compute each D_i^C exactly. For example, we may use the method of Lipton and Tarjan [53] or Bentley [54] to locate the nearest landmark, if the landmarks are represented by points, and the distances can then be computed directly.

These techniques can be made even more efficient, at the cost of a small amount of precision, by discretizing the landmark positions. In this case, the distances can be computed using the distance transform of the map, as described above. We can then use subpixel localization techniques to improve the precision over the position yielded by the discretized search space (see Section V). Our experiments have indicated that the loss of precision is quite small when using this technique.

Once the method of computing each D_i^C is determined, the remainder of the search strategy is the same as described above.

V. SUBPIXEL LOCALIZATION AND UNCERTAINTY ESTIMATION

Using this probabilistic formulation of the localization problem, we can estimate the uncertainty in the localization in terms of both the variance of the estimated positions and the probability that a qualitative failure occurred. In addition, we can perform subpixel localization in the discretized pose space by fitting a surface to the peak that occurs at the most likely robot position. This need not be limited to the single highest peak found in the pose space. Multiple possible robot positions can be considered if their likelihood is sufficiently large.

A. Subpixel Localization

Let us take as an assumption that the likelihood function approximates a normal distribution in the neighborhood around a

²This assumes that the map is defined on a convex domain [50]. Since we must deal with map errors (and search over the pose space), we always consider a convex map. This ensures that a distance value is obtained for any feature at any map position in the pose space.

peak. Fitting such a normal distribution to the computed likelihoods yields both an estimated variance in the localization estimate and a subpixel estimate of the peak location. Since the likelihood function measures the probability that each position is the actual robot position, the uncertainty in the localization is measured by the rate at which the likelihood function falls off from the peak. Our experimental results confirm that very accurate results can be achieved with this normal approximation.

Now, since we actually perform our computations in the domain of the natural logarithm of the likelihood function, we must fit these values with a polynomial of order 2. We compute the uncertainties in x and y independently. While this is not necessary, it simplifies the presentation

$$\begin{aligned} \ln L(x, y) &\approx \ln \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}\right) \\ &= -\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2} + \ln \frac{1}{2\pi\sigma_x\sigma_y}. \end{aligned} \quad (9)$$

In order to estimate the parameters that we are interested in (x_0 , y_0 , σ_x , and σ_y), we project this polynomial onto the lines $x = x_0$ and $y = y_0$ yielding

$$L(x, y_0) = -\frac{(x-x_0)^2}{2\sigma_x^2} + \ln \frac{1}{2\pi\sigma_x\sigma_y} \quad (10)$$

and

$$L(x_0, y) = -\frac{(y-y_0)^2}{2\sigma_y^2} + \ln \frac{1}{2\pi\sigma_x\sigma_y}. \quad (11)$$

We now fit these equations to the x and y cross sections of the likelihood function at the location of the peak. If the peak in the discretized search space occurs at position (x_p, y_p) , we fit $L(x, y_0)$ to the values at the surrounding five positions along $y = y_p$

$$p_{-2} = L(x_p - 2, y_p) \quad (12)$$

$$p_{-1} = L(x_p - 1, y_p) \quad (13)$$

$$p_0 = L(x_p, y_p) \quad (14)$$

$$p_1 = L(x_p + 1, y_p) \quad (15)$$

$$p_2 = L(x_p + 2, y_p). \quad (16)$$

While three is the minimum number of points necessary to fit the parabola given by (10), we use five points in order to achieve a better fit. We do not use more than five points, since more distant points may be influenced by other peaks or they may fail to model the correct peak due to random noise.

The least-squares fit to a parabola ($y = ax^2 + bx + c$) with $x = \{-2, -1, 0, 1, 2\}$ yields

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \frac{1}{7} & -\frac{1}{14} & -\frac{1}{7} & -\frac{1}{14} & \frac{1}{7} \\ -\frac{1}{5} & -\frac{1}{10} & 0 & \frac{1}{10} & \frac{1}{5} \\ -\frac{3}{35} & \frac{12}{35} & \frac{17}{35} & \frac{12}{35} & -\frac{3}{35} \end{bmatrix} \begin{bmatrix} p_{-2} \\ p_{-1} \\ p_0 \\ p_1 \\ p_2 \end{bmatrix}. \quad (17)$$

We can now solve for x_0 and σ_x using

$$x_0 = x_p - \frac{b}{a} \quad (18)$$

and

$$\sigma_x = \frac{1}{\sqrt{-2a}}. \quad (19)$$

The derivation for y_0 and σ_y is the same, except that we project onto the line $x = x_p$. The values of x_0 and y_0 yield a subpixel localization result, since this is the estimated location of the peak in the likelihood function, and σ_x and σ_y yield direct estimates for the uncertainty in the localization result.

