Proceedings of 2004 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

Landmark Selection for Vision-Based Navigation

Pablo L. Sala, University of Toronto
Robert Sim, University of British Columbia
Ali Shokoufandeh, Drexel University
Sven J. Dickinson, University of Toronto

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.**

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], [3], [5]. Such features typically encode a description of image appearance in the neighbourhood 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 therefore make them ideally suited to the problem of landmark-based navigation. If we can define a set of invariant features that uniquely define 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 [6] and Wilkes *et al.* [7], based on the LC (*linear combination of views*) 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 runtime, 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. 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's 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 problem addressed by this paper: from a set of views acquired 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 approximation 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 real world imagery of a real environment.

0-7803-8463-6/04/$20.00 ©2004 IEEE

II. Related work

In previous work on visual 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. Lowe, Se and Little [2] use scale- and rotation-invariant features as landmarks, extracted using the scale-invariant feature transform (SIFT) [1]. The robot automatically updates a 3D landmark map with the reliable landmarks seen from the current position. 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 3D locations of landmarks and their regions of confidence, with all reliable landmarks stored in a dense database.

The view-based approach of Sim and Dudek [4] consists of an off-line collection of monocular images sampled over a space of poses. The landmarks consist of PCA encodings of the neighbourhoods of salient points in the images, obtained using an attention operator. Landmarks are tracked between contiguous poses and added to a database if stable through a region of reasonable size. The localization is performed through a linear combination of views technique after finding matches between the visible landmarks and those in the database.

The linear combination of views technique was first introduced by Ullman and Basri, and later applied to vision-based navigation by Basri and Rivlin [6] and Wilkes et al. [7]. In these original applications of the LC method, the features comprising the model views were typically linear features extracted from the image. While all of these approaches 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 tens of thousands of landmark models.

Some authors have considered the problem of landmark selection for the purpose of minimizing uncertainty in the computed pose estimate. Sutherland and Thompson [9] demonstrate that the precision of a pose estimate derived from point features in 2D is dependent on the configuration of the observed features, and provide an algorithm for selecting an appropriate set of observed features from which to compute a pose estimate. While maximizing precision is clearly an important issue, our work is concerned primarily with selecting landmarks that are widely visible.[1]

III. Problem Definition

In an off-line 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 therefore 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 Figure 1. Figure 1 (a) shows a simple 2-D world having a square perimeter, a square obstacle in its center, and eight features evenly distributed along its perimeter. In figures 1 (b) - 1 (g), the area of visibility of some of the features is shown as a coloured 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.

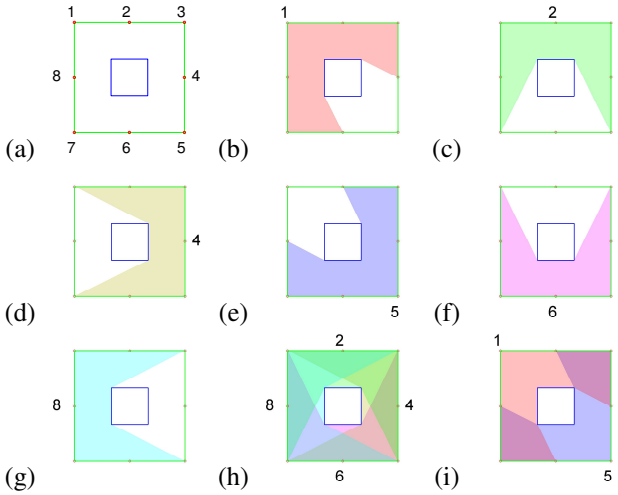


Fig. 1. (a) A simple world with a square perimeter (in green), a square (blue) obstacle in its center and eight features (red circles on its perimeter). (b)-(g) Visibility areas of some features. (h) A covering of the world using 4 features. (i) A covering of the world using 2 features.

In a view-based localization approach, the current pose of the robot is estimated using, as input, the locations of a small 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 linear combination of views technique [6], which uses a weak perspective projection imaging model. The essential matrix method [12], 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 trade-off between the accuracy of the estimation and the size (in features) of the landmark. Requiring a larger number of features for localization, i.e., larger landmarks, will yield better pose estimation. However, the larger and more constrained a landmark is, the smaller its region of visibility becomes. We will define 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

[1]The algorithms presented in this work can be easily extended to select sets of features that fulfill any given additional constraints.

