Landmark Selection for Vision-Based Navigation

Pablo Sala, *Student Member, IEEE*, Robert Sim, *Member, IEEE*, Ali Shokoufandeh, *Member, IEEE*, and Sven Dickinson, *Member, IEEE*

***Abstract*—Recent work in the object recognition community has yielded a class of interest-point-based features that are stable under significant changes in scale, viewpoint, and illumination, making them ideally suited to landmark-based navigation. Although many such features may be visible in a given view of the robot's environment, only a few such features are necessary to estimate the robot's position and orientation. In this paper, we address the problem of automatically selecting, from the entire set of features visible in the robot's environment, the minimum (optimal) set by which the robot can navigate its environment. Specifically, we decompose the world into a small number of maximally sized regions, such that at each position in a given region, the same small set of features is visible. We introduce a novel graph theoretic formulation of the problem, and prove that it is NP-complete. Next, we introduce a number of approximation algorithms and evaluate them on both synthetic and real data. Finally, we use the decompositions from the real image data to measure the localization performance versus the undecomposed map.**

***Index Terms*—Feature selection, localization, machine vision, mapping, mobile robots.**

I. Introduction

In THE domain of exemplar-based (as opposed to generic) object recognition, the computer vision community has recently adopted a class of interest-point-based features, e.g., [1]–[4]. Such features typically encode a description of image appearance in the neighborhood of an interest point, such as a detected corner or scale-space maximum. The appeal of these features over their appearance-based predecessors is their invariance to changes in illumination, scale, image translation, and rotation, and minor changes in viewpoint (rotation in depth). These properties make them ideally suited to the problem of landmark-based navigation. If we can define a set of invariant features that uniquely defines a particular location in the environment, these features can, in turn, define a visual landmark.

To use these features, we could, for example, adopt a localization approach proposed by Basri and Rivlin [5], and Wilkes *et al.* [6], based on the linear combination of views (LC) technique. During a training phase, the robot is manually "shown" two views of each landmark in the environment, by which the robot is to later navigate. These views, along with the positions at which they were acquired, form a database of landmark views. At run-time, the robot takes an image of the environment and attempts to match the visible features to the various landmark views it has stored in its database. Given a match to some landmark view, the robot can compute its position and orientation in the world.

There are two major challenges with this approach. First, from any given viewpoint, there may be hundreds or even thousands of such features. The union of all pairs of landmark views may therefore yield an intractable number of distinguishable features that must be indexed in order to determine which landmark the robot may be viewing.[1] Fortunately, only a small number of features are required (in each model view) to compute the robot's pose. Therefore, of the hundreds of features visible in a model view, which small subset should we keep?

The second challenge is to automate this process, and let the robot automatically decide on an optimal set of visual landmarks for navigation. What constitutes a good landmark? A landmark should be both distinguishable from other landmarks (a single floor tile, for example, would constitute a bad landmark since it is repeated elsewhere on the floor) and widely visible (a landmark visible only from a single location will rarely be encountered and, if so, will not be persistent). Therefore, our goal can be formulated as partitioning the world into a minimum number of maximally sized contiguous regions, such that the same set of features is visible at all points within a given region.

There is an important connection between these two challenges. Specifically, given a region, inside of which all points see the same set of features (our second challenge), what happens when we reduce the set of features that must be visible at each point (first challenge)? Since this represents a weaker constraint on the region, the size of the region can only increase, yielding a smaller number of larger regions covering the environment. As mentioned earlier, there is a lower bound on the number of features that can define a region, based on the pose-estimation algorithm and the degree to which we want to overconstrain its solution.

Combining these two challenges, we arrive at the main problem addressed by this paper: from a set of views acquired

Manuscript received November 8, 2004; revised April 18, 2005. This paper was recommended for publication by Associate Editor D. Sun and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by NSERC, in part by PREA, in part by CITO, in part by MD Robotics, and in part by ONR. This paper was presented in part at the IEEE International Conference on Intelligent Robots and Systems, Sendai, Japan, September 2004.

P. Sala and S. Dickinson are with the Department of Computer Science, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: psala@cs.toronto.edu; sven@cs.toronto.edu).

R. Sim is with the Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: simra@cs.ubc.ca).

A. Shokoufandeh is with the Department of Computer Science, College of Engineering, Drexel University, Philadelphia, PA 19104 USA (e-mail: ashokouf@cs.drexel.edu).

Digital Object Identifier 10.1109/TRO.2005.861480

[1]Worst-case indexing complexity would occur during the kidnapped localization task, in which the robot has no prior knowledge of where it is in the world. Under normal circumstances, given the currently viewed landmark and the current heading, the space of landmark views that must be searched can be constrained. Still, even for a small set of landmark views, this may yield a large search space of features.

at a set of sampled positions in a given environment, partition the world into a minimum set of maximally sized regions, such that at all positions within a given region, the same set of k features is visible, where k is defined by the pose-estimation procedure (or some overconstrained version of it). We begin by introducing a novel, graph theoretic formulation of the problem, and proceed to prove its intractability. In the absence of optimal, polynomial-time algorithms, we introduce six different heuristic algorithms for solving the problem. We have constructed a simulator that can generate thousands of worlds with varying conditions, allowing us to perform exhaustive empirical evaluation of the six algorithms. Following a comparison of the algorithms on synthetic environments, we adopt the most effective algorithm, and test it on imagery of a real environment. We conclude with a discussion of the main contributions made and possible directions for future work.

II. Related Work

In previous work on robot navigation using point-based features, little or no attention has been given to the size of the landmark database or the number of landmark lookups required for localization. This is especially problematic for map representations that rely on large numbers of generic features, such as corners or line segments [7]–[9]. Recently, a number of image-based feature detectors, such as the scale-invariant feature transform (SIFT), and other scale-, rotation-, and affine-invariant features have been developed that provide stronger discriminative power [1], [3], [4]. Despite these enhancements, maps constructed using visual features often entail mapping very large numbers of points in space.

There are several existing feature-based approaches to environment representation. Se *et al.* [10] use SIFT features as landmarks. The robot automatically updates a 3-D landmark map with the reliable landmarks seen from the current position using Kalman filtering techniques. The position of the robot is estimated using the odometry of the robot as an initial guess, and is improved using the map. Trinocular vision is used to estimate the 3-D locations of landmarks and their regions of confidence, with all reliable landmarks stored in a dense database.

Navigation by landmark recognition is also possible without knowledge of the locations of the landmarks in a map of the environment. Localization can be accomplished in a view-based fashion, in which the robot knows only the image location of the landmarks in a collection of model views of the environment acquired at known positions and orientations. One such approach is the LC technique, which was first introduced by Ullman and Basri for object recognition, and later applied to vision-based navigation by Basri and Rivlin [5]. The authors proved that if a scene is represented as a set of 2-D views, each novel view of the scene can be computed as a linear combination of the model views. From the values of the linear coefficients, it is possible to estimate the position from which the novel view was acquired, relative to that of the model views. Wilkes *et al.* [6] described a practical robot navigation system that used the LC technique. Their method consists of decomposing the environment into regions, within which a set of model views of a particular area of the environment may be used to determine the position of the robot. In these original applications of the LC method, the features comprising the model views were typically linear features extracted from the image.

The view-based approach of Sim and Dudek [11] consists of an offline collection of monocular images sampled over a space of poses. The landmarks consist of encodings of the neighborhoods of salient points in the images, obtained using an attention operator. Landmarks are tracked between contiguous poses, and added to a database if they are observed to be stable through a region of reasonable size, and sufficiently useful for pose estimation according to an *a priori* utility measure. Each stored landmark is encoded by learning a parameterization of a set of computed landmark attributes. The localization is performed by finding matches between the candidate landmarks visible in the current image and those in the database. A position estimate is obtained by merging the individual estimates yielded by each computed attribute of each matched candidate landmark.

In an attempt to address the localization of a robot in a previously mapped environment, Fairfield and Maxwell [12] used visual and spatial information associated with simple, but artificial, landmarks. Their method projects the acquired coordinates of the visual landmark in the image plane into an estimation of the distance between landmark and robot. Ideally, this estimate can be cross-validated with prestored landmark coordinates to localize the robot. Their solution for dealing with robot odometry and landmark distance-estimation errors was to use a simple Kalman filter model in order to correct for accumulated odometry and sensor errors.

DeSouza and Kak [13] presented a comprehensive survey of computer vision methods for both indoor and outdoor navigation. For indoor navigation, they considered three popular models of map-based, map-building-based, and mapless navigation. In each case, they discussed the contributions of existing vision methods to visual information acquisition, landmark detection, cross-validation of visual hypotheses and prestored models, and position estimation for localization. In the context of outdoor robotics, they surveyed the navigation of both structured and unstructured environments. In each case, the relevant contribution of vision to a variety of critical components of navigation systems was considered, including obstacle detection and avoidance, robust road detection, construction of hypothetical scene models, far-point (landmark) triangulation, and global position estimation. In other work, landmarks are used to define topological locations in the world. For example, Mata *et al.* demonstrated a system for topological localization using distinctive image regions [14].

While the focus of our work is on visual navigation, our approach to feature selection is also applicable to other feature-based representations, such as points extracted from range data. There is a rich body of work on mapping and navigation using range-based features. Leonard and Durrant-Whyte developed a map representation using "geometric beacons," corresponding to corners extracted from a sonar signature [15]. Other work has examined similar features in outdoor settings, and underwater [16], [17].

Since there is always a certain amount of uncertainty in estimating the robot's position, some authors have considered

the problem of landmark selection for the purpose of minimizing uncertainty in the computed pose estimate. Sutherland and Thompson [18] demonstrate that the precision of a pose estimate derived from point features in 2-D is dependent on the configuration of the observed features. They also provided an algorithm for selecting an appropriate set of observed features for pose estimation. Olson [19] presented a method for estimating the localization uncertainty of individual landmarks for the purpose of gaze planning. Burschka *et al.* [20] considered the effect of spatial landmark configuration on a robot's ability to navigate. Similarly, Yamashita *et al.* [21] demonstrated motion-planning strategies that take into account landmark configuration for accurate localization.

Methods have also been developed to combine multiple unreliable observations into a more reliable estimate. Measurements from various sensors, data acquired over time, and previous estimates are integrated in order to compute a more accurate estimate of the current robot's pose. In every sensor update, previous data is weighted according to how accurately it predicts the current observations. This technique, called *sensor fusion*, has generally been implemented through the use of Kalman filters and Extended Kalman filters (EKFs). It has been applied to the problem of localization by Leonard and Durrant-Whyte [22] from sonar data obtained over time. A disadvantage of Kalman filters and EKFs is that since they realize a local linear approximation to the exact relationship between the position and observations, they depend on a good *a priori* estimate, and therefore can suffer from robustness problems.

Fox introduced *Markov localization* in [23], a Bayesian approach to localization using Markov chain methods, and maintaining a probability distribution over pose space. As evidence is collected from the sensors, it is used to update the current state of belief of the robot's pose. This approach generalizes beyond the Kalman filter in that multimodal probability distributions can be represented. In [24], Thrun presents an approach based on Markov localization in which neural networks are trained to discover landmarks that will minimize the localization error. The proposed algorithm has the advantage of being widely applicable, since the robot customizes its localization algorithm to the sensors' characteristics and the particular environment in which it is navigating. The localization error achieved by the automatically selected landmarks is shown to outperform the error achieved with landmarks carefully selected by human experts. On the other hand, this approach has the drawback that the training of the neural networks can take several hours, though this process generally needs to be performed only once in an offline stage.

Another set of probabilistic mapping approaches is that of simultaneous localization and mapping (SLAM), in which, after each new measurement, both the robot's pose and the positions of landmarks in the world are reestimated. Davison's work in this direction basically computes a solution to the structure-from-motion problem online [25] using the Shi and Tomasi feature detector to construct maps in real time from a monocular sequence of images. Conventional SLAM approaches based on the Kalman filter suffer, in that the time complexity of each sensor-update step increases with the square of the number of landmarks in the database. To deal with this scalability problem, some authors suggested dividing the global map into submaps, within which the complexity can be bounded [26], [27]. Other researchers [28]–[31] have proposed hierarchical approaches to SLAM, in which a topological map is maintained, organizing the landmarks into smaller regions where feature-based mapping strategies can be applied. Most recently, several authors have tackled the scaling problem with new filter designs, such as sparse-information filters, particle filters, and thin junction-tree filters [32]–[34].

