In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2000)*, pages 1428-1434

Mobile Robot Navigation Using Self-Similar Landmarks*

Amy J. Briggs Daniel Scharstein
Darius Braziunas Cristian Dima Peter Wall

Department of Mathematics and Computer Science
Middlebury College
Middlebury, VT 05753, USA
`[briggs,schar,braziuna,dima,pwall]@middlebury.edu`

Abstract

We propose a new system for vision-based mobile robot navigation in an unmodeled environment. Simple, unobtrusive artificial landmarks are used as navigation and localization aids. The landmark patterns are designed so that they can be reliably detected in real-time in images taken with the robot's camera over a wide range of viewing configurations. The code for the recognition algorithm is publicly available at `http://www.middlebury.edu/~schar/landmark/`.

1 Introduction

Vision-based mobile robot navigation requires landmarks — either artificial or extracted from the environment — that can be quickly and reliably detected. We propose a self-similar pattern specifically designed for the application of mobile robot navigation and describe how exploration and navigation can be accomplished using these patterns as landmarks.

1.1 Landmark-based navigation

Localization and navigation based on landmarks can be very fast and reliable if landmarks are well-designed for efficient detection and well-distributed in the environment. Our approach is to use simple, unobtrusive, and easily installed patterns on the walls. The patterns are designed to be quickly recognizable under a variety of viewing conditions, even when partially occluded or mounted at an angle.

*Support for this work was provided in part by the National Science Foundation under POWRE grant EIA-9806108, VT-EPSCoR grant OSR-9350540, by Middlebury College, and by a grant to Middlebury College from the Howard Hughes Medical Institute.

1.2 Related work

The approach taken in this paper and by a number of other research groups [1, 12, 6] is to use artificial landmarks that can be easily and unobtrusively added to the environment. Techniques for mobile robot navigation based on landmarks include those that are primarily reactive [3], those planned within a geometric environment map enhanced with perceptual landmarks [5, 7], and those based on a topological description of landmark locations without a global map [4, 8, 11].

Our landmarks are designed for a navigation system in which a topological map of landmark locations is constructed through exploration of the environment and then used for navigation without relying on a global geometric map. Sensors are used for landmark detection and local obstacle avoidance. In this paper we describe a new landmark pattern specifically designed for the task of navigation. Unlike existing systems, the landmarks can be reliably detected in real-time under a wide range of viewing conditions with few assumptions about camera or landmark orientation, lighting conditions, or distances to the landmarks.

2 A self-similar landmark pattern

The key idea proposed here is to use self-similar intensity patterns [10] as visual landmarks for navigation.

2.1 Self-similar functions

We say a function $f : \mathbb{R}^+ \to \mathbb{R}$ is *p-similar* for a scale factor p, $0 < p < 1$, if $f(x) = f(px)\ \forall x > 0$. The graph of a p-similar function is self-similar; that is, it is identical to itself scaled by p in the x-direction (see

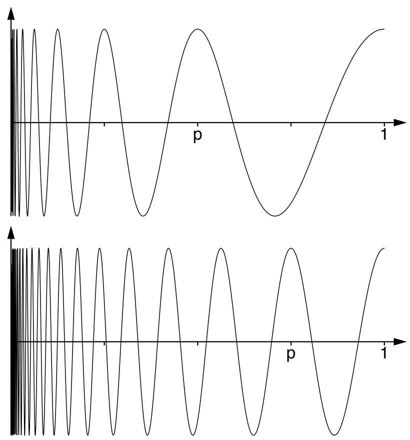

Figure 1: Examples of p-similar functions for $p = \frac{1}{2}$ (top) and $p = \frac{3}{4}$ (bottom).

Figure 1). Note that a p-similar function is also p^k-similar, for $k = 2, 3, \ldots$ Self-similar intensity patterns are attractive for recognition since the property of p-similarity is invariant to scale, and thus the distance of the pattern to the camera does not matter.

