Feature-Based Localization using Scannable Visibility Sectors

Jinsuck Kim Roger A. Pearce Nancy M. Amato
Department of Computer Science
Texas A&M University
College Station, TX 77843
{jinsuckk,rap2317,amato}@cs.tamu.edu

Abstract

This paper presents methods for navigating and localizing mobile robots in a known indoor environment. We introduce a restricted visibility concept called a scannable sector *that can aid many existing navigation and localization algorithms. The scannable sectors are based on the physical characteristics of the environment and the limitations of the localization sensors used. We describe a complete navigation system that includes a scannable sector based localizer, sonar sensors, and a probabilistic roadmap path planner. Simulation and hardware results using a real robot with sonar sensors show the potential of our approach.*

I. Introduction

As mobile robots travel and interact with their environment, physical position error may be introduced by wheel slippage, shifting payloads, inaccurate turns, etc. The accumulated error results in an increasing difference between the robot's presumed and actual position. This position error must be reduced by a *localizer* to avoid potentially costly complications.

Every localizer has an implicit/explicit algorithm which matches known landmarks/features to sensor data. We assume that the features are not uniquely identified by sensors, for example, using different colors for all features to be sensed by cameras. To identify a feature geometrically, a common approach is to preprocess the environment and store features in a dedicated data structure for localization. A popular data structure for this is an evidence grid [15]. Recent developments enable the robot to use an evidence grid only around features and to use a model description language (MDL) for inter–landmark navigation [5].

Many localization algorithms are based on the sensing ability, or visibility, of the sensors used. Visibility sectors which subdivide the free space using visibility polygons were introduced in [4] and were augmented in [10]. In [13], landmark placement in a subdivided configuration space is optimized using a formulation which considers the *localization space*, a set of robot poses $< x, y, \theta >$, and the number of visible features. For localization using vision, *visibility regions*, computed during preprocessing, allow the robot to localize using an image retrieval algorithm based on invariant features [17].

In this paper we introduce *scannable sectors* which are similar in motivation and use to the concepts of evidence grids [5], localization space [13], and visibility regions [17]. Our definition of visibility is more general so that more of the free space is registered and used for localization. Scannable sectors extend the concept of visibility to account for sensor limitations. Finally, we can localize with range sensors using only one feature as compared to two in other approaches [13].

II. Scannable Sectors

The environment is subdivided into sectors to aid the localization algorithm. Abstract definitions of terms used are given below. Their actual use will be discussed later in a more applied manner.

A. Definitions

The following concept is useful to restrict the localization algorithm to a subset of the environment.

Definition: An *uncertainty region* is the set of all possible positions of the robot. Statistical analysis has shown that it is often sufficient to use an ellipse to represent the boundary of the uncertainty region. We assume that the region is conservative so that we are guaranteed that the robot's actual position is contained in the uncertainty region.

To facilitate defining sectors, we categorize localization in two ways.

Definition: *Relaxed localization* is a method for reducing the robot's position and orientation uncertainty using sensor data.

Relaxed localization includes all ways of reducing the uncertainty of the robot's position and orientation. In the case of Kalman filters, every process that reduces the magnitude of the error covariance matrix elements is relaxed localization.

Definition: *Precise localization* is a subset of relaxed localization where the robot's uncertainty is reduced to zero (i.e., its location is known precisely).

Relaxed localization is important, for example, for range sensors because using only one range measurement will provide only one dimensional information which would not enable precise localization. In this paper, we present scannable sector data structures and algorithms that enable the robot to localize as precisely as possible using range sensors.

Definition: A *feature* is a part of an object in the environment that can be sensed for relaxed localization.

Our definition of feature and relaxed localization is not limited to range sensors. For example, a feature for a video camera can be a part of the wall with a different color or brightness, or a feature for tactile sensors could be a part of wall with a different texture.

In our experiments, the features of the environment are represented by polygonal objects that can be sensed by the robot's sonar sensor. Commonly used features are walls, desks, bookshelves and other small static objects that lie on the floor.

Definition: The *visibility number* is the number of features visible from a particular position of the robot where the robot (or sensors), may rotate 360^o using a rotating head or by rotating the body of the robot.