Our work is also closely related to the general problem of computing a minimum description length (MDL) encoding of a set of observations [35]. However, our problem is further defined by the domain-dependent constraint that the encoding must facilitate localization everywhere in the world. Some authors have also examined the problem of feature selection (which image-derived features are optimal for representing camera location) [36], [37]. These approaches also generally seek to compute optimal encodings of the observations, but tend to depend on global image properties, making them susceptible to failure in the presence of minor changes in the scene. For this paper, we assume a feature-extraction approach that recovers multiple local features from single images. In principle, one could apply our work to several such feature detectors to determine which operators produce the most compact descriptions of the world.

While all of the approaches discussed above demonstrate how robot localization can be performed from a set of landmark observations, none consider the issue of eliminating redundancy from the landmark-based map, which at times can grow to contain hundreds of thousands of landmark models. In this paper, we study the problem of automatically selecting a minimum-size subset of landmarks, such that reliable navigation is still possible. While maximizing precision is clearly an important issue, in this paper, we are concerned primarily with selecting landmarks that are widely visible. However, the algorithms presented in this paper can be easily extended to select sets of features that fulfill any given additional constraints.

III. Landmark-Selection Problem Definition

In an offline training phase, images are first collected at known discrete points in pose space, e.g., the accessible vertices (points) of a virtual grid overlaid on the floor of the environment. During collection, the known pose of the robot is recorded for each image, and a set of interest-point-based features are extracted and stored in a database. For each of the grid points, we know exactly which features in the database are visible. Conversely, for each feature in the database, we know from which grid points it is visible.

Consider the example shown in Fig. 1. Fig. 1(a) shows a 2-D world with a polygonal perimeter, a polygonal obstacle in its center, and nine features along the world and obstacle perimeters. In Fig. 1(b)–(g), the area of visibility of some of the features is shown as a colored region. The feature-visibility areas, computed from a set of images acquired at a set of grid points in the world, constitute the input to our problem.

In a view-based localization approach, the current pose of the robot is estimated using, as input, the locations of a small

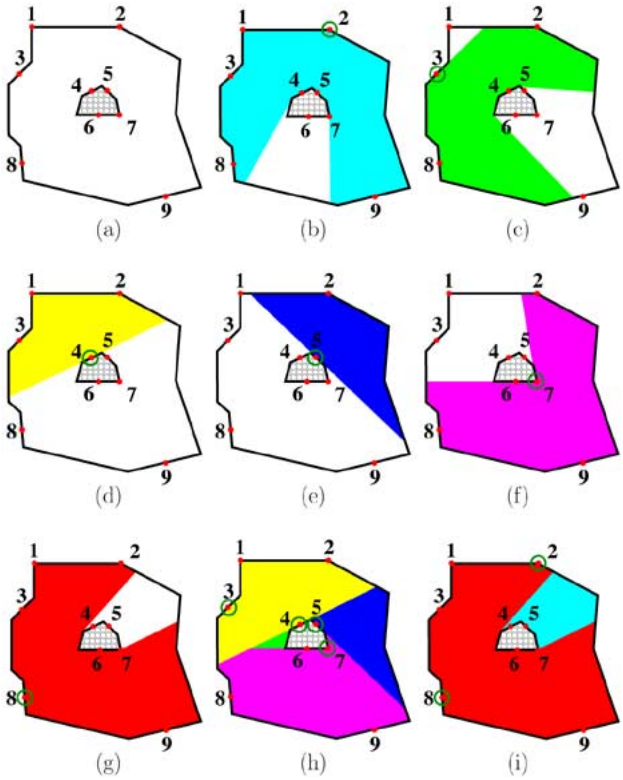

Fig. 1. (a) Simple polygonal world with a polygonal obstacle in its center and nine features. (b)-(g) Visibility areas of some (circled) features. (h) Covering of the world using four features (3, 4, 5, and 7). (i) Covering of the world using two features (2 and 8).

number of features in the current image, matched against their locations in the training images. This set of simultaneously visible features constitutes a landmark. The minimum number of features necessary for this task depends on the method employed for pose estimation. For example, three features are enough for localization in Basri and Rivlin's LC technique [5], which uses a weak perspective projection imaging model. The essential matrix method [38], that properly models perspective projection in the imaging process, requires at least eight features to estimate pose.

To reduce the effect of noise, a larger number of features can be used to overconstrain the solution. This presents a tradeoff between the accuracy of the estimation and the size (in features) of the landmark. Requiring a larger number of features for localization will yield better pose estimation. However, the more constrained a landmark is, the smaller its region of visibility becomes. We will use the parameter k as the number of features that will be employed to achieve pose estimation with the desired accuracy, i.e., the number of features constituting a landmark.

Robot localization from a given position is possible if, from the features extracted from an image taken at that position, there exists a subset of k features that exist in the database and that are simultaneously visible from at least two known locations. For a large environment, the database may be large, and such a search may be costly. For each image feature, we would have to search the entire database for a matching feature until not only k such matches were found, but that those k features were simultaneously visible from at least two separate positions (grid points).

Recalling that k is typically far less than the number of features in a given image, one approach to reducing search complexity would be to prune features from the database subject to the existence of a minimum of k features visible at each point, with those same k features being visible at one or more other positions. Unfortunately, this is a complex optimization problem whose solution still maintains all the features in a single database, leading to a potentially costly search. A more promising approach is to partition the pose space into a number of *regions*, i.e., sets of contiguous grid points, such that for each region, there are at least k features simultaneously visible from all the points in the region. Such a partitioning of the world, in turn, partitions the database of features into a set of smaller databases, each corresponding to what the robot sees in a spatially coherent region. In this latter approach, since k is small, the total number of features (corresponding to the union of all the databases) that need to be retained for localization is much smaller than that of the single database in the previous approach. Therefore, even without prior knowledge of the region in which the robot is located, the search is far less costly.

Let us return to the world depicted in Fig. 1. In this example, we will assume, for sake of clarity, that a single ($k = 1$) feature is sufficient for reliable navigation. However, the reader must note that in practice, a k greater than one is generally required for localization, its particular minimum value depending on the method employed. Under this assumption, one possible decomposition of the world into a set of regions (such that each pose of the world sees at least one feature) is achieved using features 3, 4, 5, and 7, as shown in Fig. 1(h). (In the figure, the feature-visibility areas are shown superimposed for features 3, 7, 5 and 4, in that particular order.) It is clear that all four features in this set are needed to cover the world, since removing any one of them will yield some portion of the world from which none of the remaining three features are visible, meaning that the robot is blind in this area. However, this decomposition is not optimal, since other decompositions with fewer regions are possible. Our goal is to find a minimum decomposition of the world which, in this case, has only two regions. One such decomposition corresponds to the areas of visibility of features 2 and 8, as shown in Fig. 1(i). This minimum set of maximally sized regions is our desired output, and allows us to discard from the database all but features 2 and 8. Since at least one of these two features is seen from every point in the pose space, reliable navigation through the entire world is possible.

Besides reducing the total number of features to be stored, a partitioning of the world into regions offers additional advantages. While navigating inside a region, the corresponding k features are easily tracked between the images that the robot sees. If the expected k features are not all visible in the current image, this may indicate that the robot has left the region in which it was navigating, and is entering a new region. In that case, the visible features can vote for the regions they belong to, if any, according to a membership relationship computed offline. The new region(s) into which the robot is likely moving will be those with at least k votes. Input features would, therefore, be matched to the k model features defining each of the candidate regions. This approach also provides a solution to the *kidnapped robot*

problem, i.e., if the robot is blindfolded and released at an arbitrary position, it can estimate its current pose.

A. A Graph Theoretic Formulation

Before we formally define the minimization problem under consideration, we will introduce some terms.

Definition 3.1: The set of positions at which the robot can be at any time is called the *pose space*. The discrete subset of the pose space from which images were acquired is modeled by an undirected planar graph $G = (V, E)$, where each node $v \in V$ corresponds to a sampled pose, and two nodes are adjacent if the corresponding poses are contiguous in 2-D space.

Definition 3.2: Let F be the set of computed features from all collected images. The *visibility set of* v is the set $\mathcal{F}_v \subset F$ of all features that are visible from pose $v \in V$.

Definition 3.3: A *world instance* consists of a tuple $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v \in V} \rangle$, where the graph G models a discrete set of sampled poses, F is a set of features, and $\{\mathcal{F}_v\}_{v \in V}$ is a collection of visibility sets.

Definition 3.4: A set of poses $R \subset V$ is said to be a *region* if and only if (iff) for all poses $u, v \in R$, there is a path between u and v completely contained in R, i.e., $\forall u, v \in R : \exists \{u = v_0, \ldots, v_h = v\} \subseteq R$, such that $(v_i, v_{i+1}) \in E$ for all $0 \leq i < h$.

Definition 3.5: A collection of regions $D = \{R_1, \ldots, R_d\} \subset 2^V$ is said to be a *decomposition of* V iff $\bigcup_{1 \leq i \leq d} R_i = V$.

Definitions 3.1 to *3.5* define the set of inputs and outputs of interest to our problem. In view of our optimization problem, for a given world instance $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v \in V} \rangle$, one would like to create a minimum cardinality decomposition D. In addition, it will be desirable for a given solution to the optimization problem to satisfy a variety of properties. One property of interest is that of ensuring a minimum amount of overlap between regions in the decomposition. The purpose of overlap is to ensure smooth transitions between regions, as different sets of features become visible to the robot. When one region's features start to fade at its border, the robot can be assured to be within the boundary of some other region, where the new region's landmark is clearly visible. The following definitions formalize this property.

Definition 3.6: The ρ neighborhood of a pose $v \in V$ is the set $N_\rho(v) = \{u \in V : \delta(u, v) \leq \rho\}$, where $\delta(u, v)$ is the length of the shortest path between nodes u and v in G.

Definition 3.7: A decomposition $D = \{R_1, \ldots, R_d\}$ of V is said to be *ρ-overlapping* iff $(\forall v \in V)(\exists i) : N_\rho(v) \subset R_i$.

With these definitions in hand, the problem can now be formally stated as follows.

Definition 3.8: Let k be the number of features required for reliable localization at each position, according to the localization method employed. The *ρ-minimum overlapping region decomposition problem* (ρ-MORDP) for a world instance $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v \in V} \rangle$ consists of finding a minimum-size ρ-overlapping decomposition $D = \{R_1, \ldots, R_d\}$ of V, such that $\forall i : |\bigcap_{v \in R_i} \mathcal{F}_v| \geq k$.

Note that given a solution of size d to this problem, the total number of features needed for reliable navigation is bounded by $d \cdot k$.

IV. Complexity Analysis

Before we consider the complexity of ρ-MORDP, we will present two theorems indicating that ρ-MORDP can be reduced to 0-MORDP ($\rho = 0$), and that a solution to the reduced 0-MORDP can be transformed back into a solution of the more general ρ-MORDP. The first of the following two theorems states that if there is a ρ-overlapping decomposition, such that k features are visible in each region for a certain world instance, then there is a 0-overlapping decomposition for the related problem, also with k features visible in each region. This theorem guarantees that if a solution exists for the ρ-MORDP, then there is also a solution to the related 0-MORDP.

The second theorem states that whenever the related 0-MORDP has a solution $\tilde{D}$, then the ρ-MORDP has a solution, too, and it presents the method to construct it from $\tilde{D}$. We will start by proving three auxiliary lemmas that will be used in the proofs of *Theorems 4.1* and *4.2*.