A p-similar pattern can be detected by comparing the observed intensity pattern to a version of itself scaled by p. We can accommodate patterns of limited spatial extent by restricting the comparison to a window of width w. Let $d_{p,w}(f)$ be the average absolute difference between the original and scaled functions over w:

$$d_{p,w}(f) = \frac{1}{w} \int_0^w |f(x) - f(px)|\, dx. \qquad (1)$$

Then f is *locally* p-similar over w if and only if $d_{p,w}(f) = 0$. A simple method for detecting a locally p-similar pattern in a one-dimensional intensity function $I(x)$ would then be to minimize the above measure over translations $I_t(x) = I(x+t)$:

$$t_{\text{match}} = \arg\min_t d_{p,w}(I_t).$$

The minimal value of 0 is achieved only if I is p-similar over w at translation t. Unfortunately, all constant functions are p-similar for any p. Thus $d_{p,w}(I_t)$ would also be minimal in regions of constant intensity. To exclude locally constant functions, we must detect patterns that are self-similar *only* for scale p (and $p^2, p^3, \ldots$), and not for other scales. This can be achieved by *maximizing* the *mismatch* at scale $p^{1/2}$ (and $p^{3/2}, p^{5/2}, \ldots$). Let us assume without loss of generality that the range of observable intensities is $[0, 1]$. A maximal mismatch for scale $\sqrt{p}$ is then given

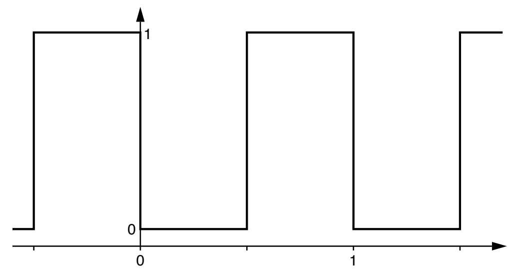

Figure 2: The square wave function $S(x)$.

if $|f(x) - f(\sqrt{p}\,x)| = 1$, or locally, if

$$d_{\sqrt{p},w}(f) = \frac{1}{w} \int_0^w |f(x) - f(\sqrt{p}\,x)|\, dx = 1. \qquad (2)$$

In this case we say that f is (locally) $\sqrt{p}$*-antisimilar.*

We can combine equations (1) and (2), and revise our method for detecting self-similar patterns that are only p-similar for a given scale p. We will maximize the match function

$$m(t) = d_{\sqrt{p},w}(I_t) - d_{p,w}(I_t) \qquad (3)$$

over all translations t. Note that the range of $m(t)$ is $[-1, 1]$, since the range of both terms on the right-hand side of (3) is $[0, 1]$. A locally constant function will yield $m(t) = 0$. Similarly, a random intensity pattern that is neither p-similar nor $\sqrt{p}$-similar will yield a response close to zero. A significant positive response is only expected for intensity patterns that are p-similar but not $\sqrt{p}$-similar.

2.2 An optimally recognizable pattern

Given the match measure m defined in (3), we now need to find an intensity function that yields the optimal response $m = 1$. That is, we seek a p-similar, $\sqrt{p}$-antisimilar function $s_p(x)$.

To derive such a function, let us consider the periodic "square-wave" function S depicted in Figure 2:

$$S(x) = \begin{cases} 0, & x - \lfloor x \rfloor < \frac{1}{2} \\ 1, & x - \lfloor x \rfloor \geq \frac{1}{2} \end{cases} = \lfloor 2(x - \lfloor x \rfloor) \rfloor, \qquad (4)$$

This function has the property that $S(x+1) = S(x)$ and $S(x+\frac{1}{2}) = 1 - S(x)$, i.e., it is a 1-periodic function that is similar under a *translation* of 1 and anti-similar under a translation of $\frac{1}{2}$. It is easy to show that we can transform S into a p-similar, $\sqrt{p}$-antisimilar function s_p by substituting $\log_p x$ for x:

$$s_p(x) = S(\log_p x) = \lfloor 2(\log_p x - \lfloor \log_p x \rfloor) \rfloor \qquad (5)$$

with the desired properties of $d_{p,w}(s_p) = 0$ and $d_{\sqrt{p},w}(s_p) = 1$ for any w (see Figure 3).

