A Self-Calibrating, Vision-Based Navigation Assistant

Olivier Koch, Seth Teller

A Self-Calibrating, Vision-Based Navigation Assistant

Olivier Koch, Seth Teller
{koch, teller}@csail.mit.edu

Massachusetts Institute of Technology
Computer Science and Artifical Intelligence Laboratory
Cambridge MA, USA

Abstract. We describe a body-worn sensor suite, environment representation, set of algorithms, and graphical-aural interface designed to provide human-centered guidance to a person moving through a complex space. The central idea underlying our approach is to model the environment as a graph of visually distinctive places (graph nodes) connected by path segments (graph edges). During exploration, our algorithm processes multiple video-rate inputs to identify visual features and construct the "place graph" representation of the traversed space. The system then provides visual and/or spoken guidance in user-centered terms to lead the user along existing or newly-synthesized paths.
Our approach is novel in several respects: it requires no precise calibration of the cameras or multi-camera rig used; it generalizes to any number of cameras with any placement on the body; it learns the correlation between user motion and evolution of image features; it constructs the place graph automatically; and it provides only coarse (rather than precise metrical) guidance to the user.
We present an experimental study of our methods applied to walking routes through both indoor and outdoor environments, and show that the system provides accurate localization and effective navigation guidance.

1 Introduction

The ability to navigate effectively while performing long-duration walking excursions through spatially extended environments is fundamentally useful and difficult to achieve. Navigation systems have been proposed for environments in which GPS is available [1], providing the blind with some degree of situational awareness and guidance. However, when moving within an environment without good GPS reception (e.g., indoors, or in outdoor spaces with limited sky visibility), some other navigation sensor must be used. Because the world provides rich visual information, and because this information can be gathered with passive sensors (video cameras), we have chosen to investigate the use of machine vision techniques to assist with way-finding in complex GPS-denied environments.

We present a vision-based navigation system designed to assist both sighted and visually-impaired people in navigating through previously unexplored environments. We envision an operating scenario in which a user moves, beginning

from some designated start point, through an environment with complex internal structure such as passages, intersections, and turns. At any point along the user's path, our system provides guidance of the following form:

- *retracing*: guide from the starting point to any specified location;
- *homing*: guide from any specified (explored) location to the starting point;
- *routing*: guide from any specified (explored) location to any other.

An additional capability of our method is *loop excision*, that is, the ability to remove from an offered path any cycles that occurred during the user's initial exploration of the environment.

The user performing the exploration need not be the same as the user receiving the guidance, thus opening the possibility of sighted users generating a delayed guidance capability for sightless users.

Our method does not perform a metrical reconstruction of the environment. Instead, it automatically builds an annotated topological map of the environment by capturing orientation- and location-dependent scene appearance, and subsequently provides user-centered guidance within the captured region. The idea of using vision for non-metric, qualitative navigation is not new [2]. However, our method is novel in several respects. First, it is self-calibrating in that neither intrinsic nor precise extrinsic camera calibration are provided to or recovered by the system. Second, it relies on a new classifier-based approach for learning the correlation between the image motion field and the user's motion. Third, it does not attempt to produce metrically accurate guidance, but rather coarse guidance about progress and direction, much like one human would provide another. Finally, the method does not specify, or limit, the number of cameras used or their respective positions on the user's body. The method does not make any assumption about the appearance of the environment except that it contains visually distinctive and trackable features. We demonstrate the capability of the system on several real-world exploration and navigation sequences.

Section 3 presents our place graph representation. Section 4 gives a description of the user interface. Section 5 describes our algorithm for capturing a place graph. Section 6 shows how the place graph can be used to generate user-centered guidance. Section 7 presents applications specific to the visually impaired. Finally, section 8 presents validation and experimental results.

2 Related Work

A natural approach to vision-based navigation is to explicitly estimate metrical egomotion from the image motion field. Building on seminal work [3], recent methods demonstrate robust real-time egomotion estimation [4]. Some exploit the idea that egomotion estimation is more robust when using wide-field-of-view (FOV) sensors [5, 6]. Meanwhile, the problem of recovering both the camera motion and the structure of the environment, known as Simultaneous Localization and Mapping (SLAM) has attracted substantial attention [7–11]. However, SLAM algorithms often come at the cost of complex algorithms and well-

calibrated sensors. In addition, achieving efficient performance as the explored environment and excursion length grow remains a long-standing problem.

In order to cope with the complexity and scale of real environments, both topological [12, 13] and hierarchical [14] approaches have been proposed. In comparison, our method is hybrid in a sense that it builds a topological representation of the environment, but also computes an estimate of the user's orientation and progress in a local frame.