B. Probability of Failure

In addition to estimating the uncertainty in the localization estimate, we can use the likelihood scores to estimate the probability of a failure to detect the correct position of the robot. This is particularly useful when the environment yields few landmarks or other references for localization and thus many positions appear similar to the robot. We estimate this probability of failure by comparing the likelihood scores for the peak selected as the most likely robot position to the scores in the rest of the space. Alternatively, when there are multiple peaks, we can select enough peaks to ensure that one of them is the robot position with some desired probability using this approach.

We estimate the total likelihood of a peak by summing a small number of values around the peak, since they generally become small very quickly. The remainder of the values are also estimated efficiently. Whenever a cell in the search space is considered, we compute not only a bound on the maximum score that can be achieved, but also an estimate on the average score that is achieved by determining the score for the center of the cell. If the cell is pruned, we estimate the score for the entire cell by multiplying the score for the center by the size of the cell. The estimated score for the entire cell is then added to a running total. This yields a very good estimate, since cells with large scores cannot be pruned until they become small. We thus get good estimates when the score is large. Lower quality estimates are obtained when the score is small, but this does not significantly affect the overall sum.

Let S_p be the sum obtained for the largest peak in the pose space and S_t be the overall sum for the pose space (including the largest peak) as described above. We estimate the probability of correctness for the largest peak as

$$P_c = \frac{S_p}{S_t}. \quad (20)$$

VI. EXPERIMENTS

We have tested our approach in a number of experiments using both synthetic data, where precise ground-truth was available for comparison, and real range data from stereo vision, including experimental localization results for the Sojourner rover on Mars.

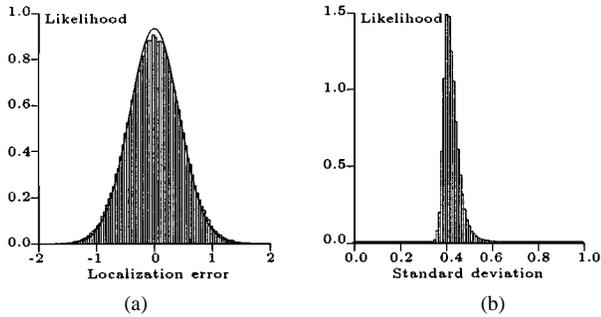


Fig. 3. Distribution of errors and estimated standard deviations in synthetic landmark localization experiment. (a) Comparison of estimated distribution of localization errors (solid line) to observed distribution of localization errors (bar graph). (b) Distribution of estimated standard deviations in the localization estimate.

A. Synthetic Data

We first applied these techniques to localization using landmarks in randomized experiments. In these experiments, we randomly generated a synthetic environment containing 160 landmarks in a 256×256 square. Let us say that each unit is 10 cm (though the entire problem scales to an arbitrary size). In each trial, seven of the ten landmarks closest to some random robot location were considered to be observed by the robot (with Gaussian error in both x and y with standard deviation $\sigma = 1$ unit) along with three spurious landmarks not included in the map. Localization was then performed using these ten observed landmarks with no knowledge of the position of the robot in this environment. Over 100 000 trials, the robot was correctly localized in 99.8% of the cases, with an average error in the correct trials of 0.356 units in each dimension. The average estimated standard deviation in the localization using the techniques from Section V was 0.427 units.

Fig. 3(a) shows the distribution of actual errors observed versus the distribution that we expect from the average standard deviation estimated in the trials. The close similarity of the plots indicates that the estimated standard deviation is a very good estimate of the actual value. It appears that this estimate is slightly smaller than the true value, since the frequency of the observed errors is slightly above the curve at the tails and lower at the peak. However, the overall similarity is very high. The similarity between these plots validates the approximation of the likelihood function as a normal distribution in the neighborhood of the peak. Fig. 3(b) shows the distribution of the estimated standard deviations in this experiment. It can be observed that the estimate is very consistent between trials, since the plot is very strongly peaked near the location of the average estimate. Taken together, these plots indicate that the standard deviation estimates are very likely to be accurate for each individual trial.