Lemma 4.1: $\{R_1, \ldots, R_d\}$ is a ρ-overlapping decomposition of V iff $\{\tilde{R}_1, \ldots, \tilde{R}_d\}$ is a 0-overlapping decomposition of V, where $\tilde{R}_i = \{v \in R_i : N_\rho(v) \subseteq R_i\}$ for all $i = 1, \ldots, d$.

Proof: This follows from the following chain of implications:

$$
\begin{aligned}
&\{R_1, \ldots, R_d\} \text{ is a } \rho\text{-overlapping decomposition of } V \\
&\iff (\forall v \in V)(\exists i : 1 \leq i \leq d) N_\rho(v) \subseteq R_i \\
&\overset{(*)}{\iff} (\forall v \in V)(\exists i : 1 \leq i \leq d) v \in \tilde{R}_i \\
&\iff (\forall v \in V)(\exists i : 1 \leq i \leq d) N_0(v) \subseteq \tilde{R}_i \\
&\iff \{\tilde{R}_1, \ldots, \tilde{R}_d\} \text{ is a 0-overlapping decomposition of } V
\end{aligned}
$$

where implication $(*)$ follows from the definition of $\tilde{R}_i$. □

Lemma 4.2: $\{\tilde{R}_1, \ldots, \tilde{R}_d\}$ is a 0-overlapping decomposition of V iff $\{R'_1, \ldots, R'_d\}$ is a ρ-overlapping decomposition of V, where $R'_i = \bigcup_{v \in \tilde{R}_i} N_\rho(v)$ for all $i = 1, \ldots, d$.

Proof: First, observe that R'_i is a region, since $\tilde{R}_i$ is a region and $N_\rho(v)$ is path-connected, as can be inferred from its definition. Now

$$
\begin{aligned}
&\{\tilde{R}_1, \ldots, \tilde{R}_d\} \text{ is a 0-overlapping decomposition of } V \\
&\iff (\forall v \in V)(\exists i) : N_0(v) \subset \tilde{R}_i \\
&\iff (\forall v \in V)(\exists i) : v \in \tilde{R}_i \\
&\overset{(*)}{\iff} (\forall v \in V)(\exists i) : N_\rho(v) \subseteq R'_i \\
&\overset{(**)}{\iff} \{R'_1, \ldots, R'_d\} \text{ is a } \rho\text{-overlapping decomposition of } V.
\end{aligned}
$$

In $(*)$, we use that $(\forall v \in \tilde{R}_i) : N_\rho(v) \subseteq R'_i$, which is a direct implication of the definition of R'_i. In $(**)$, we use that R'_i is a region. □

Lemma 4.3: If $\{R_1, \ldots, R_d\}$ is a ρ-overlapping decomposition of V, $\tilde{R}_i = \{v \in R_i : N_\rho(v) \subseteq R_i\}$ for all $i = 1, \ldots, d$, and $\tilde{\mathcal{F}}_v = \bigcap_{w \in N_\rho(v)} \mathcal{F}_w$, then for all $i = 1, \ldots, d$: $\bigcap_{v \in R_i} \mathcal{F}_v \subseteq \bigcap_{v \in \tilde{R}_i} \tilde{\mathcal{F}}_v$, with equality holding if $R_i = \bigcup_{v \in \tilde{R}_i} N_\rho(v)$.

Proof: From the definition of $\tilde{R}_i$, we know that $(\forall v \in \tilde{R}_i)$: $N_\rho(v) \subseteq R_i$, and hence, $\bigcup_{v \in \tilde{R}_i} N_\rho(v) \subseteq R_i$. Therefore

$$
\bigcap_{w \in R_i} \mathcal{F}_w \subseteq \bigcap_{w \in \left(\bigcup_{v \in \tilde{R}_i} N_\rho(v) \right)} \mathcal{F}_w
$$

and the equality holds when $R_i = \bigcup_{v\in\tilde{R}_i} N_\rho(v)$. Now

$$\bigcap_{w\in\left(\bigcup_{v\in\tilde{R}_i} N_\rho(v)\right)} \mathcal{F}_w = \bigcap_{v\in\tilde{R}_i}\left(\bigcap_{w\in N_\rho(v)} \mathcal{F}_w\right) = \bigcap_{v\in\tilde{R}_i} \tilde{\mathcal{F}}_w.$$

□

It should be noted that while the transformation from ρ-MORDP to 0-MORDP and back to ρ-MORDP may create a different ρ-overlapping decomposition, the cardinality of the decomposition under this two-step transformation will remain the same, hence, the optimality will not be affected.

Theorem 4.1: If $D = \{R_1, \ldots, R_d\}$ is a ρ-overlapping decomposition of V for a world instance $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v\in V}\rangle$, such that $|\bigcap_{v\in R_i} \mathcal{F}_v| \geq k$ for all $i = 1, \ldots, d$, then $\tilde{D} = \{\tilde{R}_1, \ldots, \tilde{R}_d\}$, where $\tilde{R}_i = \{v \in R_i : N_\rho(v) \subseteq R_i\}$ is a 0-overlapping decomposition for a world instance $\langle G = (V, E), F, \{\tilde{\mathcal{F}}_v\}_{v\in V}\rangle$, where $\tilde{\mathcal{F}}_v = \bigcap_{w\in N_\rho(v)} \mathcal{F}_w$, such that $|\bigcap_{v\in\tilde{R}_i} \tilde{\mathcal{F}}_v| \geq k$ for all $i = 1, \ldots, d$.

Proof: According to *Lemma 4.1*, we know that $\tilde{D}$ is a 0-overlapping decomposition of V. By *Lemma 4.3*, we know that $\bigcap_{v\in R_i} \mathcal{F}_v \subseteq \bigcap_{v\in\tilde{R}_i} \tilde{\mathcal{F}}_v$, for all $i = 1, \ldots, d$. Therefore, $|\bigcap_{v\in\tilde{R}_i} \tilde{\mathcal{F}}_v| \geq |\bigcap_{v\in R_i} \mathcal{F}_v| \geq k$, for all $i = 1, \ldots, d$. □

Theorem 4.2: If $\tilde{D} = \{\tilde{R}_1, \ldots, \tilde{R}_d\}$ is a solution to 0-MORDP for a world instance $\langle G = (V, E), F, \{\tilde{\mathcal{F}}_v\}_{v\in V}\rangle$, then $D' = \{R'_1, \ldots, R'_d\}$, where $R'_i = \bigcup_{v\in\tilde{R}_i} N_\rho(v)$ is a solution to ρ-MORDP for the world instance $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v\in V}\rangle$.

Proof: We have to show the following.

1) D' is a ρ-overlapping decomposition of V, i.e., $(\forall v \in V)(\exists i)$: $N_\rho(v) \subset R'_i$. (This is direct from *Lemma 4.2*.).
2) $|\bigcap_{v\in R'_i} \mathcal{F}_v| \geq k$ for all $i = 1, \ldots, d$. (Direct from *Lemma 4.3* and the facts that $\tilde{D}$ is a 0-MORDP solution, and $R'_i = \bigcup_{v\in\tilde{R}_i} N_\rho(v)$.).
3) D' is minimum size.

   (We will prove this by contradiction. We will suppose that there is solution D'' to ρ-MORDP that has size less than D', and will show that from this, we can construct a 0-MORDP decomposition $\tilde{D}''$ for the original problem of size smaller than $\tilde{D}$ with the property $|\bigcap_{v\in\tilde{R}''_i} \tilde{\mathcal{F}}_v| \geq k$. This is a contradiction, since $\tilde{D}$ was a decomposition of minimum size with that property.

   Suppose $D'' = \{R''_1, \ldots, R''_h\}$ is a decomposition for the original ρ-overlapping problem, such that $h < t$ and $|\bigcap_{v\in R''_i} \mathcal{F}_v| \geq k$ for all $i = 1, \ldots, h$.

   Let $\tilde{D}'' = \{\tilde{R}''_1, \ldots, \tilde{R}''_h\}$, where $\tilde{R}''_i = \{v \in R''_i : N_\rho(v) \subseteq R''_i\}$ for all $i = 1, \ldots, h$. By *Lemma 4.1*, we know that $\tilde{D}''$ is a 0-overlapping decomposition of V, and by *Lemma 4.3*, we can affirm that $|\bigcap_{v\in\tilde{R}''_i} \tilde{\mathcal{F}}_v| \geq |\bigcap_{v\in R''_i} \mathcal{F}_v| \geq k$.)

□

The transformation applied in *Theorem 4.1* from a ρ-overlapping to a 0-overlapping solution effectively shrinks the regions of D by ρ, and reduces the visibility set of each vertex v to correspond to only those features that are visible over the entire neighborhood $N_\rho(v)$ of v.[2] *Theorem 4.2* assumes that the collection of visibility sets $\tilde{\mathcal{F}}$ input to 0-MORDP is defined by a reduction of the ρ-overlapping instance of the problem to a 0-overlapping instance, using the transformation described in *Theorem 4.1*.

$$\begin{aligned} U &= \{A, B, C, D\} \\ S &= \{\{A, B\}, \{C\}, \\ &\quad \{A, D\}, \{C, D\}\} \end{aligned}$$

Fig. 2. Instance of the MSCP.

A. 0-MORDP is NP-Complete

We will now show that 0-MORDP is NP-complete. The proof is by reduction from the minimum set cover problem (MSCP).

Definition 4.1: Given a set U, and a set of subsets $S = \{S_1, \ldots, S_m\}$ of U, the MSCP consists of finding a minimum set $C \subset S$ such that each element of U is covered at least once, i.e., $\bigcup_{S_i\in C} S_i = U$.

Fig. 2 presents an instance of the MSCP. The optimal solution for this instance is $C = \{\{A, B\}, \{C, D\}\}$ and, in fact, this solution is unique. An instance $\langle U, S, r\rangle$ of the set cover (SC) decision problem, where r is an integer, consists of determining if there is a SC of U, by elements of S, of size at most r. The decision version of SCP was proven to be NP-complete by Karp [39], with the size of the problem measured in terms of $|S|$.

Theorem 4.3: The decision problem $\langle$0-ORDP$, d\rangle$ is NP-complete.

Proof: It is clear that 0-MORDP is in NP, i.e., given a world instance $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v\in V}\rangle$ and $D = \{R_1, \ldots, R_d\}$, it can be verified in time polynomial in $\max(|V|, |F|)$ if D is a ρ-overlapping decomposition of V, such that $\forall i : |\bigcap_{v\in R_i} \mathcal{F}_v| \geq k$. We now show that any instance of SCP can be reduced to an instance of 0-ORDP in time polynomial in $|V|$. Given an instance $\langle U, S = \{S_1, \ldots, S_m\}\rangle$ of the MSCP, we construct a 0-ORDP for the world instance $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v\in V}\rangle$ in the following way:

- let v^* be an element not in U; then $V = U \cup \{v^*\}$;
- $E = \{(u, v^*) : u \in U\}$ (Note that the graph G thus generated is planar);
- $F = \{f_1, \ldots, f_m\}$ where $f_i = S_i \cup \{v^*\}$;
- $\mathcal{F}_v = \{f \in F : v \in f\}$;
- $k = 1$.

The introduction of the dummy vertex v^* will be used in the proof to ensure that elements of U that belong to the same subset S_i can be part of the same region in the decomposition, by virtue of their mutual connection to v^*. Each visibility set $\mathcal{F}_v$ in the transformed problem instance corresponds to a list of the sets S_i in the SCP instance that element v is a member of.

Now we show that from a solution to 0-ORDP of size d, we can build a SC of size d. Let $D = \{R_1, \ldots, R_d\}$ be a solution to the transformed 0-ORDP instance, i.e.,

1) $R_i \subseteq V$ is a region, for $i = 1, \ldots, d$;
2) $\bigcup_{1\leq i\leq d} R_i = V$;
3) $|\bigcap_{v\in R_i} \mathcal{F}_v| \geq k = 1$, for $i = 1, \ldots, d$.

[2]Strictly speaking, the region reduction is impervious to boundary effects at the boundary of G, due to the definition of $N_\rho(v)$.