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.

Let's return to the simple world depicted in Figure 1. Assuming that a single ($k = 1$) feature is sufficient for reliable navigation, 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 2, 4, 6, and 8, as shown in Figure 1 (h). 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 the remaining three features are not visible, meaning that the robot is blind in this area. However, this decomposition is not optimal, since other decompositions with less regions are possible. Our goal is to find the minimum decomposition of the world which, in this case, has only two regions, corresponding to the areas of visibility of features 1 and 5, (or its symmetric solution using features 3 and 7), as shown in Figure 1 (i). This minimum set of maximally sized regions is our desired output, and allows us to discard from the database all but features 1 and 5. Since at least one of these two features is seen from every point in pose space, reliable navigation through the entire world is possible.

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.

IV. A Graph Theoretic Formulation

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

Definition 4.1: 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 2D space.

Definition 4.2: 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 4.3: 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 4.4: A set of poses $R \subset V$ is said to be a *region* 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 4.5: A set 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 4.1 to 4.5 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 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 ensured 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 4.6: 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 4.7: 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 stated as follows:

Definition 4.8: 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 into regions, 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 $d \cdot k$.

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 to 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 for 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 $\tilde{D}$. The proofs of these theorems are presented in [8]. 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 $i = 1, \ldots, d$.

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 $\langle G = (V, E), F, \{\mathcal{F}_v\}_{v \in V} \rangle$.

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.

V. Complexity of 0-MORDP

Now we will show that 0-MORDP is NP-complete. The proof is by reduction from the Minimum Set Cover Problem.

Definition 5.1: Given a set U, and a set of subsets $S = \{S_1, \ldots, S_m\}$ of U, the *Minimum Set Cover Problem* (MSCP) consists of finding a minimum set $C \subset S$ such that each element of U is covered at least once, i.e., $\bigcup_{S_i \in C} S_i = U$.

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

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

Fig. 2. An instance of the *Minimum Set Cover Problem*

Figure 2 presents an instance of 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 *decision* problem, where r is an integer, consists of determining if there is a set cover of U, by elements of S, of size at most r. The decision version of SCP was proven to be NP-complete by Karp [10].

Theorem 5.1: 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 $|V|$ 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 Minimum Set Cover Problem, 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) $\bigcup_{1 \leq i \leq d} R_i = V$, and
3) $|\bigcap_{v \in R_i} \mathcal{F}_v| \geq k = 1$, for $i = 1, \ldots, d$.

Claim: $C = \{C_1, \ldots, C_d\}$, with $C_i = \mathsf{first}_{lex}(\bigcap_{v \in R_i} \mathcal{F}_v) - \{v^*\}$ is a Set Cover for the original problem, where $\mathsf{first}_{lex}(A)$ returns the first element in lexicographical order from the non-empty 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 that:

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) $\bigcup_{1\le i\le d} C_i = U$:
From the definition of $\mathcal{F}_v$:

$$\begin{aligned}\bigcap_{v\in R_i} \mathcal{F}_v &= \bigcap_{v\in R_i} \{f \in F : v \in f\} \\ &= \{f \in F : R_i \subseteq f\}\end{aligned}$$

Therefore, from the definition of C_i:

$$\begin{aligned}C_i = \mathsf{first}_{lex}\{f \in F : R_i \subseteq f\} - \{v^*\} \\ \implies R_i \subseteq C_i \cup \{v^*\} \\ \implies V = \bigcup_{1\le i\le d} R_i \subseteq \bigcup_{1\le i\le d} C_i \cup \{v^*\} \subseteq V \\ \implies \bigcup_{1\le i\le d} C_i \cup \{v^*\} = V \\ \implies \bigcup_{1\le i\le d} C_i = V - \{v^*\} = U,\end{aligned}$$

Finally, we have to show that if there is a set cover of size d, then there is a decomposition of size d for the 0-ORDP. Let $C' = \{C'_1, \ldots, C'_d\}$ be a set cover 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 $|\bigcap_{v\in R'_i} \mathcal{F}_v| \ge k$.