We also tested the probability of correctness measure in these trials. The average probability of correctness computed for the trials that resulted in the correct localization was 0.993, while the average probability of correctness for the failures was 0.643. The probability of correctness measure thus yields information that can be used to evaluate whether the localization result is

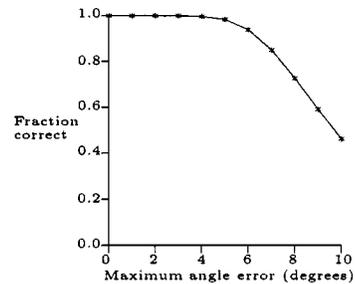


Fig. 4. Fraction of trials in which the correct qualitative robot position was found in experiments using synthetic data with varying amounts of error in robot orientation.

reliable and whether additional sensing is necessary to disambiguate between multiple positions.

In order to study the sensitivity of our approach to rotation errors, we tested a problem where 90% of the data were inliers. In these experiments, a random error bounded by some maximum value was added to the robot orientation in each trial. The search still examined only the space of translations of the robot. Fig. 4 shows the fraction of trials in which the correct position was found as a function of the allowable error. The fraction of successful trials started dropping noticeably when angle errors above 5° were allowed, but the performance was very good when lower levels of error were allowed. This indicates that search over the space of robot orientations is necessary when the orientation is not known to better than 5° .

B. Localization Using Stereo Range Data

In practice, we perform matching between three-dimensional occupancy maps in order to achieve localization for planetary rovers. For these occupancy maps, we consider each cell to be either occupied or unoccupied (with no in-between states). While several methods can be used for generating such a representation, we use stereo vision on the Rocky 7 rover [14] to compute range images using the techniques that have been previously described by Matthies [18], [55].

Once a range image has been computed from the stereo imagery, we convert it into a voxel-based map representation. We first rotate the data such that it has the same relative orientation as the map we are comparing it to. Here we operate under the assumption that the orientation of the robot is known through sensors other than vision (for example, both Sojourner and Rocky 7 have rate gyros and accelerometers and Rocky 7 also uses a sun sensor for orientation determination [56]). The localization techniques can also be generalized to determine the robot's orientation.

The next step is to bin the range points in a three-dimensional occupancy map of the surroundings at some specified scale. We eliminate the need to search over the possible translations of the robot in the z -direction by subtracting a local average of the terrain height from each cell (i.e., a high-pass filter). This step is not strictly necessary, and it reduces our ability to determine height changes in the position of the robot, but it also reduces the computation time that is required to perform localization. A subsequent step can be performed to determine the robot elevation, if desired. Each cell in the occupancy map that contains a

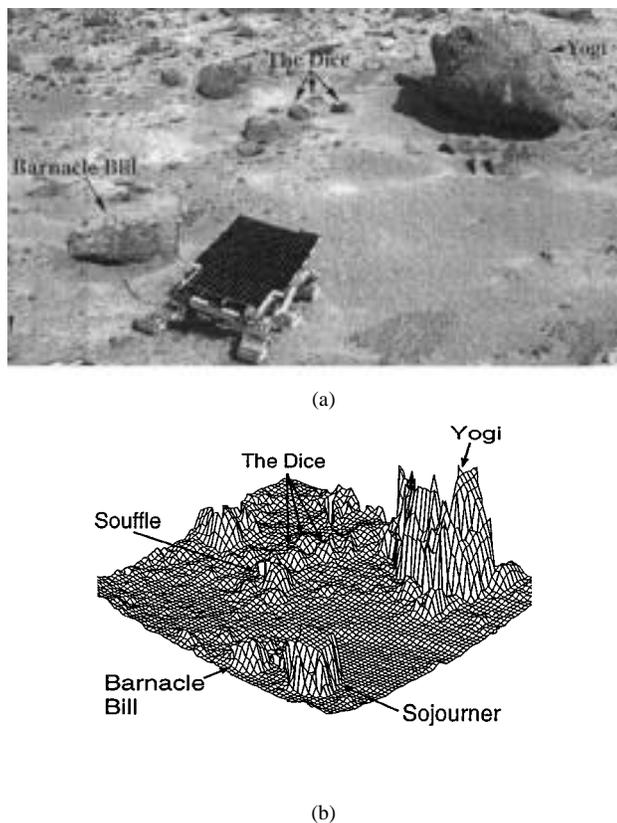


Fig. 5. Terrain map generated from Pathfinder imagery: (a) annotated image mosaic of Sojourner and rocks on Mars and (b) terrain map generated from stereo imagery.

range pixel is said to be *occupied*, and the others are said to be *unoccupied*. Fig. 5 gives an example of a terrain map that was generated using imagery from the Mars Pathfinder mission.