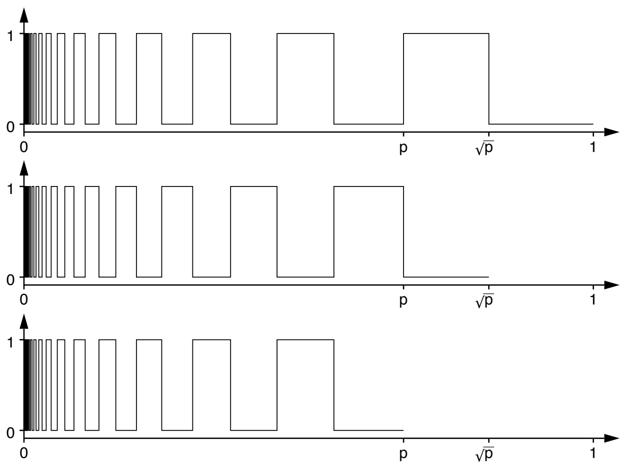

Figure 3: Self-similar square wave $s_p(x)$ for $p = \frac{2}{3}$ (top); maximal mismatch at scale $p^{1/2}$ (middle); and match at scale p (bottom).

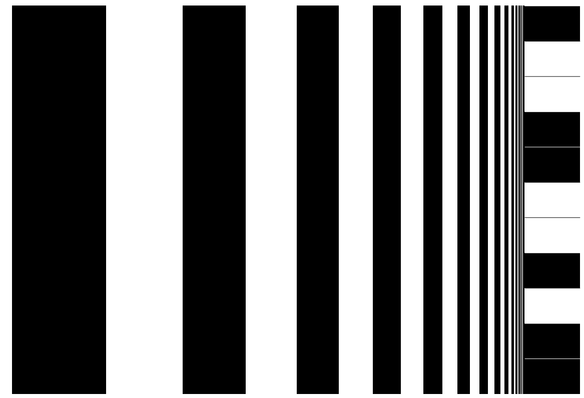

Figure 4: Our self-similar landmark pattern with barcode.

2.3 Two-dimensional landmarks

We now have all the components for landmark design and recognition. The key step for moving to two dimensions is to use a pattern that is p-similar in one direction (say, horizontally), and constant in the other direction. See Figure 4 for an illustration. If such a pattern is then sampled along any non-vertical line, the resulting intensity function is still p-similar because of the scale invariance of self-similarity. This allows us to detect two-dimensional p-similar patterns that have undergone an affine transformation by examining isolated scanlines.

Formally, let

$$L(x, y) = s_p(x) \tag{6}$$

be our two-dimensional landmark pattern, where $s_p(x)$ is the self-similar square wave function (equation (5)) for a fixed p. An affine transformation yields

$$A(L(x, y)) = L(ax+by+c, dx+ey+f) = s_p(ax+by+c).$$

Sampled at $y = y_0$, we get $s_p(ax + (by_0 + c)) = s_p(ax + t)$. Thus, the problem of finding an affine transformation of the two-dimensional pattern L has been reduced to finding a translation of the one-dimensional pattern s_p.