Claim: $C = \{C_1, \ldots, C_d\}$, with $C_i = \text{first}_{lex}(\bigcap_{v \in R_i} \mathcal{F}_v) - \{v^*\}$, is a SC for the original problem, where $\text{first}_{lex}(A)$ returns the first element in lexicographical order from the nonempty set A. (For each C_i, the choice of an element f from $\bigcap_{v \in R_i} \mathcal{F}_v$ is arbitrary in that any such f yields a valid solution.) Note that C_i is well-defined, since $|\bigcap_{v \in R_i} \mathcal{F}_v| \geq 1$.

Proof: We must show the following.

1) $\forall i = 1, \ldots, d$: $C_i \in S$:
   From the definition of C_i, we can affirm that $(\exists j)$: $[1 \leq j \leq m$ and $C_i = f_j - \{v^*\}]$. Hence, $C_i = S_j \in S$.
2) $\bigcup_{1 \leq i \leq d} C_i = U$:
   From the definition of $\mathcal{F}_v$

$$\bigcap_{v \in R_i} \mathcal{F}_v = \bigcap_{v \in R_i} \{f \in F : v \in f\} = \{f \in F : R_i \subseteq f\}.$$

   Therefore, from the definition of C_i

$$\begin{aligned} C_i &= \text{first}_{lex}\{f \in F : R_i \subseteq f\} - \{v^*\} \\ &\Longrightarrow R_i \subseteq C_i \cup \{v^*\} \\ &\Longrightarrow V = \bigcup_{1 \leq i \leq d} R_i \subseteq \bigcup_{1 \leq i \leq d} C_i \cup \{v^*\} \subseteq V \\ &\Longrightarrow \bigcup_{1 \leq i \leq d} C_i \cup \{v^*\} = V \\ &\Longrightarrow \bigcup_{1 \leq i \leq d} C_i = V - \{v^*\} = U. \end{aligned}$$

Finally, we have to show that if there is a SC of size d, then there is a decomposition of size d for the 0-ORDP. Let $C' = \{C'_1, \ldots, C'_d\}$ be a SC for the original SCP instance.

Claim: $D' = \{R'_1, \ldots, R'_d\}$, where $R'_i = C'_i \cup \{v^*\}$, is a 0-overlapping region decomposition such that $|\bigcap_{v \in R'_i} \mathcal{F}_v| \geq 1$.

Proof: We must show the following.

1) Each $R'_i \subseteq V$ is a region:[3]
   $\forall i : 1 \leq i \leq d$, since $C'_i \subseteq U$, then $R'_i = C'_i \cup \{v^*\} \subseteq V$.
   R'_i is a region because $v^* \in R'_i$ and, by the definition of the graph G, v^* is connected to all other nodes in R_i.
2) $\bigcup_{1 \leq i \leq d} R'_i = V$:

$$\bigcup_{1 \leq i \leq d} R'_i = \bigcup_{1 \leq i \leq d} C'_i \cup \{v^*\} = U \cup \{v^*\} = V.$$

3) $|\bigcap_{v \in R'_i} \mathcal{F}_v| \geq 1$:
   C' is a SC

$$\begin{aligned} &\Longrightarrow \forall i : 1 \leq i \leq d : C'_i \in S \\ &\Longrightarrow \exists j = 1, \ldots, m : C'_i = S_j \\ &\Longrightarrow R'_i = S_j \cup \{v^*\} = f_j \in F \\ &\Longrightarrow 1 \leq |\{f \in F : R'_i \subseteq f\}| \\ &\quad = \left| \bigcap_{v \in R'_i} \{f \in F : v \in f\} \right| = \left| \bigcap_{v \in R'_i} \mathcal{F}_v \right|. \end{aligned}$$

□

[3]Recall that a region corresponds to a subset R of vertices in V, for which a path exists between any two vertices in R that lies entirely within R.

Input: world $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v \in V} \rangle$
Output: decomposition D

```
1: U = {v ∈ V : |F_v| ≥ k}, D = ∅
2: while U ≠ ∅ do
3:    Select v ∈ U (See text)
4:    R = {v}
5:    repeat
6:       W = {u ∈ {N_1(v) : v ∈ R} − R : |F_u ∩ [⋂_{v∈R} F_v]| ≥ k}
7:       if W ≠ ∅ then
8:          if W ∩ U ≠ ∅ then
9:             W := W ∩ U
10:         end if
11:         u = arg max_{w∈W} |F_w ∩ [⋂_{v∈R} F_v]|
12:         R = R ∪ {u}
13:      end if
14:   until W = ∅
15:   U = U − R
16:   D = D ∪ {R} (See Section V-E)
17: end while
```

Fig. 3. Algorithm A.x.

V. Searching for an Approximate Solution

The previous section established the intractability of our problem. Fortunately, the full power of an optimal decomposition is not necessary in practice, for a decomposition with a small number of regions is sufficient for practical purposes. We therefore developed and tested six different greedy approximation algorithms, divided into two classes, to realize the decomposition.

A. Limitations in the Real World

In real-world visibility data, there are usually sampled poses at which the count of visible features is less than the required number k. This is generally the case for poses that lie close to walls and object boundaries, as well as for areas that are located far from any visible object, and are, therefore, beyond the visibility range of most features. For this reason, the set of poses that should be decomposed into regions has to include only the *k-coverable poses,* i.e., those sampled poses whose visibility-set sizes are at least k.

B. Growing Regions From Seeds

The A.x class of algorithms decomposes pose space by greedily growing new regions from poses that are selected according to three different criteria. Once a new region has been started, each growth step consists of adding the pose in the vicinity of the region that has the largest set of visible features in common with the region. This growth is continued until adding a new pose would cause that region's visibility set to have a cardinality less than k.

The pseudocode of this class of algorithms is shown in Fig. 3. Algorithms A.1, A.2, and A.3 implement each of three different criteria for selecting the pose from which a new region is grown. These three algorithms differ only in the implementation of line 3 (Fig. 3).

- A.1 selects the pose $v \in U$ at which the least number of features is visible, i.e., $v = \arg\min_{u \in U} |\mathcal{F}_u|$.
- A.2 selects the pose $v \in U$ at which the greatest number of features is visible, i.e., $v = \arg\max_{u \in U} |\mathcal{F}_u|$.
- A.3 randomly selects a pose $v \in U$.

In cases of ties in line 3, they are broken randomly.

The set U, which is initialized in line 1 of the algorithm, contains the k-coverable poses which are still *unassigned* to some region. The set D that will contain the regions in the achieved decomposition is also initialized to be empty. The main loop starts in line 2, and is executed while there are unassigned poses. In lines 3 and 4, a pose v is selected from U according to the criteria given above, and a new region R containing only v is created. The loop that starts in line 5 adds neighboring poses to the region R, until the addition of a new pose would cause the set of features commonly visible in the region to have cardinality less than k. An iteration of this loop is realized in the following way. In line 6, the set W is formed by all poses u in the vicinity of the region R (i.e., the set of poses not in R that are at distance exactly one from a pose in R), such that u, together with the poses in R commonly see at least k features.

In lines 8 through 10, if W contains unassigned poses, then W is restricted to those poses. Since the region R is going to grow with a pose selected from W, this step is intended to give priority to the growth of R with poses that still have not been assigned to any other region. In lines 11 and 12, the pose from W, that together with the poses in R commonly sees the maximum number of features, is added to R. In case of a tie, it is broken randomly. Finally, in lines 15 and 16, the poses in R are removed from the set of unassigned poses U, and the new region R is added to the decomposition set D.

C. Shrinking Regions Until k Features are Visible

Algorithms B.x and C take an incremental approach to defining the k features, starting with a large region that "sees" one feature, and iteratively shrinking the region as additional features (up to k) are added. The resulting region is added to the decomposition, a new region is started, and the process continued until the world is covered. These algorithms select as a new region the set of poses from which the most widely visible feature, taken from a set $\mathcal{F}$, is seen among the poses that are not yet assigned to a region. Algorithms B.x and C differ in the criteria by which $\mathcal{F}$ is defined, as shown in Figs. 4 and 5, respectively. In the case of algorithm B.x, $\mathcal{F}$ is just the set of all features, while algorithm C systematically selects as $\mathcal{F}$ the set of features commonly visible in a circular area centered at each pose $v \in V$. If the number of unassigned poses in the circular area is less than a certain fraction α of the size of the circular area, or the size of $\mathcal{F}$ is less than k, then no further processing is performed for pose v, and the next pose is processed.

The class B.x comprises two algorithms, B.1 and B.2, that differ only in their treatment of the decomposition D after adding to it a new region R (line 12). While Algorithm B.1 leaves D as it is, Algorithm B.2 greedily eliminates regions from D as long as the total number of poses that become unassigned, after the regions are removed from D, is less than the number of cells that the recently added region R has covered but were unassigned before.[4] This discarding rule is adapted from the algorithm "Altgreedy," appearing in [40], where it is empirically shown to achieve very good approximation results for the SC problem.

[4]Notice that this discarding rule ensures that the number of poses assigned to regions strictly increases with each iteration, so that the algorithm always terminates.

Input: world $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v \in V} \rangle$
Output: decomposition D
1: $U = \{v \in V : |\mathcal{F}_v| \geq k\}, D = \emptyset$
2: **while** $U \neq \emptyset$ **do**
3: $R = U, L = \emptyset$
4: **for** $i = 1$ to k **do**
5: $f = \arg\max_{\phi \in (F - L)} |\{v \in R : \phi \in \mathcal{F}_v\}|$
6: $R = \{v \in R : f \in \mathcal{F}_v\}$
7: $L = L \cup \{f\}$
8: **end for**
9: $R = \{v \in V : L \subseteq \mathcal{F}_v\}$
10: $U = U - R$
11: $D = D \cup \{R\}$ (See Section V-E)
12: Purge D (See text)
13: **end while**

Fig. 4. Algorithm B.x.

Input: world $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v \in V} \rangle$
Output: decomposition D
1: $U = \{v \in V : |\mathcal{F}_v| \geq k\}, D = \emptyset$
2: $r = \max\{\rho \in \mathbb{N} : |\{u \in U : |\bigcap_{w \in N_\rho(u) \cap U} \mathcal{F}_w| \geq k\}| \geq \frac{|U|}{2}\}$
3: **for all** $v \in V$ **do**
4: $\mathcal{C} = N_r(v) \cap U$
5: $\mathcal{F} = \bigcap_{u \in \mathcal{C}} \mathcal{F}_u$
6: **if** $\frac{|\mathcal{C}|}{|N_r(v)|} \geq \alpha$ **and** $|\mathcal{F}| \geq k$ **then**
7: $R = U, L = \emptyset$
8: **for** $i = 1$ to k **do**
9: $f = \arg\max_{\phi \in (\mathcal{F} - L)} |\{v \in R : \phi \in \mathcal{F}_v\}|$
10: $R = \{v \in R : f \in \mathcal{F}_v\}$
11: $L = L \cup \{f\}$
12: **end for**
13: $R = \{v \in V : L \subseteq \mathcal{F}_v\}$
14: $U = U - R$
15: $D = D \cup \{R\}$ (See Section V-E)
16: **end if**
17: **end for**

Fig. 5. Algorithm C.

In line 1 of Algorithm B.x, the sets U and D are initialized as in Algorithm A.x. The main loop starts in line 2, and it is executed while there are unassigned poses. In line 3, a new region R is initialized, containing all unassigned poses, and the set L, which will contain features that all poses in the region commonly see, is initialized to be empty. Each iteration of the for-loop in lines 4–8 greedily selects the feature f not in L that is most widely visible in the region R, shrinks R to be formed only by those poses, and extends L to include f. At the exit of the for-loop, which is executed k times, R contains at least one pose (since R entered the loop containing k-coverable poses), and the set L contains the k features that greedily decreased the least the size of the region R. In line 9, R is set to be the set of all poses (not only the unassigned ones) that see at least the k features in L. Finally, in lines 10 and 11, the poses in R are removed from the set of unassigned poses U, and the region R is added to the decomposition D.