We have tested this method using both terrestrial data and data from the Mars Pathfinder mission. These experiments indicate that self-localization can be performed using our method with approximately the same results as a human operator, without requiring a downlink cycle. In addition, only a few seconds are needed to perform localization. Experiments indicate that localization can be performed on a SPARCstation 20 in under 5 s with maps discretized at 2-cm resolution. Similar experiments performed on-board Rocky 7 (Motorola 68060 CPU) require approximately 20 s to perform localization.

C. Mars Yard

We initially tested our method with images taken in the JPL Mars Yard³ using cameras mounted on a tripod at approximately the Rocky 7 mast height. Fig. 6 shows a set of images that was used in testing the localization techniques. The set consists of 12 stereo pairs acquired at one meter intervals along a straight line with approximately the same heading.

In these tests, we determined the estimated position changes by finding the relative position between each pair of consecutive images. The localization techniques yielded a qualitatively correct position between each pair of consecutive images. The

average absolute error in the position estimates was 0.0342 m in the downrange direction and 0.0367 m in the sideways direction from the position measured by hand. Much of this error can be attributed to human error in determining the ground truth for the data.

Additional tests were performed on imagery where the camera system was panned 25° left and right. In these tests, occupancy maps from the panned images were matched to occupancy maps for the unpanned images. All 24 trials yielded the correct qualitative result. The average absolute error was 0.0138 m in the downrange direction and 0.0225 m in the sideways direction.

In these tests, the average number of positions examined was 18.45% of the total number of positions in the discretized search space. A speedup of greater than 5 was thus achieved through the use of the efficient search techniques.

D. Mars Pathfinder

To validate our approach for use on a Mars rover, we have tested it using data from the Mars Pathfinder mission. A map of the terrain surrounding the Pathfinder lander was first generated using stereo imagery. For each position of Sojourner at which we tested the localization techniques, we generated an occupancy map of the terrain using range data from Sojourner's stereo cameras. This local map was then compared to the global map from the lander.

Unfortunately, this test has only been possible at a few locations due to the limited amount of data returned to Earth, the lack of interesting terrain in some of the imagery we do have, and the lack of a comparison value for most positions (except those where Sojourner was imaged by the lander cameras). In practice, these techniques could be exercised much more frequently since they would not require downlinking image data to Earth and the comparison value is only necessary for testing. We envision a scenario where the data from the rover's body-mounted cameras, which would be operating frequently in order to perform obstacle detection, would be used to perform localization whenever sufficient terrain was evident in the imagery. In addition, the imagery from mast cameras could be used for localization when the positional uncertainty grows beyond the desired level and the imagery from the body-mounted cameras is unsuitable.

As an example of the data, Fig. 7 shows the position of Sojourner as seen from the lander and the view from Sojourner at the end of sol 21⁴ of the Mars Pathfinder mission. Note that the stereo data obtained from Sojourner is not as good as we hope to achieve in future missions. Accurate stereo data is achieved only for the central portion of the Sojourner imagery due to inaccurate calibration of the fish-eye lenses. The field-of-view that we have to work with is thus relatively small. However, we have achieved good localization results with this data.

Table I shows the results of localization using the techniques described in this paper versus the localization that was obtained by a human operator through overlaying a rover model on the stereo data obtained from imaging the rover from the lander.

³See <http://robotics.jpl.nasa.gov/tasks/sci/over/marsyard>

⁴A *sol* is a Martian day.

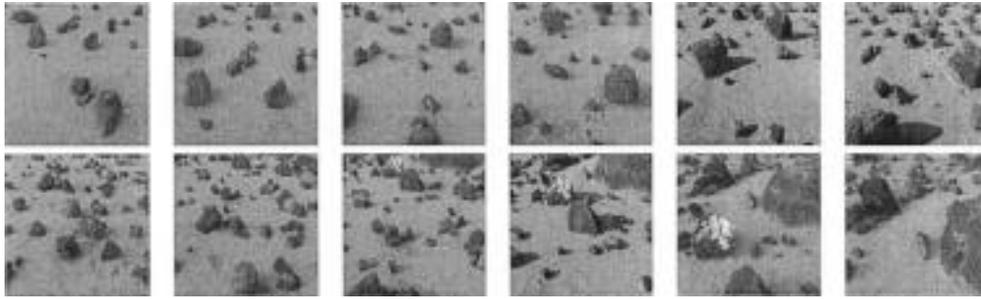


Fig. 6. A sequence of images used for testing the localization techniques.