Definition: A *visibility polygon* is a maximal planar region of the environment in which a feature is visible from all points in the sector.

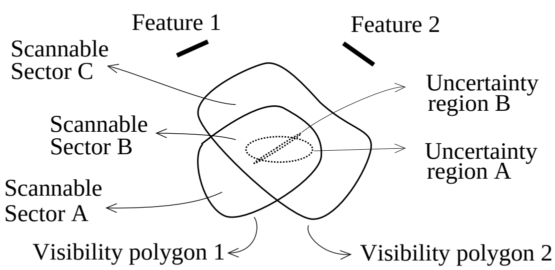

Fig. 1. Features and scannable sectors.

Figure 1 shows two abstract features and two visibility polygons. Polygon 1 represents the region where the robot can scan feature 1 and perform relaxed localization. Using range sensors, each feature gives one dimensional information which is the distance to the feature. Localizing to one feature shrinks the ellipse into a thin ellipse whose major axis is perpendicular to the line of sight to the feature. This is shown in the figure where the region A is reduced to B.

Definition: A *scannable sector* is a maximal planar region of the environment in which the robot can sense the same features from all points in the sector.

If the robot is inside scannable sector B which is the intersection of sector 1 and sector 2, then the resulting uncertainty will be the intersection of the thin ellipses shown in Figure 1 and another thin ellipse corresponding to feature 2 (not shown). This is a case where precise localization is possible.

B. Computing Scannable Sectors

In Figure 1, two visibility polygons yield three scannable sectors. Let s and p denote scannable sectors and visibility polygons, respectively. Computing s from p requires three Boolean polygon operations: $s_A = p_1 \setminus p_2, s_B = p_1 \cap p_2$, and $s_C = p_2 \setminus p_1$. Computing the scannable sectors s_A, s_B and s_C from visibility polygons p_1, and p_2 is described in the pseudo code in Figure 2. The scannable sectors S_A, S_B, S_C output are sets of polygons. We use a generalized polygon representation which allows empty, single or multiple components for the scannable sectors s_A, s_B and s_C. Examples are shown in Figure 3.

COMPUTE SCANNABLE SECTORS(p_1, p_2, S_A, S_B, S_C)
$s_A \leftarrow p_1, \quad s_B \leftarrow p_1 \cap p_2, \quad s_C \leftarrow p_2$
if $s_B = \emptyset$
  $S_A \leftarrow S_A \cup \{s_A\}$
  $S_C \leftarrow S_C \cup \{s_C\}$
else
  for each component $s_{Bi} \in s_B$
    $s_A \leftarrow s_A \setminus s_{Bi}$
    $s_C \leftarrow s_C \setminus s_{Bi}$
    $S_B \leftarrow S_B \cup \{s_{Bi}\}$
  $S_A \leftarrow S_A \cup s_A$ (for each component of s_A)
  $S_C \leftarrow S_C \cup s_C$ (for each component of s_C)

Fig. 2. Pseudo-code to compute scannable sectors from two visibility polygons.

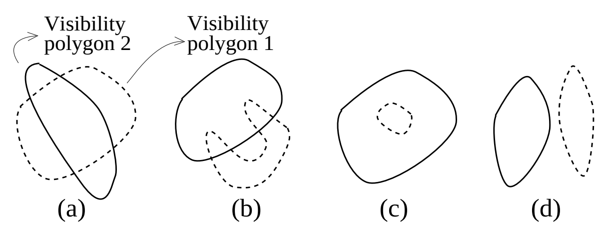

Fig. 3. Sectors represented by generalized polygons.

The algorithm sketched in Figure 4 constructs all the scannable sectors in the environment. It uses the COMPUTE SCANNABLE SECTORS subroutine shown in Figure 2. The input is a set of visibility polygons $P = \{p_1, p_2, \ldots, p_{n_p}\}$, and the output is a set of scannable sectors $S = \{s_1, s_2, \ldots, s_{n_s}\}$ where n_p and n_s are the number of visibility polygons and scannable sectors, respectively. In the pseudo code, t is the visibility polygon that will be partitioned into scannable sectors in each iteration.