Algorithm C, in line 1, initializes the set of unassigned poses U and the decomposition set D in the same way that Algorithms A.x and B.x do. In line 2, the variable r is assigned the maximum natural number such that at least half of the k-coverable poses have an r- neighborhood, such that the k-coverable poses of the neighborhood commonly see at least k features. The main loop of this algorithm starts in line 3, and is executed for every pose $v \in V$. In line 4, $\mathcal{C}$ is assigned the set of unassigned poses in the r-neighborhood of v, and in line 5, $\mathcal{F}$ is assigned the set of features commonly visible in all poses of $\mathcal{C}$. The condition

verified in line 6 requires the proportion of unassigned poses in the r-neighborhood of the current pose v to be greater than or equal to a constant α (defined by the user), and the number of features commonly visible from all unassigned poses in the r-neighborhood of v to be at least k. If this condition is true, then the process continues in a way similar to lines 3–11 of Algorithm B.x. A for-loop greedily select the k "most visible features" from the set of unassigned poses, and finally, a region containing all poses seeing those k features is created. The only difference is in the fact that in the for-loop of this algorithm, the features are greedily selected from the set $\mathcal{F} - L$, while in Algorithm B.x, such features are selected from $F - L$. With this difference, Algorithm C ensures that the for-loop will exit with a region R that has a minimum number (which depends on r and α) of newly assigned poses that are in $N_r(v)$. This algorithm may terminate leaving some poses unassigned to a region. A process (not shown in the pseudocode) is therefore applied to cover those areas. Such a process is equivalent to Algorithm B.1, but with line 1 making U equal to the set of unassigned poses.

Algorithms B.x and C are based on the assumption that the set of poses from which each feature is visible form a connected region, and that the intersection of such feature-visibility areas is also a connected region. This assumption is true if all feature-visibility areas are simple and convex. In our experiments with real data, we have observed that the feature-visibility regions are not always convex or connected, and that they sometimes have some small holes. Since the number of extracted features is quite large, we can afford to exclude from the decomposition process those features with significant holes in their visibility regions. Visibility regions with many concavities can also be trimmed to the set of poses that have a more or less convex shape. Also, if a visibility region has more than one connected component, each component of significant size can be considered to be the visibility region of a different feature.

D. Elimination of Redundant Regions

All algorithms, except B.2, can terminate with a solution that is not minimal. Redundancy is, therefore, eliminated from their solutions by discarding regions one by one until a minimal solution is obtained. This process greedily selects for elimination the region R from the solution D with the largest *minimum-overlapping-count* ω value, where $\omega = \min\{|\{R' \in D : v \in R'\}| : v \in R\}$, i.e., it is the minimum number of regions that overlap at a pose contained in the region. The worst-case running-time complexity of Algorithm A.x is bounded by $O(|V|^2|F|)$, while Algorithms B.x and C are bounded by $O(k|V|^2|F|)$.

E. Relaxing the Requirement for a Complete Decomposition

A decomposition that tries to cover all k-coverable poses may include a large number of regions in total, since many regions will serve only to cover small "holes" that could not be otherwise covered by larger regions. These holes generally lie in areas for which the size of the visibility set is very close to k, leaving very few features to choose from. In order to avoid the inclusion of regions that are only covering small holes, our implementations of the algorithms add a region to the decomposition only if its number of otherwise uncovered poses is greater than a certain value σ.[5]

TABLE I
PARAMETERS OF A WORLD

Component	Parameters
Perimeter	• Sides count • Vertex radius
Obstacles	• Total obstacles count • Sides count • Vertex radius
Features	• Total features count • Visibility angular extent • Visibility range

VI. RESULTS

We performed experiments on both synthetic and real visibility data. Synthetic data was produced using a simulator that randomly generates worlds, given a mixture of probability distributions for each of the defining parameters of the world (see Table I). A world consists of a 2-D top view of the pose space defined by a polygon, with internal polygonal obstacles and a collection of features on the polygons (both external and internal). Each feature is defined by two parameters, an angle (*visibility angle extent*), and a range of visibility (*visibility range*), determining the span of the area on the floor from which the feature is visible. An example of a randomly generated world and the visibility area of some of its features is illustrated in Fig. 6.

A. Decomposition of Synthetic Worlds

Independent tests of the algorithms on synthetic data were performed for four different world settings. The settings combined different feature-visibility properties with different shape complexities for the world and obstacle boundaries. Two types of features were used, having visibility ranges $\mathcal{N}(0.65, 0.2)$ to $\mathcal{N}(12.5, 1)$ m, with an angular range $\mathcal{N}(25, 3)$ degrees for Type 1, and $\mathcal{N}(0.65, 0.2)$ to $\mathcal{N}(17.5, 2)$ m with an angular range $\mathcal{N}(45, 4)$ degrees for Type 2 (where $\mathcal{N}(\mu, \sigma)$ is normally distributed with mean μ and variance σ^2). Two classes of shapes were tested for the world and obstacles: irregular and rectangular. For the case of irregular worlds, the number of sides of its perimeter was generated from the mixture distribution $\{\mathcal{U}(4, 4)$ with $p = 0.1$; $\mathcal{N}(5, 0.5)$ with $p = 0.45$; $\mathcal{N}(7, 2)$ with $p = 0.45]\}$, and the number of obstacles from the distribution $\{\mathcal{U}(5, 9)$ with $p = 0.5$; $\mathcal{N}(8, 2)$ with $p = 0.5\}$. The number of obstacles in each rectangular world was generated from the mixture distribution $\{\mathcal{U}(6, 9)$ with $p = 0.5$; $\mathcal{N}(10, 2)$ with $p = 0.5\}$. The generated worlds had an average diameter of 40 m, and feature visibility was sampled in pose space at points spaced at 50-cm intervals.

The parameters used in the experiments were overlapping $\rho = 1$, and features commonly visible per region $k = 4$. (Basri and Rivlin [5] showed that reliable localization can be accomplished using their linear combination of model views method with as few as three point correspondences between the current

[5]The presence of a few small holes does not prevent reliable navigation. In general, whenever the robot is at a point for which the number of visible features is less than k, advancing a short distance in most directions will get it to a point that is assigned to some region.

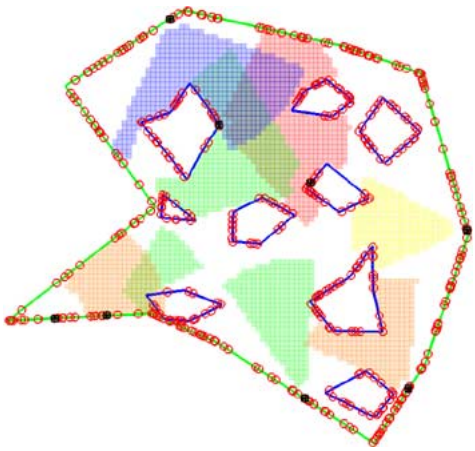

Fig. 6. Randomly generated world. The green polygon defines the perimeter of the world. The blue polygons in the interior define the boundaries of obstacles. The small red circles on the polygons are the features. As an illustration, the visibility areas of selected features are shown as colored regions.

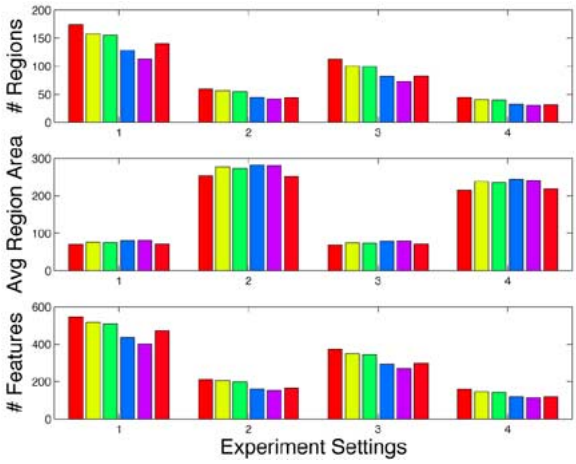

Fig. 7. Results for experiments on synthetic data. The x axes of the charts represent the four world settings used in the experiments. (Rectangular worlds were used in settings 1 and 2, while irregularly shaped worlds were used in settings 3 and 4. Type 1 features were used in settings 1 and 3, and Type 2 features in settings 2 and 4.) The y axes correspond to the average value of 300 experiments for the total number of regions, average number of poses per region, and total number of used features in each decomposition. From left to right, the bars at each setting correspond to Algorithms A.1, A.2, A.3, B.1, B.2, and C.

image and two stored model views.) The allowed maximum area of a hole was set to $\sigma = 9$ poses, i.e., on average, a hole has a diameter of at most 1.5 m. The parameter α of algorithm C was set to 0.85.

Fig. 7 shows the results of the experiments on synthetic data. The performance of each algorithm in the four settings described above is compared in terms of the number of regions in the decomposition, the average area of a region in a decomposition, and the size of the set formed by the union of the k features commonly visible from each region in a decomposition. Each value in the figure is the average computed over a set of 300 randomly generated worlds. The decomposition of each world took only a few seconds for each algorithm.

Unsurprisingly, the average size of a region is strongly dependent on the stability of its defining features in the pose space. Also as expected, the total number of regions in each decomposition increases as the average size of the regions decreases. Tables II and III show the number of regions and the average

TABLE II
AVERAGE NUMBER OF REGIONS IN A DECOMPOSITION

Setting	A.1	A.2	A.3	B.1	B.2	C
1	173.81	156.96	154.97	127.76	112.63	140.10
2	59.30	56.45	54.72	44.74	42.10	44.17
3	112.40	100.46	98.97	82.11	73.08	82.29
4	44.71	40.00	39.11	31.99	30.02	31.11

TABLE III
AVERAGE NUMBER OF POSES PER REGION

Setting	A.1	A.2	A.3	B.1	B.2	C
1	70.76	76.49	75.74	80.60	80.99	71.85
2	253.88	276.37	272.83	281.63	279.81	251.86
3	69.04	74.60	73.95	78.63	79.29	71.61
4	215.15	237.68	234.67	244.44	241.26	218.56

TABLE IV
AVERAGE NUMBER OF FEATURES VISIBLE FROM A POSE

Setting	Average Number of Features
1	30
2	95
3	41
4	117

number of poses in a region, respectively, achieved by each algorithm and setting, averaged over all the randomly generated worlds. In the case of worlds with widely visible features (settings 2 and 4), the best results, in terms of minimum number of regions in the decomposition, are achieved by Algorithm B.2, closely followed by Algorithms B.1 and C. For the worlds with less visible features (settings 1 and 3), Algorithm B.2 outperformed the rest.

In our simulations, we obtained fairly big regions, as seen in Table III. Each pose corresponds to a sampled area of 0.25 m^2 (50 cm by 50 cm), so the averages achieved by the best algorithm correspond to region areas of 20 m^2 for features of Type 1, and 65 m^2 for features of Type 2. These results were achieved with only a few features visible per pose, as shown in Table IV, where the average number of features visible per pose was on the order of 100. In real image data, however, the number of stable features visible per pose is on the order of several hundred, and each feature has a visibility range close to that of our simulated features of Type 1 (see [1], for example). These findings lead us to predict that this technique will successfully find decompositions useful for robot navigation in real environments.

B. Region Decomposition Using Real Data

We took Algorithm B.2, the algorithm that achieved the best results on synthetic data, and as a further evaluation, we applied it to real visibility data acquired in a 6 m×3 m grid sampled at 25 cm, (i.e., a lattice of 25×12 poses), from Room 408, McConnell Engineering Building, at McGill University (Montreal, QC, Canada). Images were taken with the robot's camera orientation fixed in four different orientations at 0°, 90°, 180°, and 270°. Each image's position was measured using a laser tracker and a target mounted on the robot [41]. Fig. 8 shows two images of the employed dataset where the variation in image scale can be appreciated. The images correspond to poses that are furthest front and furthest back along the 270° orientation, respectively.

Fig. 8. Examples of the images used in the experiments on visibility data collected in a 6 m×3 m space.