(a)



(b)

Fig. 7. Sojourner on sol 21 (near “Souffle”): (a) image from the lander and (b) image from Sojourner.

For sol 42, we have two localization results, one prior to and one after a turn by the rover. The operator localization was performed after the turn.

The results show very close agreement between our techniques and the operator localization for four of the cases. For sols 4, 27, and 72, there is some disagreement. Possible sources of error include inaccurate calibration of either the rover or lander cameras and operator error in performing localization. Manual examination of the maps has shown that the localization techniques determine the qualitatively correct position in these cases. While no ground truth exists, the similarity of the positions estimated by our method and by the human operator indicate that our method can perform localization approximately as well as a human operator.

TABLE I
COMPARISON OF ROVER POSITIONS DETERMINED BY A HUMAN OPERATOR OVERLAYING A ROVER MODEL ON STEREO DATA OF THE ROVER AND BY OUR LOCALIZATION TECHNIQUES

Sol	Operator		Localization		Difference	
	x (m)	y (m)	x (m)	y (m)	x (m)	y (m)
4	3.28	-2.69	3.01	-2.64	-0.27	0.05
10	4.34	-3.24	4.24	-3.27	-0.10	-0.03
21	3.32	-2.60	3.37	-2.65	0.05	-0.05
27	-5.42	2.85	-4.98	2.75	0.44	-0.10
42a	-3.00	-1.86	-3.02	-1.87	-0.02	-0.01
42b	-3.00	-1.86	-3.00	-1.87	0.00	-0.01
72	-8.93	-1.57	-8.99	-1.35	-0.06	0.22

VII. DISCUSSION

The primary contribution of our approach is the development of a robust, global localization method that uses all available three-dimensional information. Accurate localization is performed in the presence of outliers and significant noise in the map data through the use of the maximum-likelihood map matching measure. An efficient search of the global pose space is performed through the use of the branch-and-bound pruning techniques. The maximum-likelihood measure yields a likelihood surface over the possible positions of the robot and thus our method can be used to generate probability distributions for use in the Markov localization method (see, for example, [5]–[10]). This method provides a means for combining probability distributions generated at multiple robot positions and incorporating uncertainties from dead-reckoning errors.

Most work on mobile robot localization has dealt with indoor environments, where the environment is usually modeled in two dimensions. In contrast to previous research, we have concentrated on achieving localization in completely unstructured outdoor terrain, such as a rover might encounter on Mars. In such terrain, a three-dimensional representation of the environment is crucial. We note, however, that our approach is general, and can be applied to virtually any environment that can be mapped. The application of this approach to other environments, such as a building interior is straightforward using either discrete features or an occupancy grid representation. It should be noted that in such cases, more locations in the environment will look quite similar and thus selecting a single robot position may not be sufficient for robust navigation. In this case, we should maintain a list of the robot positions that achieve a likelihood above

some threshold. An even better strategy would be to maintain the entire likelihood surface using a multi-resolution data structure [57] or sampling [58]. This likelihood can then be propagated after the robot moves in order to disambiguate the positions.

A possible drawback to this technique is the assumption that the environment sensed by the robot contains sufficient shape to reliably estimate the robot position. However, since our approach can be extended to allow disambiguation using multiple robot positions, this is not a strong limitation. Any localization method requires some distinctive feature in the environment that allows that robot to determine its position in a map.

At present, this method has been implemented only for robot translations. Our experiments indicate that an error in the robot orientation of up to 5° is acceptable for an implementation that does not search over the robot orientation. For cases where the orientation is completely unknown, the extension of these techniques to additional degrees of freedom is not complex. If we know the orientation of the robot to some degree through other sensors, and thus can bound the search space, the additional search time will not grow drastically. However, a fully unconstrained pose space will require significant computational resources using this search strategy.