Proof: We must show that:

1) Each $R'_i \subseteq V$ is a region[3]:
$\forall i : 1 \le i \le d$, 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) $\bigcup_{1\le i\le d} R'_i = V$:

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

3) $|\bigcap_{v\in R'_i} \mathcal{F}_v| \ge k = 1$:

$$\begin{aligned}&C'_i \text{ is a set cover} \\ &\implies C'_i \in S \\ &\implies \exists j = 1, \ldots, m : C'_i = S_j \\ &\implies R'_i = S_j \cup \{v^*\} = f_j \in F \\ &\implies 1 \le |\{f \in F : R'_i \subseteq f\}| \\ &\qquad = |\bigcap_{v\in R'_i} \{f \in F : v \in f\}| = |\bigcap_{v\in R'_i} \mathcal{F}_v|\end{aligned}$$

□.

VI. Heuristic Methods for 0-ORDP

The previous section established the intractability of our problem. Fortunately, the full power of an optimal decomposition is not necessary in practice. 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.

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 Figure 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 (Figure 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 lines 3 and 12 of the algorithm, they are broken randomly.

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 Figures 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 uncovered 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 set $\mathcal{F}$ is selected for the current v.

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 uncovered is less than a monotonically decreasing value q. (This q is initialized as ∞ at the beginning of the algorithm. As a new region R is added to D, q is updated to be the minimum between its previous value and the number of uncovered cells in R.)[4] This algorithm is adapted from the algorithm "Altgreedy" appearing in [13], where it is empirically shown to achieve very good approximation results for the set cover problem.

[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.

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

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 although the visibility areas of features are generally convex, 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. Algorithm C may terminate leaving some poses unassigned to a region. A process is therefore applied to cover those areas. This process is equivalent to Algorithm B.1, but with step 1 making U equal to the set of unassigned poses.

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 $O(k|V|^2|F|)$.

There are sampled poses of the world 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.

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]

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

$U = \{v \in V : |\mathcal{F}_v| \geq k\}, D = \emptyset$
while $U \neq \emptyset$ **do**
  Select $v \in U$ (See text)
  $R = \{v\}$
  repeat
    $Vicinity(R) = \{v \in V - R : \exists u \in R \text{ s.t. } (u, v) \in E\}$
    $W = \{u \in Vicinity(R) : |\mathcal{F}_u \cap [\bigcap_{v \in R} \mathcal{F}_v]| \geq k\}$
    if $W \neq \emptyset$ **then**
      if $W \cap U \neq \emptyset$ **then**
        $W := W \cap U$
      end if
      $u = \arg\max_{w \in W} |\mathcal{F}_w \cap [\bigcap_{v \in R} \mathcal{F}_v]|$
      $R = R \cup \{u\}$
    end if
  until $W = \emptyset$
  $U = U - R$
  $D = D \cup \{R\}$
end while

Fig. 3. Algorithm A.x

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

$U = \{v \in V : |\mathcal{F}_v| \geq k\}, D = \emptyset$
while $U \neq \emptyset$ **do**
  $R = U, L = \emptyset$
  for $i = 1$ to k **do**
    $f = \arg\max_{\phi \in (F - L)} |\{v \in R : \phi \in \mathcal{F}_v\}|$
    $R = \{v \in R : f \in \mathcal{F}_v\}$
    $L = L \cup \{f\}$
  end for
  $R = \{v \in V : L \subseteq \mathcal{F}_v\}$
  $U = U - R$
  $D = D \cup \{R\}$
  Purge D (See text)
end while

Fig. 4. Algorithm B.x

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

$U = \{v \in V : |\mathcal{F}_v| \geq k\}, D = \emptyset$
$r = \max\{\rho \in \mathbb{N} : |\{u \in U : |\bigcap_{w \in N_\rho(u) \cap U} \mathcal{F}_w| \geq k\}| \geq \frac{|U|}{2}\}$
for all $v \in V$ **do**
  $\mathcal{R} = N_r(v) \cap U$
  $\mathcal{F} = \bigcap_{u \in \mathcal{R}} \mathcal{F}_u$
  if $\frac{|\mathcal{R}|}{|N_r(v)|} \geq \alpha$ and $|\mathcal{F}| \geq k$ **then**
    $R = U, L = \emptyset$
    for $i = 1$ to k **do**
      $f = \arg\max_{\phi \in (\mathcal{F} - L)} |\{v \in R : \phi \in \mathcal{F}_v\}|$
      $R = \{v \in R : f \in \mathcal{F}_v\}$
      $L = L \cup \{f\}$
    end for
    $R = \{v \in V : L \subseteq \mathcal{F}_v\}$
    $U = U - R$
    $D = D \cup \{R\}$
  end if
end for

Fig. 5. Algorithm C

VII. Results

We performed experiments on both synthetic and real data. Synthetic data was produced using a simulator that randomly generates worlds given a set of distributions for each world parameter. 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 and a range of visibility, 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 Figure 6.