3 The landmark recognition algorithm

The idea underlying the recognition algorithm is to find locations (x_m, y_m) in the image at which a scanline is locally p-similar and $\sqrt{p}$-antisimilar. To do this, we adopt the matching function m from equation (3) for scanlines in an image $I(x, y)$:

$$\begin{aligned} m_y(x) = \quad & \frac{1}{w} \int_0^w \left| I(x+\xi, y) - I(x+\sqrt{p}\,\xi, y) \right| d\xi \\ & - \frac{1}{w} \int_0^w \left| I(x+\xi, y) - I(x+p\xi, y) \right| d\xi. \end{aligned} \tag{7}$$

The value of $m_y(x)$ depends on the intensities along the line (x, y) to $(x + w, y)$, and is constrained to the interval $[-1, 1]$. If the pattern s_p is present at (x, y), then $m_y(x) = 1$. It is easy to see that the pattern cs_p with reduced contrast $c < 1$ (i.e., difference between maximal and minimal intensities) will only yield a response $m_y(x) = c$. Other (non p-similar) intensity patterns will yield responses close to or below zero.

An algorithm for finding affine transformations of a landmark with intensities $L(x, y) = s_p(x)$ for a known p in an image can be formulated as follows:

for every k-th scanline y
    for all x
        compute $m_y(x)$
    mark all strong local maxima of m_y as matches.

This simple algorithm requires only $O(nw/k)$ operations for an n-pixel image. The computation of $m_y(x)$ can be adapted to discrete images by replacing the integrals in equation (7) by summations, and determining inter-pixel intensities using linear interpolation. The two parameters, k, the spacing of scanlines to search, and w, the window size, only depend on the smallest expected size of the landmark pattern and can be fixed. Typical values for an image of size 640×480 are $k = 6$ and $w = 45$.

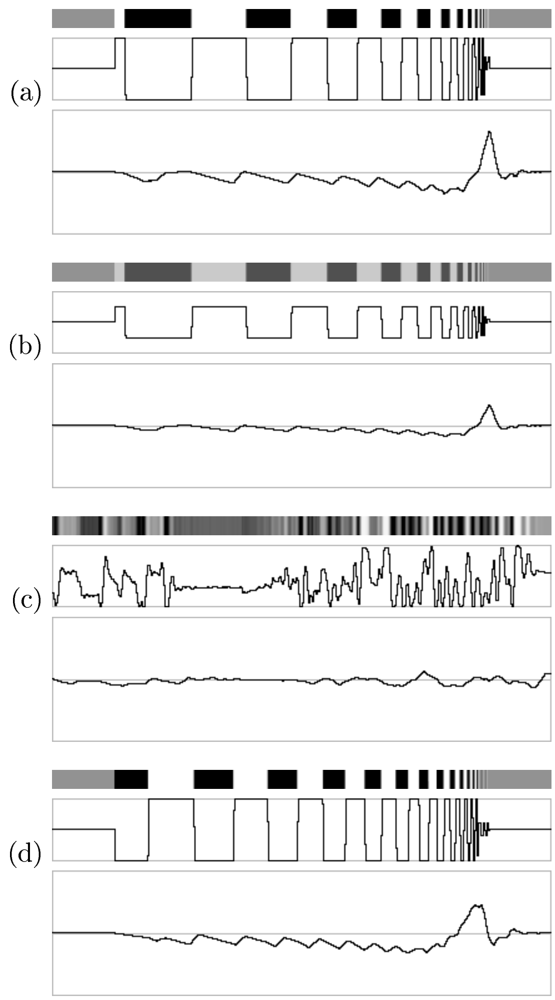

Figure 5: Various intensity patterns $I(x)$ and match functions $m_y(x)$ for $p = \frac{2}{3}$ and $w = 50$.