Gutmann *et al.* [59] have compared Markov localization techniques to scan matching using a Kalman filter. They find that scan matching techniques are more efficient and accurate, but that Markov localization techniques are more robust to noise and sensor error. Our technique shares the robustness of Markov localization techniques (if we do not limit our analysis to the single highest peak in the likelihood function). In addition, since we use all of the available metric information and detect the robot position to subpixel accuracy in a discretized pose space, we are able to achieve high accuracies. Furthermore, the techniques that we describe can be used as part of a Kalman filter position estimator. However, since we perform global localization, we are not able to match the efficiency of scan matching, where an initial estimate is refined in order to determine the position of the robot.

VIII. SUMMARY

We have described a method for performing self-localization for mobile robots using maximum-likelihood matching of maps. The map of visible features at the robot's current position is compared to a global map that has been previously generated (possibly by combining the maps from the robot's previous positions). The best relative position between the maps is detected using a global branch-and-bound search technique that does not require an initial estimate of the robot position. The search is performed relative to a novel maximum-likelihood map similarity measure that selects the robot position at which the maps best agree. This map measure is very general and robust to map errors.

This probabilistic formulation of the map matching problem allows the uncertainty in the localization of individual map features to be treated accurately in the matching process. In addition, performing a polynomial fit to the log-likelihood function allows both subpixel localization to be performed and uncertainty estimates to be computed, which can be propagated in a position tracking mechanism such as the extended Kalman filter.

Our goal in the design of this approach is to provide greater autonomy for Mars rovers. Through the use of our method, we can perform self-localization on Mars within the confines of a science site where panoramic stereo imagery has been taken from the lander or from the rover mast cameras. These techniques can also be used to improve position estimation on long traverses by periodically stopping to perform localization versus the previous position and to image the terrain ahead of the rover. The application of these techniques to data from the Mars Pathfinder mission indicates that we can perform autonomous localization with approximately the same accuracy as a human operator without requiring communication with Earth.

An area that bears further study is the development of a localizability measure for terrain maps in order to plan effective localization steps. In the future, we also plan to integrate our approach into an integrated navigation methodology, in which a Kalman filter is used to synthesize a robot position estimate from a variety of sensors and the robot's path planner interacts with the Kalman filter and the localization techniques to plan when and where localization should be performed.

ACKNOWLEDGMENT

This research is an element of the Long Range Science Rover project, which is developing technology for future Mars missions. The author would like to thank all of the people who have been involved with the Mars Pathfinder mission and the Long Range Science Rover project. In particular, thanks are due to L. Matthies for generating the range images from the Pathfinder IMP cameras that were used in some experiments reported here and to S. Laubach for collecting the data in the Mars Yard used in other experiments. Finally, the author would like to thank the people who have read and commented on various versions of this research, including B. Wilcox, S. Hayati, M. Maimone, A. Johnson, and the anonymous reviewers.

REFERENCES

- [1] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques," *J. Robot. Syst.*, vol. 14, pp. 231–249, Aug. 1997.
- [2] R. Talluri and J. K. Aggarwal, "Position estimation techniques for an autonomous mobile robot — A review," in *Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds. Singapore: World Scientific, 1993, ch. 4.4, pp. 769–801.
- [3] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 850–863, Sept. 1993.
- [4] C. F. Olson, "A probabilistic formulation for Hausdorff matching," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 1998, pp. 150–156.
- [5] I. Nourbakhsh, R. Powers, and S. Birchfield, "DERVISH: An office-navigating robot," *AI Mag.*, vol. 16, no. 2, pp. 53–60, 1995.
- [6] R. Simmons and S. Koenig, "Probabilistic navigation in partially observable environments," in *Proc. Int. Joint Conf. Artificial Intell.*, vol. 2, 1995, pp. 1660–1667.
- [7] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien, "Acting under uncertainty: Discrete Bayesian models for mobile robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 2, 1996, pp. 963–972.
- [8] S. Koenig and R. G. Simmons, "Unsupervised learning of probabilistic models for robot navigation," in *Proc. IEEE Conf. Robot. Automat.*, vol. 3, 1996, pp. 2301–2308.
- [9] D. Fox, W. Burgard, and S. Thrun, "Active Markov localization for mobile robots," *Robot. Auton. Syst.*, vol. 25, no. 3–4, pp. 195–207, Nov. 1998.