III. SECTORS FOR SONAR SENSORS

The shape of the visibility polygons depends on the the physical limitations of the sensors. Sectors for sonar sensors are discussed in this section.

```
CONSTRUCT SCANNABLE SECTORS(P, S)
S = {p_1}
for i = 2 to n_p
    t ← p_i,   S_t ← ∅
    for each component s_i ∈ S
        COMPUTESCANNABLESECTORS(s_i, t, S_A, S_B, S_C)
        S_t ← S_t ∪ S_A  (for each component of S_A)
        S_t ← S_t ∪ S_B  (for each component of S_B)
        t ← S_C
    S ← S ∪ t   (for each component of t)
    S ← S ∪ S_t  (for each component of S_t)
```

Fig. 4. Pseudo-code to compute scannable sectors from visibility polygons.

A. Visibility polygons and Sectors

Example sonar sensor measurements using our hardware are shown in Figure 5(a). From a total of 60 scans around a fixed position, the correct range readings are obtained only for the walls to the right, to the bottom and the corner on the upper left. Other sonar beams do not echo back to the robot due to maximum range limit (Figure 5(b)) and incidence angle range limits (Figure 5(c)). Thus, only walls approximately perpendicular to the sensor beam and concave corners can be correctly measured.

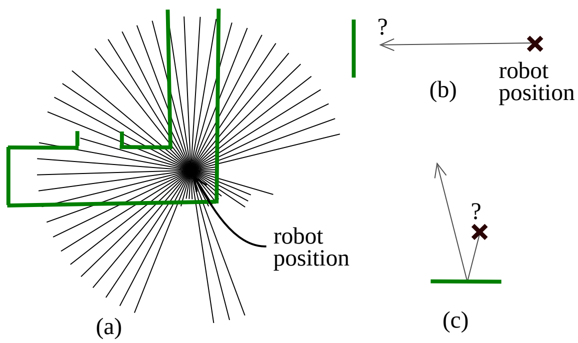

Fig. 5. Sonar sensor measurements.

The visibility polygon boundary represents the limitations of the sensors used. Sonar sensors have a minimum scanning range which leaves a gap between a visibility polygon and a feature. The maximum range of the sensor leaves a gap on the far side as well. The wall feature of Figure 6 shows a visibility polygon representing the minimum and maximum range limitations. Concave corner features of Figure 6 create a donut like (a portion of annulus) sector whose inner and outer radius are the minimum and maximum range of the sensors. In Figure 6, two of the corners create such sectors. For the donut like sector, the uncertainty ellipse is reduced to a crescent like shape after relaxed localization. The ellipse enclosing the crescent shape will be used as the result of a relaxed localization using a corner feature. The visibility numbers of the scannable sector are marked.

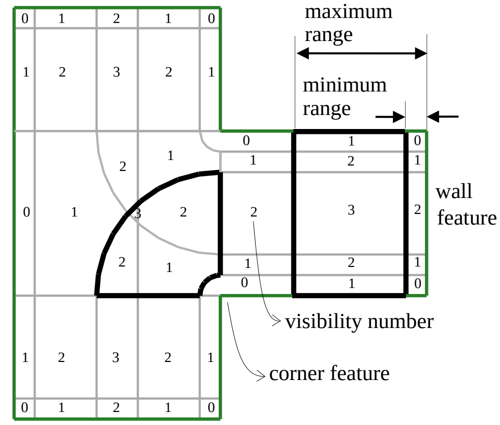

Fig. 6. Two features (wall and corner) with visibility polygons (thick–line regions), and scannable sectors with visibility numbers.

B. Localization using Sonar Sensors

Scannable sectors, which are data structures, can be used in various localization methods such as Monte Carlo localization and Kalman filtering.

Application to other methods. Particle filters [3], state estimators of a dynamic system, use sets of samples to represent the belief of the state. After computing the next state x_t from odometry and the previous state x_{t-1}, we need a perceptual model describing the likelihood of an observation y_t given that the robot is at location x_t. The benefit of scannable sectors, computed during a preprocessing phase, is that the probability can be obtained quickly.

Kalman filters are effective if Gaussian noise is prevalent compared to other uncertainty factors. With sensors having minimum/maximum range and incidence angle limitations, the uncertainty must be encoded as a nonlinear function in the measurement model. In discrete extended Kalman filters, if a valid linearization of the measurement model exists, then scannable sectors will help decide for what feature the function should be linearized.