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

[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.

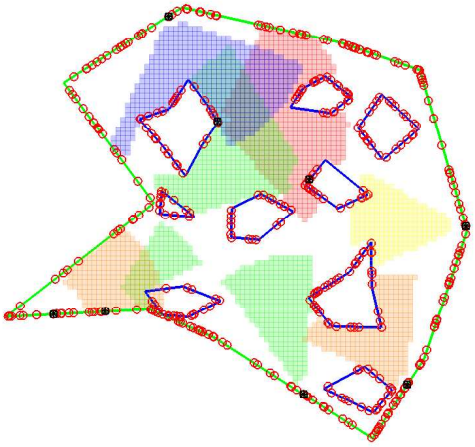


Fig. 6. A randomly generated world. The green polygon defines the perimeter of the world. The blue polygons in the interior define the boundaries of obstacles. The small red circles on the polygons are the features. As an illustration, the visibility areas of selected features are shown as coloured regions.

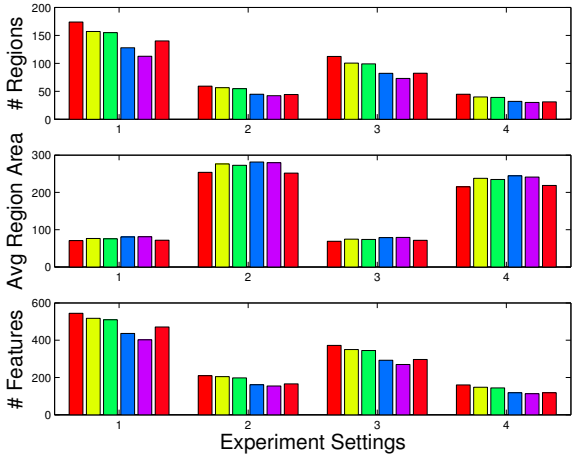


Fig. 7. Results for Experiments on Synthetic Data. The x-axes of the charts represent the four world settings used in the experiments. (Rectangular worlds were used in settings 1 and 2, while irregularly shaped worlds in settings 3 and 4. Type 1 features were used in settings 1 and 3, and Type 2 features 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.

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 40m, and feature visibility was sampled in pose space at points spaced at 50cm intervals.

The parameters used in the experiments were overlapping $\rho = 1$, and features commonly visible per region $k = 4$. (Basri and Rivlin [6] 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 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 1.5m. The parameter α of algorithm C was set to 0.85.

Figure 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 pose space. Also as expected, the total number of regions in each decomposition increases as the average size of the regions decreases. Tables I and II show the number of regions and the average 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.

TABLE I

AVERAGE NUMBER OF REGIONS IN A DECOMPOSITION

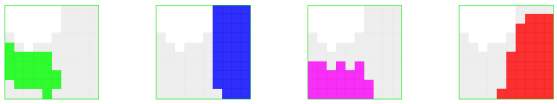
Setting	A.1	A.2	A.3	B.1	B.2	C
1	173.81	156.96	154.97	127.76	112.63	140.10
2	59.30	56.45	54.72	44.74	42.10	44.17
3	112.40	100.46	98.97	82.11	73.08	82.29
4	44.71	40.00	39.11	31.99	30.02	31.11

TABLE II

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

When applied to rectangular worlds, the algorithms produced decompositions with significantly more regions (between 40% and 55% for the top algorithms) than when applied to the irregularly shaped worlds. One of the reasons for this is the fact that considerably more features



(a) (b) (c) (d)

Fig. 8. (a)-(d) The 4 regions of the decomposition of real visibility data collected in a 2m by 2m space, sampled at 20 cm intervals.

are visible, on average, at each pose in irregular worlds (settings 3 and 4) than in rectangular worlds (settings 1 and 2), as can be seen in Table III. This is likely due to the fact that the range of visibility of each feature spans a symmetric field of view from a direction more or less perpendicular to the side (of the world or obstacle) where the feature is located. With this in mind, for several features to be visible in a pose, they have to be located on sides that are perpendicularly facing the pose (i.e., sides such that the pose location point perpendicularly projects inside the line segment defining them). In the case of rectangular worlds, this restricts the features visible in the pose to more likely come from at most four sides. In irregular worlds, there are likely more than just four sides facing each pose, and hence a larger number of features is visible from it.