- [10] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Mach. Learning*, vol. 31, no. 1–3, pp. 29–53, 1998.
- [11] D. P. Huttenlocher and W. J. Rucklidge, "A multi-resolution technique for comparing images using the Hausdorff distance," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 1993, pp. 705–706.
- [12] C. F. Olson and D. P. Huttenlocher, "Automatic target recognition by matching oriented edge pixels," *IEEE Trans. Image Processing*, vol. 6, pp. 103–113, Jan. 1997.
- [13] W. J. Rucklidge, "Efficiently locating objects using the Hausdorff distance," *Int. J. Comput. Vision*, vol. 24, no. 3, pp. 251–270, Sept. 1997.
- [14] S. Hayati et al., "The Rocky 7 rover: A Mars sciencecraft prototype," in *Proc. IEEE Conf. Robot. Automat.*, vol. 3, 1997, pp. 2458–2464.
- [15] D. Shirley and J. Matijevic, "Mars Pathfinder microrover," *Auton. Robot.*, vol. 2, pp. 283–289, 1995.
- [16] L. H. Matthies, C. F. Olson, G. Tharp, and S. Laubach, "Visual localization methods for Mars rovers using lander, rover, and descent imagery," in *Proc. 4th Int. Symp. Artificial Intell. Robot. Automat. Space*, 1997, pp. 413–418.
- [17] S. L. Laubach, C. F. Olson, J. W. Burdick, and S. Hayati, "Long range navigation for Mars rovers using sensor-based path planning and visual localization," in *Proc. 5th Int. Symp. Artificial Intell. Robot. Automat. Space*, 1999, pp. 455–461.
- [18] L. Matthies, "Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation," *Int. J. Comput. Vision*, vol. 8, no. 1, pp. 71–91, July 1992.
- [19] C. F. Olson, "Mobile robot self-localization by iconic matching of range maps," in *Proc. Int. Conf. Adv. Robot.*, 1997, pp. 447–452.
- [20] C. F. Olson and L. H. Matthies, "Maximum-likelihood rover localization by matching range maps," in *Proc. Int. Conf. Robot. Automat.*, vol. 1, 1998, pp. 272–277.
- [21] C. F. Olson, "Subpixel localization and uncertainty estimation using occupancy grids," in *Proc. Int. Conf. Robot. Automat.*, vol. 3, 1999, pp. 1987–1992.
- [22] N. Ayache and O. D. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 804–819, Dec. 1989.
- [23] A. Curran and K. J. Kyriakopoulos, "Sensor-based self-localization for wheeled mobile robots," *J. Robot. Syst.*, vol. 12, no. 3, pp. 163–176, 1995.
- [24] J. Horn and G. Schmidt, "Continuous localization of a mobile robot based on 3d-laser-range-data, predicted sensor images, and dead-reckoning," *Robot. Auton. Syst.*, vol. 14, pp. 99–118, 1995.
- [25] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 376–382, June 1991.
- [26] Z. Zhang and O. Faugeras, "A 3D world model builder with a mobile robot," *Int. J. Robot. Res.*, vol. 11, no. 4, pp. 269–285, August 1992.
- [27] K. Sugihara, "Some location problems for robot navigation using a single camera," *Comput. Vision, Graphics, Image Processing*, vol. 42, pp. 112–129, 1988.
- [28] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 251–263, Apr. 1997.
- [29] P. C. Gaston and T. Lozano-Pérez, "Tactile recognition and localization using object models: The case of polyhedra on a plane," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 257–265, May 1984.
- [30] W. E. L. Grimson and T. Lozano-Pérez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robot. Res.*, vol. 3, no. 3, pp. 3–35, 1984.
- [31] M. Drumheller, "Mobile robot localization using sonar," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 325–332, Mar. 1987.
- [32] K. T. Simsarian, T. J. Olson, and N. Nandhakumar, "View-invariant regions and mobile robot self-localization," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 810–816, Oct. 1996.
- [33] R. Talluri and J. K. Aggarwal, "Mobile robot self-location using model-image feature correspondence," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 63–77, Feb. 1996.
- [34] I. J. Cox, "Blanche—An experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 193–204, Apr. 1991.
- [35] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2D range scans," *J. Intell. Robot. Syst.*, vol. 18, pp. 249–275, 1997.
- [36] J. L. Crowley, F. Wallner, and B. Schiele, "Position estimation using principal components of range data," *Robot. Auton. Syst.*, vol. 23, pp. 267–276, 1998.
- [37] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE J. Robot. Automat.*, vol. RA-3, pp. 249–265, June 1987.