Our proposed method. Our proposed method borrows the uncertainty ellipse from Kalman filters, but does not use statistically optimal estimation; we focus on the nonlinear behavior of sonar sensors illustrated in Figure 5. Pseudo code for localization using scannable sectors and sonar sensors is shown in Figure 7. A set of sectors S is obtained by Boolean operations on scannable sectors and the uncertainty ellipse. In Figure 8(b), three sectors are found from the situation in Figure 8(a).

The expected feature position boundary (FB in the pseudo code) allows the robot to select the features from the real scan data that matches with each of the preprocessed features. This is similar to the concept of RCD (regions of constant depth) in [6]. In Figure 8(c), the expected feature boundary FB is characterized by two measurements: angle A and length B. Angle A corresponds to the orientation error range of the robot, and length B represents the minimum and maximum distance to the feature from the scannable sector. In this case, the vertical length of s_2 is the same as B. In 8(d), because the robot is not in sector s_3, the expected feature boundary does not contain the wall and none of the features scanned in FB will be found. Real scanned features are shown in Section IV-C.

```
LOCALIZATION
S = Sectors intersecting uncertainty ellipse
F = Features of real scan
L = ∅ (a region of possible configurations)
for i=1 to SizeOf(S)
    FB = expected feature boundary of s_i
    if SizeOf(F ⊂ FB) ≥ 2
        cfg_2 = precise localization
        return cfg_2
    else
        cfg_1 = relaxed localization
        L = L ∪ cfg_1
return L
```

Fig. 7. Pseudo-code to localize using uncertainty ellipses, *scannable* sectors, and sonar sensors.

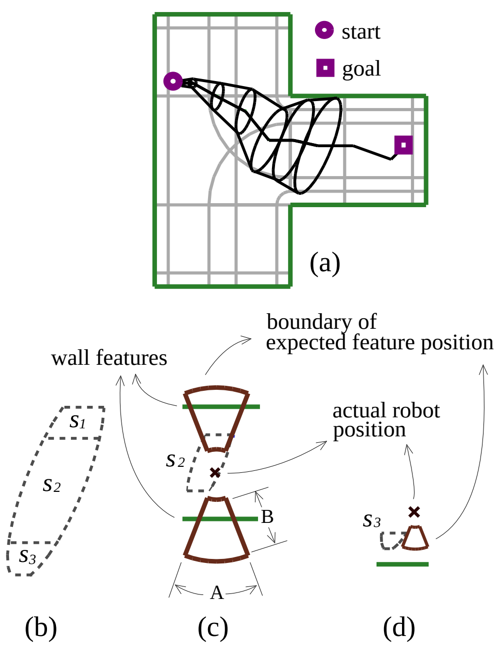

Fig. 8. Localization with scannable sectors.

Figure 9 shows the second iteration where the ellipse at the new start reflects the result of localization in Figure 8. The width of the ellipse corresponds to the horizontal length of s_2 in Figure 8. In the second iteration, since three walls provide two non–parallel distance measurements, precise localization is successful and its result is marked by $\times$ in Figure 9(b).

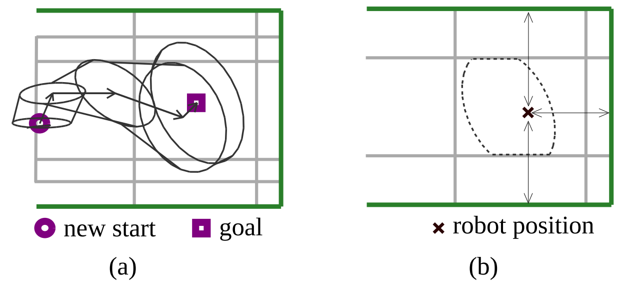

Fig. 9. Localization: second iteration.

Improvements from our previous methods. Compared to our previous work [8] where only regions with visibility number greater than one were used as features, we can now localize in a much larger part of the environment because we allow relaxed localization as well as precise localization.

IV. Experimental Results

For software development, we used C++ in Linux with the LEDA library [12] for geometric calculations and SAPHIRA [14] for hardware communication with an AmigoBot [2].