We extracted SIFT features from the images in the dataset using Lowe's implementation [42]. On average, about 420 SIFT feature vectors were extracted from each image. We then used the method proposed in [43] to match the feature vectors from different images, and to discard those that were ambiguous.

As a result of the matching process, an equivalence relation among the image features was constructed. Specifically, two features from different images were placed in the same *equivalence class* iff they are close to each other in the feature space. As a result, SIFT features in each class of this partition correspond to the same scene feature. Less distinctive image features, i.e., features for which pertinence to a class cannot be unambiguously decided, were discarded in the following way. If the distance in the feature space between two image features belonging to different classes was not significantly larger than the radius of the classes in the feature space, all image features belonging to both classes were eliminated. We also removed those features which were not widely visible, i.e., visible from a certain minimum number of poses. This heuristic reduces the complexity of the feature decomposition by eliminating unstable/ambiguous features.

Following this step, we ended up with a total of 897 classes of features, each feature visible from at least 16 poses. An example of the typical feature-visibility regions that we obtained after we ran the feature-matching algorithm proposed in [43] can be seen in Fig. 9. Each of these images represents the visibility region of a particular feature in the 25×12 pose sampling grid. Each thumbnail corresponds to the appearance of a 30×30 pixels context around the feature point extracted from the image taken at the corresponding grid position in the pose space.

From the set of distinctive features that remained after the grouping into classes, we only retained those that were widely and consistently visible, that is, those that were visible from at least 16 poses, whose visibility regions had few small holes, and that contained at least one connected component of at least 3×3 poses. The set of poses of each of these feature-visibility regions was further reduced to a subset that had a fairly convex shape. This was achieved by first retaining only the poses in the largest connected component of the visibility region. Secondly, poses were then removed from this component, which did not have a neighbor with at least seven out of eight of its neighbor poses in the region. After these steps, the feature-visibility regions of each class not only reduced in size, but the total number of image feature classes decreased to 554, since many of the visibility regions became empty as a result of the filtering process. Fig. 10 shows the distribution of feature-visibility regions by size before and after this filtering process. The visibility regions, after filtering, had an average size of 33 poses, and a median size of 23.

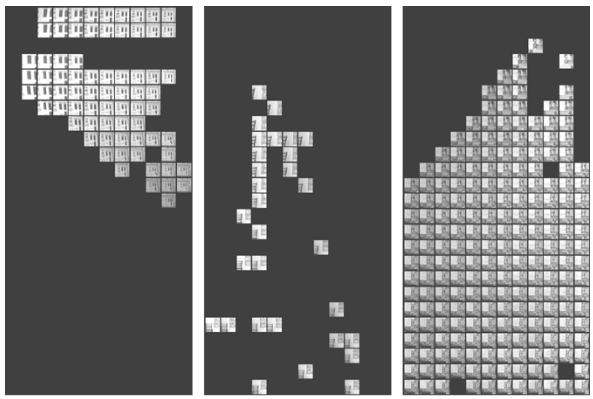

Fig. 9. Typical examples of feature-visibility regions obtained after executing the feature-matching algorithm in [43].

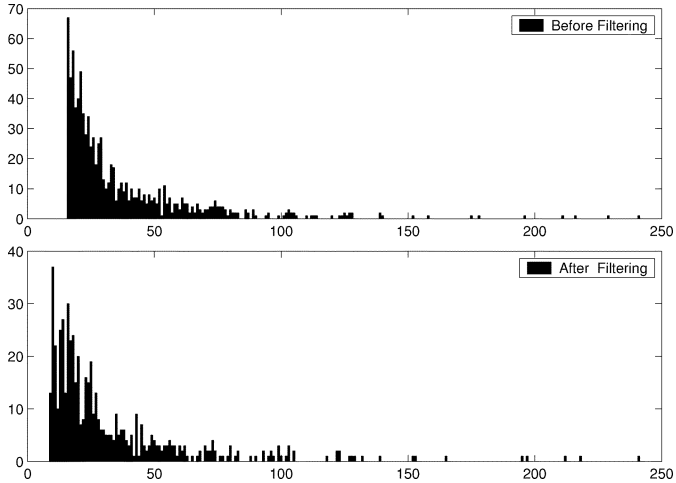

Fig. 10. Distribution of feature-visibility regions by size (i.e., number of poses).

In Fig. 11, we can see the seven regions obtained in the decomposition when we used these visibility regions as input to Algorithm B.2, using parameters $k = 4$, $\rho = 0$, and $\sigma = 3$. (Each region of the decomposition is shown as a separate bird's-eye image of the pose space with the poses of the region colored black, and the k-coverable area of the pose space colored in a lighter gray.) The decomposition obtained using these same parameters but with $\rho = 1$ has nine regions, as shown in Fig. 12. The decompositions obtained when using the value 10 for k, and 0 and 1 for ρ, can be seen in Figs. 13 and 14. As expected, the decompositions for larger values of k contain a larger number of regions of smaller size. As an example of this, notice that some of the regions in Fig. 13 are too small or irregularly shaped, and therefore do not seem useful for navigation purposes. It can also be observed in the figures that the 1-overlapping decompositions have a larger number of regions than when no overlapping is required. It is interesting to note that the regions of the 1-overlapping decompositions are generally more regularly shaped than their 0-overlapping counterparts. This is a natural consequence of the method used to obtain these types of decompositions, which imposes a minimum diameter on the obtained regions. As an example, compare Figs. 13 and 14, in which the regions in the 1-overlapping decomposition are more suitable for navigation than those obtained for $\rho = 0$.

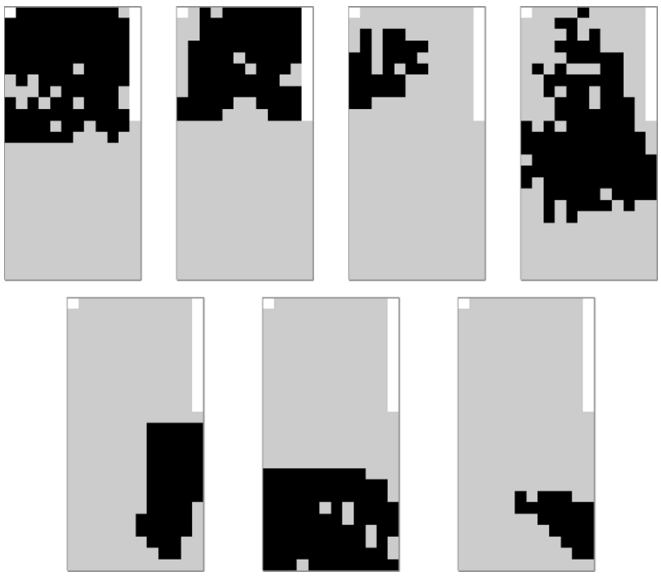

Fig. 11. Region decomposition of the 6 m×3 m real world for $k = 4$ and $\rho = 0$ using Algorithm B.x.

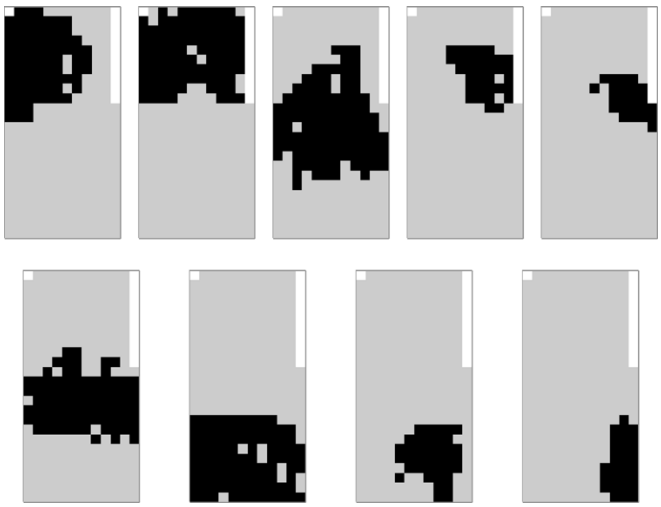

Fig. 12. Region decomposition of the 6 m×3 m real world for $k = 4$ and $\rho = 1$ using Algorithm B.x.

C. Metric Localization Using Region Decomposition

In order to demonstrate the utility of our approach for navigation, we applied the decompositions computed in the previous experiment to the problem of robot pose estimation. Specifically, for each decomposition, we computed a visual map using the framework described by Sim and Dudek in [11], and subsequently computed localization estimates (specifically, probability distributions over the pose space) for a set of test observations. The visual map framework computes generative models of feature behavior as a function of viewing position, and can represent a wide variety of feature properties. Please refer to the cited paper for further details of the representation. The total number of SIFT features encoded in the visual map for each decomposition, and its required storage size on disk, are shown in the second and third columns of Table V. In addition, for baseline comparison, we computed a visual map using all 554 features detected in the undecomposed map.

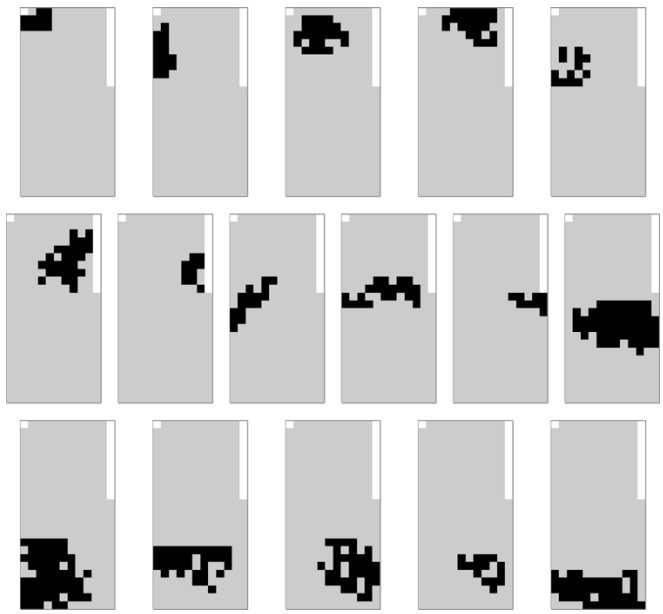

Fig. 13. Region decomposition of the 6 m×3 m real world for $k = 10$ and $\rho = 0$ using Algorithm B.x.

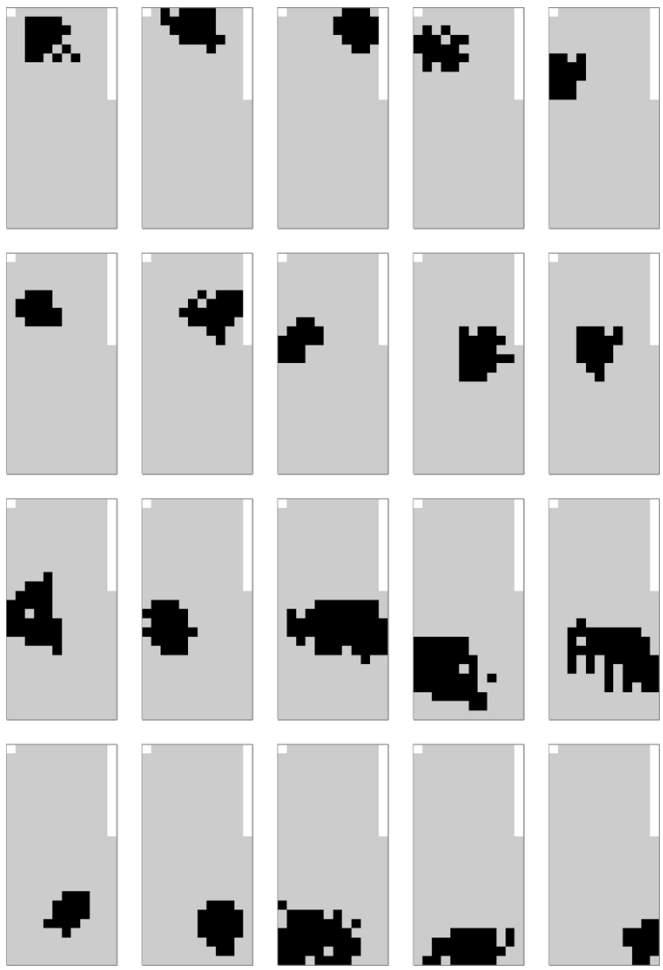

Fig. 14. Region decomposition of the 6 m×3 m real world for $k = 10$ and $\rho = 1$ using Algorithm B.x.