TABLE III

AVERAGE NUMBER OF FEATURES VISIBLE FROM A POSE

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

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 feature visibility data collected in a 2m by 2m space sampled at 20 cm intervals, with a total of 46 visible features.[6] All images were taken with the camera in a fixed orientation (looking forward), and features were extracted using the Kanada-Lucas-Tomasi (KLT) operator [11]. The parameters used in the decomposition were $k = 4$, $\rho = 0$, $\sigma = 3$ The four regions of the decompositions can be seen in figure 8. The larger gray area present in each one of the images of the regions, corresponds to the set of k-coverable poses. As can be seen from the figure, the union of the four regions covers almost completely the k-coverable area of the world.

In our simulations, we obtained fairly big regions, as seen in Table II. Each pose corresponds to a sampled area of 0.25m^2 (50cm by 50cm), so the averages achieved by the best algorithm correspond to region areas of 20m^2 for features of Type 1, and 65m^2 for features of Type 2. These results were achieved with only a few features visible per pose, as shown in Table III, where the average number of features visible per pose was on the order of a hundred. 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.

[6]We used a small world and features with reduced visibility so that such a world can be interestingly divided into several regions to exemplify our method. For general applications in large environments an alternate class of features with visibility larger than this should be chosen.

VIII. CONCLUSIONS

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 most promising algorithm was then tested on real-world data with encouraging results.

ACKNOWLEDGMENTS

The authors wish to thank the many individuals (since 1994) who have contributed to earlier versions of this work, including David Wilkes, Ehud Rivlin, Ronen Basri, Lenore Cowan, James Ezick, David Rosenberg, Caroline Klivans, Sarmad Abbasi, and Sadaf Baqir. The authors would also like to gratefully acknowledge the support of NSERC, PREA, CITO, MD Robotics, and ONR.

REFERENCES

[1] D. Lowe, "Object Recognition from Local Scale-Invariant Features", *ICCV 1999*, Greece, 1999, pp. 1150-1157.

[2] S. Se, D. Lowe and J. Little, "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks", *Inter. J. of Robotics Research, Sage Publications,* Vol. 21, No. 8, 2002, pp. 735-758.

[3] G. Dudek and D. Jugessur, "Robust Place Recognition using Local Appearance based Methods", *ICRA 2000,* San Francisco, CA, USA, 2000, pp. 1030-1035.

[4] R. Sim and G. Dudek, "Learning and Evaluating Visual Features for Pose Estimation", *ICCV 1999,* Greece, 1999, pp. 1217-1222.

[5] G. Carneiro and A. D. Jepson, "Multi-scale Phase-based Local Features", *CVPR 2003,* USA, 2003, pp. 736-743.

[6] Ronen Basri and Ehud Rivlin, "Localization and Homing Using Combinations of Model Views", *AI*, Vol. 78 No. 1-2, 1995, pp. 327-354.

[7] D. Wilkes, S. Dickinson, E. Rivlin, and R. Basri, "Navigation Based on a Network of 2D Images", *ICPR-A 94,* pp. 373-378.

[8] P. L. Sala, R. Sim, A. Shokoufandeh, S. J. Dickinson, "Technical Report: Optimal Landmark Acquisition for Vision-Based Navigation", http://www.cs.toronto.edu/∼psala/Papers/MORDP-TR.pdf

[9] K. T. Sutherland and W. B. Thompson, "Inexact Navigation", *ICRA 1993*, USA, pp. 1–7.

[10] R. Karp, "Reducibility among combinatorial problems", *Complexity of Computer Computations, Plenum Press,* New York 1972, pp. 85-103.

[11] S. Birchfield, "KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker", http://vision.stanford.edu/∼birch/klt

[12] Z. Zhang, "Determining the Epipolar Geometry and its Uncertainty: A Review", *IJCV 98,* 27(2), pp. 161-198.

[13] T. Grossman and A. Wool, "Computational Experience with Approximation Algorithms for the Set Covering Problem", *Eur. J. Oper. Res.*, vol. 101, No 1, 1997, pp. 81-92.