A. Global Navigation

A high level description of our system is described in the pseudo code in Figure 10. The overall strategy is the same as in our previous work [7], [11]. During the preprocessing phase, the environment is subdivided into sectors containing useful localization information and a roadmap is created. Initially, the robot is in a known starting configuration and the goal configuration is known. In the main loop a collision–free path is extracted from the roadmap. A subgoal is selected where the robot is expected to be able to move safely, localize, and then continue moving toward the goal.

```
NAVIGATOR(start, goal)
preprocess environment
while goal is not reached {
    extract path from start to goal
    determine localizable subgoal in the path
    drive robot to subgoal and stop
    scan and localize
    set start to current configuration
}
```

Fig. 10. Pseudo-code for our navigator.

Our framework is based on a roadmap–based path planner [9]. The *roadmap* is a graph representing feasible paths in the environment. Roadmap nodes correspond to collision–free configurations of the robot, and two nodes are connected by an edge if a path between the two corresponding configurations is collision–free. Our roadmap is a *probabilistic* roadmap which is well known for its ability to solve complicated high–dimensional problems [9].

The *scannable sector* data structure can enhance the roadmap generation step by providing information about localizable features in the environment. Inside regions with high *visibility*, roadmap nodes may be more concentrated and the local planner may increase the number of edge connections. These additions decrease the probability that localization fails.

B. Effects of Sensor Limitations

The plots in Figure 11 show the effect of varying the sensor maximum range on scannable sector attributes. For the simulations, the environment shown in Figure 12 with total area of 24.2 square meters was used.

Among five different sensor maximum ranges (50, 100, 150, 200, 250), 200 inches shows the largest total number of sectors. As sensor limitations become greater, i.e., the maximum range becomes shorter, Figure 11(b) shows that the majority of the sectors have smaller visibility numbers.

The largest visibility number in a sector is found when the sensor maximum range is longer than the size of environment (approximately 250 inches). The largest visibility number is 10, and there are two such sectors when the maximum range is 200 or 250 inches, and one sector when the maximum range is 150 inches.

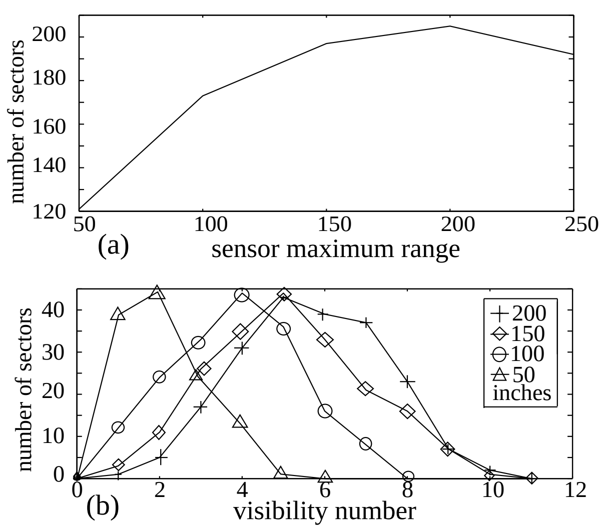

Fig. 11. Scannable sectors with different sensor maximum ranges.

C. Hardware Experiments

Our hardware experiments were conducted in a portion of our lab shown in Figure 12(a). The size of the environment is 256 by 269 inches. For this experiment, the minimum and maximum scan range was set to 5 and 100 inches, respectively. This range is large enough to cover all the nodes generated by MAPRM so that all of them have visibility number two or higher. The preprocessed scannable sectors, the robot's start and goal, and roadmap are represented by Figure 12(a). After a path is extracted and the first subgoal is chosen, the robot travels toward the subgoal and stops. Figure 12(a) also depicts the robot's growing uncertainty and actual position at the subgoal. Stopped at the subgoal in Figure 12(b), the robot scans the environment and potential features are gathered. The features observed by the sonar scans are matched to features in the environment and the robots pose is updated, Figure 12(c). Continuing, a new path is computed from the robot's actual position to the goal and the process restarts, Figure 12(d).