Once the visual maps were computed, we collected 93 test observations from random poses distributed over the environment

TABLE V
VISUAL MAP RESULTS

Decomposition	Visual Map		Mean Localization Error (cms.)	Localization Time (secs/pose)
	Feature Count	Size (Mb)		
All Features	554	9.2	18.66	25.91
$k = 10,\ \rho = 0$	72	3.1	29.75	10.56
$k = 10,\ \rho = 1$	85	3.4	31.34	11.02
$k = 4,\ \rho = 0$	23	1.5	64.48	5.82
$k = 4,\ \rho = 1$	28	1.6	50.42	6.72

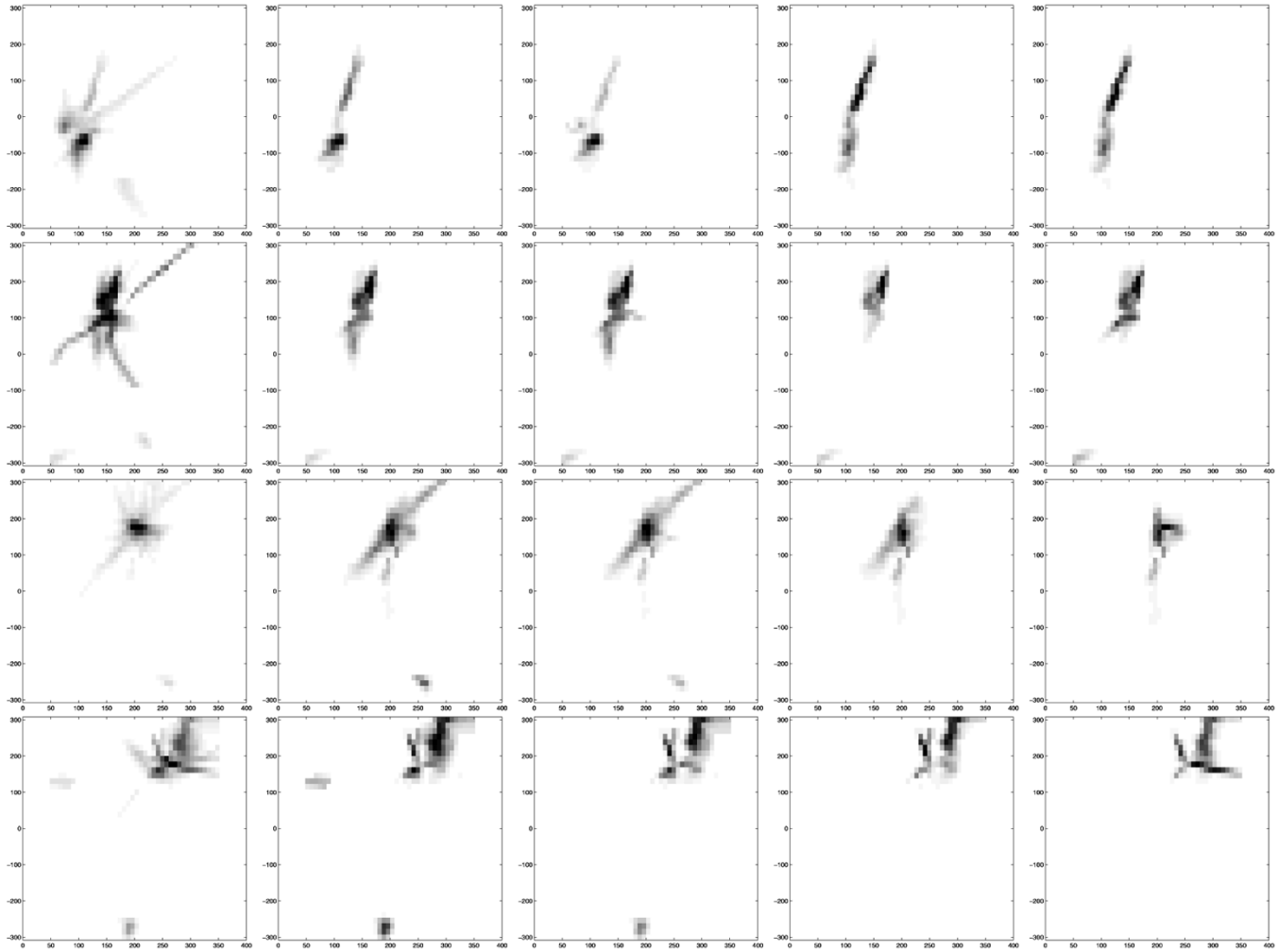

Fig. 15. Probability distribution of some chosen test poses. The rows correspond to different poses and the columns from left to right correspond to the estimates computed using all features, $(k = 10, \rho = 0)$, $(k = 10, \rho = 1)$, $(k = 4, \rho = 0)$, and $(k = 4, \rho = 1)$. The x and y axes of each plot span a space of 3m by 6m (as in Fig. 16).

(these were novel observations that were not used in the training phase). The ground-truth positions of these observations were estimated using a robot-mounted target and laser range finder, as described in [41]. Once these observations were collected, SIFT features were extracted, and matches were found in each of the visual maps. Note that in this step, we are reducing the number of features that could potentially be used to localize the observation from around 450 (the typical number of SIFT features in the image), to approximately k, depending on match quality and region overlap. In these experiments, k is 4 or 10, resulting in a compression level of 97.5%–99%.

Once features matching the map features were detected in the test images, the image position and SIFT scale of each feature were then employed to compute probability distributions over the pose space, indicating the probability of a pose x, given the observation z

$$p(x|z) \propto p(z|x)p(x)$$

where $p(x)$ is a uniform prior distribution over the pose space. The details of computing the observation likelihood $p(z|x)$ are also provided in [11]. Some example distributions are plotted in Fig. 15. The absolute localization results are shown in Fig. 16, plotting each ground-truth pose as an "o," connected with a line segment to the estimated pose, plotted as an "x." For each map, the mean distance between the maximum-likelihood pose estimate and ground truth as provided by the laser tracker is shown in the fourth column of Table V. The mean time employed for localization of the 93 test poses is shown in the last column of Table V for each decomposition. These times include the feature detection and matching.

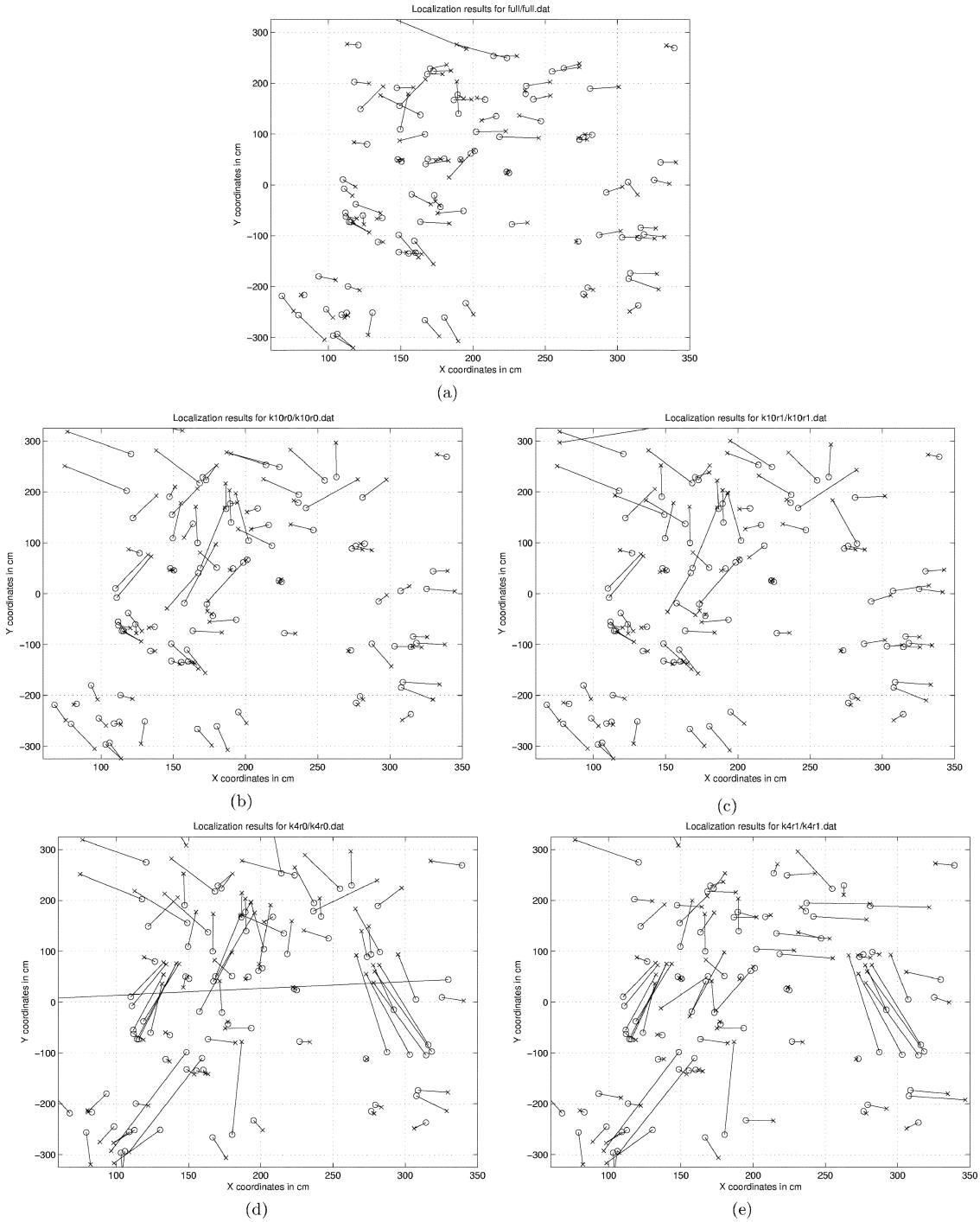

Fig. 16. Ground truth (o) versus estimated (x) poses using: (a) All features. (b) $k = 10, \rho = 0$. (c) $k = 10, \rho = 1$. (d) $k = 4, \rho = 0$. (e) $k = 4, \rho = 1$.

In addition to absolute error, we are also interested in the difference between the decomposition-based pose probability distributions and the pose distributions based on the complete set of landmarks. To measure this difference, we computed the KL-divergence between each decomposition distribution and the baseline distribution. Let Y be the set of all test poses. For each test pose $y \in Y$, and each decomposition d, the KL-divergence was computed as

$$D_y^d = \sum_x p_y^d(x) \log \left(\frac{p_y^d(x)}{q_y(x)} \right)$$

where $p_y^d(x)$ is the probability distribution of y when computed using the features of decomposition d, and $q_y(x)$ is the probability distribution of y computed from the complete set of features. Notice that there are grid poses x that have value zero in distribution $q_y(x)$ but nonzero in $p_y^d(x)$, making the divergence infinity. To deal with this problem, we mixed a uniform distribution with a small weight with each distribution before computing the divergence measure. This is a reasonable regularization procedure, because it should never really be the case that a grid pose has identically zero probability. For each decomposition d, we computed the mean and standard deviation of the set of KL-divergences $\{D_y^d\}_{y \in Y}$ between the decomposition distri-

TABLE VI
KL-DIVERGENCE BETWEEN BASELINE AND DECOMPOSITION DISTRIBUTIONS

Decomposition	Mean of KL-Divergence	Std. Deviation of KL-Divergence
$k = 10,\ \rho = 0$	0.3472	0.1355
$k = 10,\ \rho = 1$	0.3561	0.1595
$k = 4,\ \rho = 0$	0.6522	0.2678
$k = 4,\ \rho = 1$	0.5943	0.2457

bution and the baseline distribution of test poses. These values are shown in Table VI.