3.1 Finding matches reliably

The interesting question is: what constitutes a "strong" local maximum? A simple answer would be to require that a local maximum be greater than a fixed threshold $c_{\min}$ (corresponding to a minimum contrast). More information, however, can be gained from observing the shape of $m_y(x)$ in the vicinity of a local maximum. Figure 5 shows several intensity profiles of scanlines that were synthesized from continuous functions using 10-fold oversampling. Below each intensity plot is a plot of $m_y(x)$ for $p = \frac{2}{3}$ and $w = 50$. The length of each scanline is 400. The top two patterns are locally $\frac{2}{3}$-similar square waves s_p with full and half contrast, respectively. Both patterns result in clear peaks at the correct location $x_m = 350$; however, the value of the maximum $m_y(x_m)$ is less than expected. The observed values are 0.66 and 0.33, while the expected values are 1 and $\frac{1}{2}$. The differences are due to sampling and interpolation, in particular at locations of discontinuities or strong change. The bottom two patterns, on the other hand, are *not* $\frac{2}{3}$-similar: (c) is a "random" intensity pattern taken from a real image, and (d) is a $\frac{3}{4}$-similar square wave. The match function for the random pattern (c) has no strong peaks, but there is a distinctive local maximum in the match function for (d). It is, however, much more rounded and not as "sharp" as the top two peaks.

An experimental study analyzing the shape of $m_y(x)$ for a wide range of parameters has revealed a simple but effective test for "sharpness": check that a peak at half its height is no wider than a fixed threshold, which only depends on the window size w. That is, given a local maximum $v = m_y(x_m)$ greater than a threshold $c_{\min}$, test whether $m_y(x_m \pm \delta_x) < v/2$, where δ_x is approximately $w/10$.

3.2 Grouping matches

The actual landmark patterns can be detected in an image by grouping individual matches found on consecutive scanlines. To be able to distinguish different landmarks, we add a simple binary barcode to the right side of the self-similar pattern. (See Figure 4 for illustration.) The patterns can then be recognized and identified as follows: First, the position and orientation of all landmark patterns in the image is determined by fitting a straight line to each set of three or more individual matches detected on consecutive scanlines. The exact vertical extent of each pattern is then estimated from the intensity distribution to the left of this line. Once the locations of the patterns are known, their barcodes can be decoded easily. Figure 6 shows a Pioneer 2 mobile robot equipped with a camera, and two sample pictures taken by the robot. All landmarks are detected correctly.

3.3 Achieving real-time performance

The recognition algorithm as described above is fairly fast: typical running times for a straightforward implementation on a 450 MHz Pentium II machine are 0.33 seconds for 640×480 images and 0.08 seconds for 320×240 images (using parameters $w = 45$ and $k = 6$).

A dramatic speed-up can be achieved by further restricting the set of pixels for which $m_y(x)$ is computed. Recall that we are already considering only every k-th scanline. We can start by looking at only every $2k$-th scanline, and, only if a match is found, look at its