In the experiment shown in Figure 12, the robot's position at the first subgoal was off by 6 inches to the west and 6 inches to the north. The localization step corrected this but the sonar scans also included error because of an unknown obstacle, a trash can, shown in Figure 12(a). After localization the robot's position continued to be off target by 4 inches to the east and 2 inches to the south.

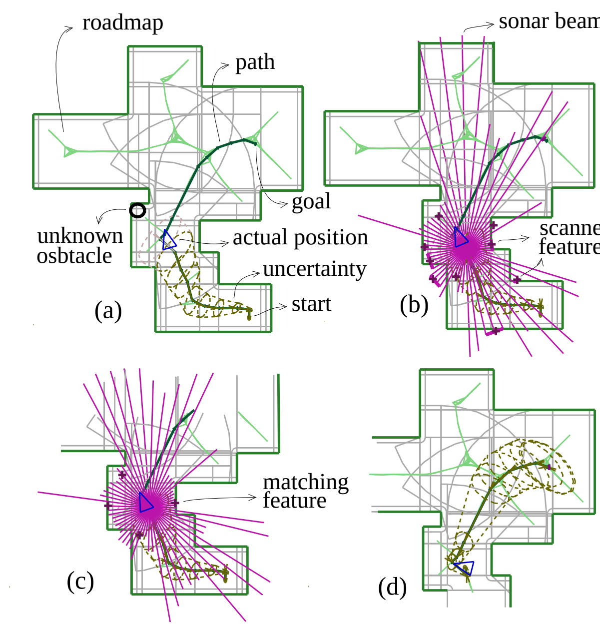

Fig. 12. Hardware example.

V. Conclusion

In this paper, we introduced a new restricted visibility concept, the scannable sector, that is general enough to be adapted to many other mobile robot algorithms. Scannable sectors are similar in concept to our previous work on visibility sectors [8], [11], but scannable sectors are more flexible, allowing for localization in a larger portion of the environment by matching fewer features. We described the design and implementation of methods for navigating a robot in an indoor environment using the scannable sector data structure. Navigation and localization can be aided by a new roadmap generation technique using scannable sectors. Our simulations and hardware experiments show the practicality and potential of our approach in the presence of sonar sensor limitations.

Our current research includes extending these methods for multiple robots. Multiple robots increase the complexity of navigation and localization, but the same principles behind scannable sectors apply.

VI. REFERENCES

[1] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 155–168, 1998.

[2] Amigobot. http://www.amigobot.com/.

[3] Dieter Fox. Kld-sampling: Adaptive particle filters and mobile robot localization. Technical Report UW-CSE-01-08-02, University of Washington, Seattle, WA, 2001.

[4] Leonidas J. Guibas, R. Motwani, and P. Raghavan. The robot localization problem. *SIAM J. Comput.*, 26(4):1120–1138, August 1997.

[5] D. Hristu-Varsakelis and S. Andersson. Directed graphs and motion description languages for robot navigation. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2689–2694, 2002.

[6] Leonard J. and Durrant-Whyte H. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3), 1991.

[7] J. Kim, N.M. Amato, and S. Lee. An integrated mobile robot path (re)planner and localizer for personal robots. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3789–3794, 2001.

[8] J. Kim, R.A. Pearce, and N.M. Amato. Robust geometric–based localization in indoor environments using sonar range sensors. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, 2002. To appear.

[9] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.

[10] S. Lee, N. M. Amato, and J. P. Fellers. Fast localization of mobile robots using visibility sectors. Technical report, Department of Computer Science, Texas A&M University, 1999. Appeared in ICRA 2000.

[11] S. Lee, N. M. Amato, and J. P. Fellers. Localization based on visibility sectors using range sensors. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3505–3511, 2000.

[12] K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, New York, 1998.

[13] Cedric Pradalier and Sepanta Sekhavat. Localization space: a framework for localization and planning, for systems using a sensor/landmark module. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 708–713, 2002.

[14] Saphira. http://www.ai.sri.com/∼konolige/saphira/.

[15] A. C. Schultz and W. Adams. Continuous localization using evidence grids. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2833–2839, 1998.

[16] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1024–1031, 1999.

[17] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization for mobile robots using an image retrieval system based on iavariant features. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 359–365, 2002.