From these results, it can be seen that while there is some degradation in the quality of the pose estimates as k decreases, the decomposition-based estimates remain sufficiently robust to successfully localize. Furthermore, navigation, as well as improved pose estimates, can be achieved by computing the $p(x|z)$ using a Markov chain and a model of the robot's motion [15], [44]. It should be noted that in this work, initial landmark selection was based on the tracking stability and viewing range of the selected features. The localization results presented here could be improved by adding additional criteria to the decomposition framework, such as selecting features that provide improved constraints for localization (for example, selecting features whose image-domain observations are expected to be widely separated). The key result of this experiment is that a high degree of map compression can be achieved with only a small degradation in the localization performance.

VII. CONCLUSIONS AND FUTURE WORK

We have presented a novel graph theoretic formulation of the problem of automatically extracting an optimal set of landmarks from an environment for visual navigation. Its intractable complexity (which we prove) motivates the need for approximation algorithms, and we present six such algorithms. To systematically evaluate them, we first test them on a simulator, where we can vary the shape of the world, the number and shape of obstacles, the distribution of the features, and the visibility of the features. The algorithm that achieved the best results on synthetic data was then demonstrated on real visibility data. The resulting decompositions find large regions in the world in which a small number of features can be tracked to support efficient online localization. Our formulation and solution of the problem are general, and can accommodate other classes of image features.

There are a number of extensions to this work for future research:

- integrating the image-collection phase with the region-decomposition stage into a unique online process as the robot is exploring its environment, in a view-based SLAM fashion;
- path planning through decomposition space, minimizing the number of region transitions in a path;
- extending the proposed framework to detect and cope with environmental change;
- computing the performance guarantee of our heuristic methods and providing tight upper bounds on the quality of our solution compared with those of optimal decompositions;
- studying the use of feature tracking during the image-collection stage to achieve larger areas of visibility for each feature, since tracking the features between images taken from adjacent viewpoints allows for tracking small variations of appearance (which may integrate to large ones over large areas). Such a framework would require maintaining equivalence classes of features in the database;
- adding constraints to the algorithms for feature selection in terms of a quality measure of the feature reliability for localization.

ACKNOWLEDGMENT

The authors wish to thank the many individuals (since 1994) who have contributed to earlier versions of this work, including D. Wilkes, E. Rivlin, R. Basri, L. Cowan, J. Ezick, D. Rosenberg, C. Klivans, S. Abbasi, and S. Baqir. They would also like to thank A. Jepson for his careful reading of and corrections to earlier drafts of this paper.

REFERENCES

[1] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kerkyra, Greece, Sep. 1999, pp. 1150–1157.
[2] G. Dudek and D. Jugessur, "Robust place recognition using local appearance-based methods," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, Apr. 2000, pp. 1030–1035.
[3] G. Carneiro and A. D. Jepson, "Multi-scale phase-based local features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Madison, WI, Jun. 2003, pp. 736–743.
[4] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. Eur. Conf. Comput. Vis.*, Copenhagen, Denmark, 2002, pp. 128–142.
[5] R. Basri and E. Rivlin, "Localization and homing using combinations of model views," *Artif. Intell.*, vol. 78, no. 1–2, pp. 327–354, Oct. 1995.
[6] D. Wilkes, S. Dickinson, E. Rivlin, and R. Basri, "Navigation based on a network of 2D images," in *Proc. ICPR-A*, 1994, pp. 373–378.
[7] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, Haifa, Israel, Mar. 1988, pp. 147–151.
[8] E. Krotkov, "Mobile robot localization using a single image," in *Proc. IEEE Int. Conf. Robot. Autom.*, Scottsdale, AZ, 1989, pp. 978–983.
[9] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, 1991.
[10] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int. J. Robot. Res.*, vol. 21, no. 8, pp. 735–758, Aug. 2002.
[11] R. Sim and G. Dudek, "Learning generative models of scene features," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 45–61, Oct. 2004.
[12] N. Fairfield and B. Maxwell, "Mobile robot localization with sparse landmarks," in *Proc. SPIE Workshop Mobile Robots XVI*, Oct. 2001, pp. 148–155.
[13] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 237–267, Feb. 2002.
[14] M. Mata, J. M. Armingol, A. de la Escalera, and F. J. Rodriguez, "A deformable model-based visual system for mobile robot topologic navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, Oct. 2003, pp. 3504–3509.
[15] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 376–382, Jun. 1991.
[16] J. Guivant, E. Nebot, and H. Durrant-Whyte, "Simultaneous localization and map building using natural features in outdoor environments," in *Proc. 6th Int. Conf. Intell. Auton. Syst.*, vol. 1, 2000, pp. 581–588.
[17] J. J. Leonard, B. A. Moran, I. J. Cox, and M. L. Miller, "Underwater sonar data fusion using an efficient multiple hypothesis algorithm," in *Proc. IEEE Conf. Robot. Autom.*, Nagoya, Japan, May 1995, pp. 2995–3002.
[18] K. T. Sutherland and W. B. Thompson, "Inexact navigation," in *Proc. IEEE Conf. Robot. Autom.*, Atlanta, GA, May 1993, pp. 1–7.

[19] C. F. Olson, "Selecting landmarks for localization in natural terrain," *Auton. Robots*, vol. 12, no. 2, pp. 201–210, Mar. 2002.
[20] D. Burschka, J. Geiman, and G. Hager, "Optimal landmark configuration for vision-based control of mobile robots," in *Proc. IEEE Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., 2003, pp. 3917–3922.
[21] A. Yamashita, K. Fujita, T. Kaneko, and H. Asama, "Path and viewpoint planning of mobile robots with multiple observation strategies," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, 2004, pp. 3195–3200.
[22] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*. Boston, MA: Kluwer, 1992.
[23] D. Fox, "Markov localization: A probabilistic framework for mobile robot localization and navigation," Ph.D. dissertation, Univ. Bonn, Dept. Comput. Sci., Bonn, Germany, 1998.
[24] S. Thrun, "Finding landmarks for mobile robot navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Leuven, Belgium, May 1998, pp. 958–963.
[25] A. Davison, "Real-time simultaneous localization and mapping with a single camera," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nice, France, 2003, pp. 1403–1410.
[26] M. Bosse, P. Newman, J. Leonard, and S. Teller, "An atlas framework for scalable mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., Sep. 2003, pp. 1899–1906.
[27] S. Simhon and G. Dudek, "A global topological map formed by local metric maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Victoria, BC, Canada, Oct. 1998, pp. 1708–1714.
[28] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset, "Hierarchical simultaneous localization and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, Oct. 2003, pp. 448–453.
[29] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robot. Auton. Syst.*, vol. 8, no. 1-2, pp. 46–63, Nov. 1991.
[30] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, pp. 125–137, Apr. 2001.
[31] G. Dudek, P. Freedman, and S. Hadjres, "Using multiple models for environmental mapping," *J. Robot. Syst.*, vol. 13, no. 8, pp. 539–559, Aug. 1996.
[32] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. Robot. Res.*, vol. 23, no. 7–8, pp. 693–716, 2004.
[33] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Nat. Conf. Artif. Intell.*, Edmonton, AB, Canada, 2002, pp. 593–598.
[34] M. A. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *Proc. 18th Int. Joint Conf. Artif. Intell.*, G. Gottlob and T. Walsh, Eds., San Francisco, CA, 2003, pp. 1157–1164.
[35] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2743–2760, Jun. 1998.
[36] B. Kröse, N. Vlassis, R. Bunschoten, and Y. Motomura, "Feature selection for appearance-based robot localization," in *Proc. Real World Comput. Symp.*, Tokyo, Japan, Jan. 2000, pp. 223–228.
[37] S. K. Nayar, H. Murase, and S. A. Nene, "Learning, positioning, and tracking visual appearance," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Diego, CA, May 1994, pp. 3237–3246.
[38] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *Int. J. Comput. Vis.*, vol. 27, no. 2, pp. 161–198, 1998.
[39] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. New York: Plenum, 1972, pp. 85–103.
[40] T. Grossman and A. Wool, "Computational experience with approximation algorithms for the set covering problem," *Eur. J. Oper. Res.*, vol. 101, no. 1, pp. 81–92, Aug. 1997.
[41] I. Rekleitis, R. Sim, G. Dudek, and E. Milios, "Collaborative exploration for the construction of visual maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3, Maui, HI, Oct. 2001, pp. 1269–1274.
[42] D. G. Lowe. (2003) Demo Software: SIFT Keypoint Detector. [Online]. Available: http://www.cs.ubc.ca/~lowe/keypoints/
[43] P. Sala, "Selection of an optimal set of landmarks for vision-based navigation," Master's thesis, Univ. Toronto, Toronto, ON, Canada, 2004.
[44] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *J. Artif. Intell. Res.*, vol. 11, pp. 391–427, 1999.

Pablo Sala (S'05) received the Licentiate degree from the University of Buenos Aires, Buenos Aires, Argentina, in 2002, and the M.S. degree in 2004 from the University of Toronto, both in computer science. He is currently working toward the Ph.D. degree in computer science at the University of Toronto, Toronto, ON, Canada.

His research interests are in computer vision and mobile robotics.

Robert Sim (M'96) received the Ph.D. degree in computer science from McGill University, Montreal, QC, Canada, in 2004.

In 2004, he was a Postdoctoral Fellow with the Department of Computer Science, University of Toronto, Toronto, ON, Canada, and he is currently a Postdoctoral Fellow with the Department of Computer Science, University of British Columbia, Vancouver, BC, Canada. His research interests include vision-based mapping and environment representations, autonomous exploration, and machine learning.

Ali Shokoufandeh (M'02) received the Ph.D. and M.Sc. degrees in computer science from Rutgers University, New Brunswick, NJ, in 1999 and 1996, respectively; and the B.Sc. degree in computer science from Tehran University, Tehran, Iran, in 1989.

Currently, he is an Associate Professor of Computer Science with Drexel University, Philadelphia, PA. His research interests are in computer vision and combinatorial optimization.

Sven Dickinson (M'83) received the B.A.Sc. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 1983, and the M.S. and Ph.D. degrees in computer science from the University of Maryland, College Park, in 1988 and 1991, respectively.

He is currently an Associate Professor of Computer Science with the University of Toronto, Toronto, ON, Canada. From 1995–2000, he was an Assistant Professor of Computer Science with Rutgers University, New Brunswick, NJ, where he also held a joint appointment in the Rutgers Center for Cognitive Science (RuCCS) and the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS). From 1994–1995, he was a Research Assistant Professor in the Rutgers Center for Cognitive Science, and from 1991–1994, a Research Associate at the Artificial Intelligence Laboratory, University of Toronto. He has held affiliations with the MIT Media Laboratory (Visiting Scientist, 1992–1994), the University of Toronto (Visiting Assistant Professor, 1994–1997), and the Computer Vision Laboratory of the Center for Automation Research at the University of Maryland (Assistant Research Scientist, 1993–1994, Visiting Assistant Professor, 1994–1997). Prior to his academic career, he was in the computer vision industry, designing image processing systems for Grinnell Systems Inc., San Jose, CA, 1983–1984, and optical character recognition systems for DEST, Inc., Milpitas, CA, 1984–1985. His major field of interest is computer vision with an emphasis on shape representation, object recognition, and mobile robot navigation.

Dr. Dickinson was co-chair of the 1997, 1999, and 2004 IEEE Workshops on Generic Object Recognition, while in 1999, he co-chaired the DIMACS Workshop on Graph Theoretic Methods in Computer Vision. In 1996, he received the NSF CAREER award for his work in generic object recognition, and in 2002, received the Government of Ontario Premiere's Research Excellence Award (PREA), also for his work in generic object recognition. From 1998–2002, he has served as Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, in which he also co-edited a special issue on graph algorithms and computer vision, which appeared in 2001. He currently serves as Associate Editor for the journal *Pattern Recognition Letters*.