- [38] S. Atiya and G. D. Hager, "Real-time vision-based robot localization," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 785–800, Dec. 1993.
- [39] R. Talluri and J. K. Aggarwal, "Position estimation for an autonomous mobile robot in an outdoor environment," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 573–584, Oct. 1992.
- [40] F. Stein and G. Medioni, "Map-based localization using the panoramic horizon," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 892–896, Dec. 1995.
- [41] W. B. Thompson, T. C. Henderson, T. L. Colvin, L. B. Dick, and C. M. Valiquette, "Vision-based localization," in *Proc. DARPA Image Understanding Workshop*, 1993, pp. 491–498.
- [42] F. Cozman and E. Krotkov, "Automatic mountain detection and pose estimation for teleoperation of lunar rovers," in *Proc. IEEE Conf. Robot. Automat.*, vol. 3, 1997, pp. 2452–2457.
- [43] I. S. Kweon and T. Kanade, "High-resolution terrain map from multiple sensor data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 278–292, Feb. 1992.
- [44] R. Szeliski, "Estimating motion from sparse range data without correspondence," in *Proc. Int. Conf. Comput. Vision*, 1988, pp. 207–216.
- [45] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vision*, vol. 13, no. 2, pp. 119–152, 1994.
- [46] W. E. L. Grimson and D. P. Huttenlocher, "On the sensitivity of the Hough transform for object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 255–274, Mar. 1990.
- [47] W. E. L. Grimson, D. P. Huttenlocher, and D. W. Jacobs, "A study of affine matching with bounded sensor error," *Int. J. Comput. Vision*, vol. 13, no. 1, pp. 7–32, 1994.
- [48] W. E. L. Grimson and D. P. Huttenlocher, "Analyzing the probability of a false alarm for the Hausdorff distance under translation," in *Proc. Workshop Performance versus Methodology in Comput. Vision*, 1994, pp. 199–205.
- [49] A. Rosenfeld and J. Pfaltz, "Sequential operations in digital picture processing," *J. ACM*, vol. 13, pp. 471–494, 1966.
- [50] J. Piper and E. Granum, "Computing distance transformations in convex and nonconvex domains," *Pattern Recognit.*, vol. 20, no. 6, pp. 599–615, 1987.
- [51] G. Borgefors, "Distance transformations in digital images," *Comput. Vision, Graphics, Image Processing*, vol. 34, pp. 344–371, 1986.
- [52] S. Pavel and S. G. Akl, "Efficient algorithms for the Euclidean distance transform," *Parallel Processing Lett.*, vol. 5, no. 2, pp. 205–212, 1995.
- [53] R. J. Lipton and R. E. Tarjan, "Applications of a planar separator theorem," *SIAM J. Computing*, vol. 9, no. 3, pp. 615–627, 1980.
- [54] J. L. Bentley, B. W. Weide, and A. C. Yao, "Optimal expected time algorithms for closest point problems," *ACM Trans. Math. Software*, vol. 6, pp. 563–580, 1980.
- [55] L. Matthies, A. Kelly, T. Litwin, and G. Tharp, "Obstacle detection for unmanned ground vehicles: A progress report," in *Proc. Int. Symp. Robot. Res.*, 1996, pp. 475–486.
- [56] R. Volpe, "Navigation results from desert field tests of the Rocky 7 Mars rover prototype," *Int. J. Robot. Res.*, vol. 18, no. 7, pp. 669–683, July 1999.
- [57] W. Burgard, A. Derr, D. Fox, and A. B. Cremers, "Integrating global position estimation and position tracking for mobile robots: The dynamic Markov localization approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 1998, pp. 730–735.
- [58] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proc. IEEE Conf. Robot. Automat.*, 1999, pp. 1322–1328.
- [59] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige, "An experimental comparison of localization methods," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 1998, pp. 736–743.



Clark F. Olson (S'92–M'94) received the B.S. degree in computer engineering and the M.S. degree in electrical engineering from the University of Washington, Seattle, in 1989 and 1990, respectively. He received the Ph.D. degree in computer science from the University of California, Berkeley, in 1994.

After spending two years as a Postdoctoral Researcher at Cornell University, he was employed by the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, where he is currently a Member of the Technical Staff in the Machine Vision

Group. His research interests include computer vision and mobile robotics.