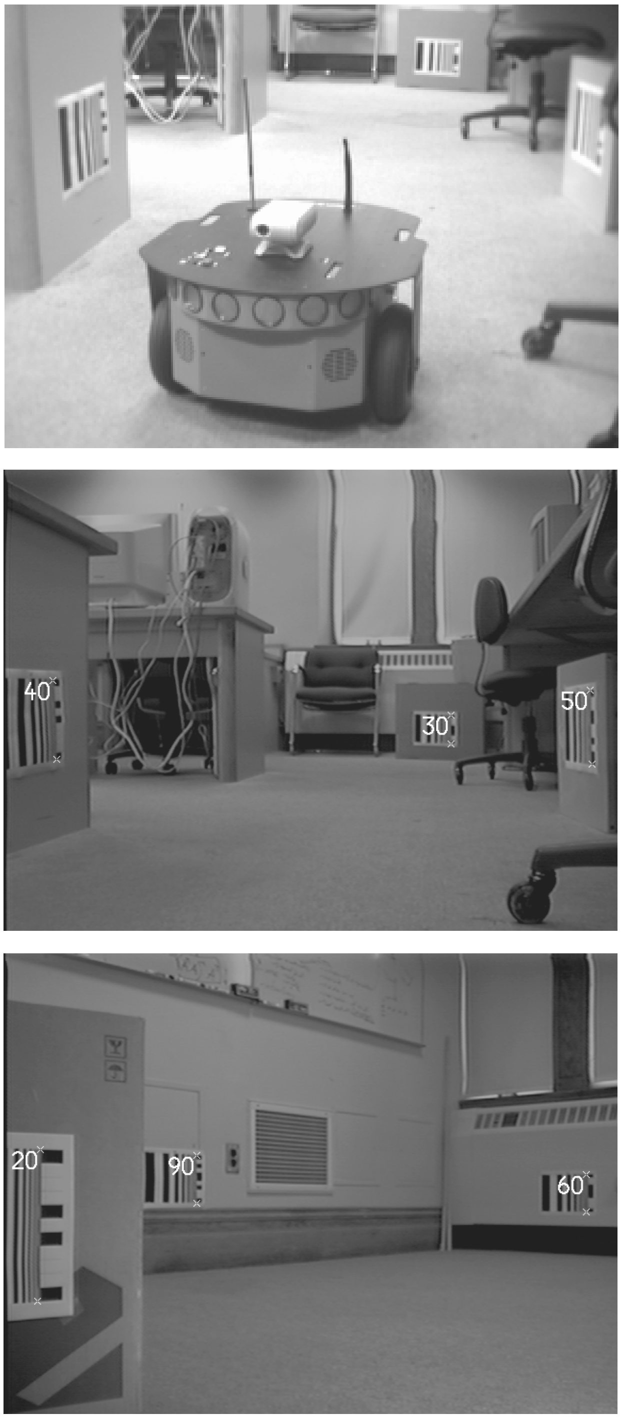

Figure 6: The mobile robot, and two images taken with its camera. All landmarks have been found and correctly identified by their barcodes.

neighboring scanlines as well. Since isolated matches are discarded anyway in the grouping process, no landmark will be missed. This technique can yield a speed-up of up to 2 if few or no landmark patterns are present in the image. Another opportunity for restricting the search is to make use of the fact that peaks corresponding to matches have a certain minimum width at a given height h (e.g., $h = c_{\min}/4$). Given a lower bound l for this width, it is possible to scan for peaks by looking only at every l-th pixel. A conservative bound for l is typically given by δ_x, i.e., half the allowable peak width used in the test for sharpness discussed in Section 3.1. Typical values for this number are 4 or 5. We have found, however, that even with much higher values for l (e.g., 10), only very few peaks are overlooked (typically those resulting from patterns with low contrast), so further speedup can be achieved with minimal impact on robustness.

These modifications result in new running times of 0.027 seconds for 640×480 images and 0.012 seconds for 320×240 images, corresponding to frame rates of 36 and 81 frames per second, respectively.

4 Exploration and navigation

Our landmark-based navigation system operates in two modes: exploration and navigation. In the exploration mode, the robot explores the environment to learn the relative positions of landmarks and builds a graph representing the connectivity between landmarks. This graph is used during navigation to plan reliable and efficient routes between landmarks.

4.1 Explore mode

During exploration, a directed graph is built denoting the connectivity between landmarks. An edge from landmark A to landmark B denotes that landmark B is visible from landmark A and that therefore the path from A to B can be traveled by moving toward B using visual servoing. For each landmark A, we keep a list of all other landmarks that can be seen from the robot when it is placed in front of A at a fixed distance from A. As is discussed in Section 3.2, the landmarks all have barcodes that enable them to be distinguished from one another. Each landmark in the visibility list for A is annotated with an ID, an estimated distance from A, and an estimated angle from A in a global coordinate system. A landmark's distance can be calculated from its size in an image and a landmark's relative angle can be determined from

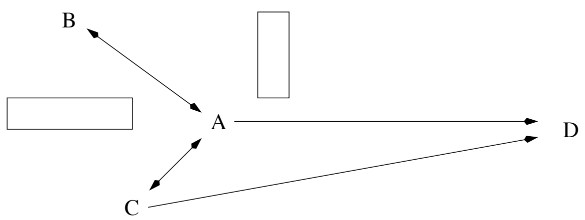

Figure 7: Example landmark visibility graph.

its horizontal location in an image. Angles are measured in a global coordinate system in which the robot assumes it heads in direction 0 when turned on and then uses previously obtained landmark visibility information to recalibrate its direction. Whenever the robot translates or rotates it updates its estimated position in the global coordinate system using odometry. If an edge of the landmark visibility graph is actually traversed during exploration, then the edge is further annotated with the measured distance.

The exploration algorithm works as follows. When at a landmark, scan for all visible landmarks. As a landmark is detected during the scan, estimate the angle and distance to it, and update the landmark graph with this information. After the scan step, visit the closest unexplored landmark, where distance is a weighted sum of the estimated edge lengths along the path and the time spent at each landmark to rotate and scan for the next landmark to be visited. In this way the landmark graph is built in a depth-first manner. Because the closest unexplored landmark may not be visible from the current landmark, a shortest path is planned along edges of the landmark graph. For example, in Figure 7, after starting at A and visiting C, the closest unvisited landmark in the graph is B. To reach B, the robot temporarily switches into navigation mode with B as the goal.

4.2 Navigation mode

In navigation mode the robot uses the state information currently in the landmark graph to plan paths between landmarks. When commanded to enter navigation mode, it moves to the nearest visible landmark so that it can construct a plan based on the visibility information from a landmark. A shortest path is planned using edges from the landmark graph, annotated with either an estimated or a traveled distance. As in the explore mode, when an edge is traversed the distance information for the edge is updated.

Due to a number of changing environmental conditions, such as lighting, moving obstacles, and changes in viewpoint, the visibility information between landmarks may differ over time and over runs of the system. It is likely, for example, that at some point during navigation a landmark in a planned path is no longer visible. A straightforward approach for countering these difficulties is to use recent visibility information to plan more reliable paths: Each time the robot is at a landmark along the planned path, it rotates to the direction in which it expects to see the next landmark. If that landmark is not visible, the landmark graph is updated to reflect this change in visibility. A new path is then planned using the current graph. If at some point during navigation the robot is at a landmark from which the destination landmark is no longer reachable, the robot uses dead reckoning to navigate to the closest landmark from which the destination is reachable.

4.3 A probabilistic model of visibility

While the strategy outlined above is a workable approach to dealing with the uncertainties associated with landmark detection, a more robust algorithm results from applying a probabilistic model of landmark detection, which allows optimal path planning based on the notion of *expected* shortest paths [2]. To do this, we annotate each edge $a \to b$ in the visibility graph with a probability p_{ab} that measures the likelihood that landmark b can be detected from landmark a. An estimate $\hat{p}_{ab}$ for this probability can be derived from the history of observations made by the robot at landmark a; in the simplest case (assuming independent observations) by dividing the number of times landmark b was visible by the total number of observations made at a. In practice, a more sophisticated estimate that takes into consideration the times at which the observations were made will be superior.

Independent of how estimates for the probabilities are derived, the path planning problem can now be phrased as follows: Given a directed graph in which each edge $a \to b$ has an associated cost l_{ab} and probability p_{ab}, and given a specified goal node g, compute a path with minimal expected length from the current node to g. This involves computing a quantity E_{xg} for each node x reflecting the expected length of the shortest path from x to g. Note that we are assuming that every time a landmark x is reached, the robot can see a subset of the landmarks corresponding to the outgoing edges from x. The robot can then choose to go to one of the landmarks currently visible, from which a new (random) subset of landmarks will be visible. The robot also has the option of staying at the current landmark and taking another picture, waiting for more landmarks to become visible. To prevent the robot from staying at one landmark indefinitely, we associate a non-zero cost with this option (e.g., the time it takes to acquire a new image). That is, each node x in the graph has a self-edge $x \to x$ with cost $l_{xx} > 0$ and probability $p_{xx} = 1$ (staying is always an option).

Given a graph with these properties, it is now possible to relate the unknown quantities E_{xg} (the expected lengths of the shortest paths from each node to the goal) in recursive equations. It can be shown that there is a unique solution to this system of equations if there is a path with non-zero probabilities from each node in the graph to goal g. It turns out that this problem is a special instance of a Markov decision process (in particular it is a non-discounted negative expected total-reward model) [9]. There are efficient ways of solving the equations, including variants of the traditional value iteration and policy iteration algorithms. Given a solution, the robot can at each landmark take the path with the expected shortest length given the current subset of visible landmarks. Since the probability estimates are continuously updated as the robot travels, this represents a robust way of dealing with both temporary failures of detection (e.g., temporary occlusion) and permanent changes to the environment (e.g., added or removed landmarks). For more details see [2].

5 Conclusion

In this paper we have presented a new system for landmark-based navigation in an unmodeled environment. Localization and navigation are done using simple and unobtrusive patterns mounted on the walls. Our landmarks consist of a self-similar intensity pattern coupled with a barcode for unique identification. The pattern was designed to enable very fast detection under a variety of viewing conditions. The system described in this paper represents the first real-time landmark-based navigation package that is reliable and robust with respect to landmark orientation, varying lighting conditions, and distances to the landmarks. The code for the landmark recognition algorithm is publicly available at `http://www.middlebury.edu/~schar/landmark/`.

Acknowledgements

We are grateful to Huan Ding, Cuong Nguyen, and Sorin Talamba for their insights and their assistance in implementing the algorithms and conducting the experiments reported here.

References

[1] C. Becker, J. Salas, K. Tokusei, and J.-C. Latombe. Reliable navigation using landmarks. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 401–406, June 1995.

[2] A. Briggs, D. Scharstein, and S. Abbott. Reliable mobile robot navigation from unreliable visual cues. In *Fourth International Workshop on Algorithmic Foundations of Robotics (WAFR 2000)*, Hanover, NH, March 2000.

[3] D. P. Huttenlocher, M. E. Leventon, and W. J. Rucklidge. Vision-guided navigation by comparing edge images. In *Algorithmic Foundations of Robotics (Proceedings WAFR'94)*, pages 85–96, 1995.

[4] A. Kosaka and J. Pan. Purdue experiments in model-based vision for hallway navigation. In *Proceedings of the Workshop on Vision for Robots in IROS'95, Pittsburgh, PA*, pages 87–96, 1995.

[5] A. Lazanas and J.-C. Latombe. Landmark-based robot navigation. *Algorithmica*, 13(5):472–501, May 1995.

[6] C. Lin and R. Tummala. Mobile robot navigation using artificial landmarks. *Journal of Robotic Systems*, 14(2):93–106, 1997.

[7] B. Nickerson, P. Jasiobedzki, D. Wilkes, M. Jenkin, E. Milios, J. Tsotsos, A. Jepson, and O. N. Bains. The ARK project: Autonomous mobile robots for known industrial environments. *Robotics and Autonomous Systems*, 25:83–104, 1998.

[8] C. Owen and U. Nehmzow. Landmark-based navigation for a mobile robot. In *Proceedings of Simulation of Adaptive Behaviour*. MIT Press, 1998.

[9] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, NY, 1994.

[10] D. Scharstein and A. Briggs. Fast recognition of self-similar landmarks. In *Workshop on Perception for Mobile Agents (in conjunction with IEEE CVPR'99)*, pages 74–81, June 1999.

[11] R. Sim and G. Dudek. Mobile robot localization from learned landmarks. In *Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems (IROS), Victoria, BC*, October 1998.

[12] C. J. Taylor and D. J. Kriegman. Vision-based motion planning and exploration algorithms for mobile robots. In *Algorithmic Foundations of Robotics (Proceedings WAFR'94)*, pages 69–83, 1995.