3 The Place Graph Representation

Our method builds a topological representation of the environment on-line during exploration. We define a "place graph" as an undirected graph where nodes represent places of interest, and edges represent physical paths between successive nodes (Figure 1). Given a function $\mathbf{f}$ that measures the variability in visual appearance of the scene, our method involves finding local maxima in the gradient of $\mathbf{f}$ which correspond to places of highest variability in the scene appearance. Nodes in the map correspond to local maxima of $\nabla\mathbf{f}$.

Our algorithm takes as input a sequence of images, and computes feature points for each image, associating the features with edges and nodes in the place graph. Sec 5 describes the *rotation guidance algorithm*, which takes as input the feature sets for two images and computes the optimal user rotation that would bring them in alignment. The algorithm is based on a voting scheme which provides a variance associated with its output. The variance represents a measure of confidence in the output and by extension, a measure of similarity between the two input images. When two images are visually similar, the votes will aggregate around a consensus and the variance will be low. On the other hand, when two images are dissimilar, the votes will randomly spread across $[0, 2\pi)$ and the variance will be high.

Let us define $\mathbf{f}(t_1, t_2)$ as the variance between a frame acquired at time t_1 and one acquired at time t_2. As the user moves about the environment, the algorithm computes $\mathbf{f}(t, t - \Delta t)$ at intervals (typically, Δt =one second) and searches for maxima of the gradient of $\mathbf{f}$. For every local maximum, a node is added to the place graph (we smooth the data to avoid detecting multiple maxima too close together in time or space). While the user navigates from a node to the next, the system records the set of image features at every frame.

4 User Interface

Our system includes four cameras, two pointing forward, two pointing backward, loosely mounted on the shoulder straps of a backpack (Figure 2). The overall field of view of the rig is 360° degrees horizontally and 90° vertically. The cameras are connected to a laptop, which captures visual data as the user explores the environment. Feature points are computed online.

The user interacts with the system using a Portable Device Assistant (PDA), headset and microphone. While the system builds a place graph automatically,

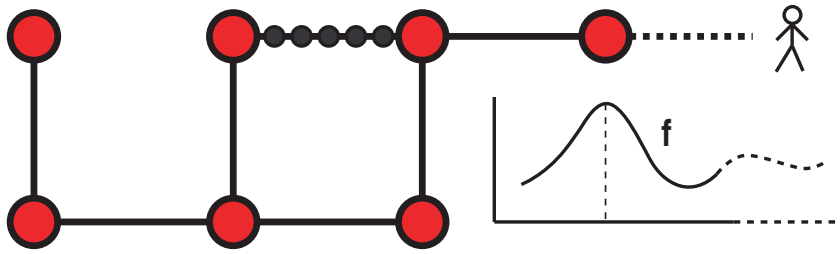

Fig. 1. We represent the environment as a place graph where nodes (red circles) are places of high variability in visual appearance of the scene and edges (black lines) are physical paths between nodes. The map is built automatically by the system during exploration. Features observed along edges (black dots) are stored in a database.

Fig. 2. Our operating scenario involves a human exploring a complex internal space (notional view at left; user path shown as dotted line). The user wears a multi-camera processing rig (center). At any point in time, the system provides audio information to the user about his/her global location in the place graph as well as user-centered directions toward the target destination.

the user may also wish to add a node in the map whenever s/he feels s/he is passing through a notable place. The user does so by pressing a button on the PDA. The system subsequently records the user's voice for a few seconds. The audio file is repeated to the user any time the node is revisited.

At any time during navigation, the user may request that the system replay the sound track attached to the previous and next node in the map. In addition, the system provides user-centered navigation guidance at any point by uttering one of the following phrases: "Rear Center", "Rear left", "Rear right", "Front Center", "Front left", "Front right", "Side left", "Side right". The system also periodically utters a progress estimate, expressed as a percentage, as the user traverses each place graph edge.

5 Capturing the Place Graph

The topological map relies on the existence of a function $\mathbf{f}$ that captures the location-dependent variation in visual appearance of the environment. We now describe an algorithm that takes as input two images and returns the optimal

rotation that would bring one in alignment with the other. We refer to this algorithm as the *rotation guidance algorithm*. First, it provides guidance to the user when reaching a place that has been visited before. Second, the variance in its output provides a measure of the visual similarity between the two input images.

The intuition underlying our method is that motion fields are hardly ever ambiguous [3], and that the ambiguity is further decreased when using multiple (wide-FOV) cameras. Ideally, one could compute optical flow at pixel resolution and reconstruct both the 3D structure of the environment and the 3D motion of the camera. However, as our method demonstrates, these steps are not necessary for successful navigation. Instead, learning a coarse mapping between the user egomotion and the motion fields is sufficient. The algorithms works with an arbitrary number of cameras, for which no intrinsic or extrinsic calibration is required.

We reduce the mapping function to a hash table (or classifier), in which each row corresponds to a region (bin) in one camera image and each column to another region in another camera image. The value in each cell of the table represents the user egomotion required to bring a point from the source region to the target region.

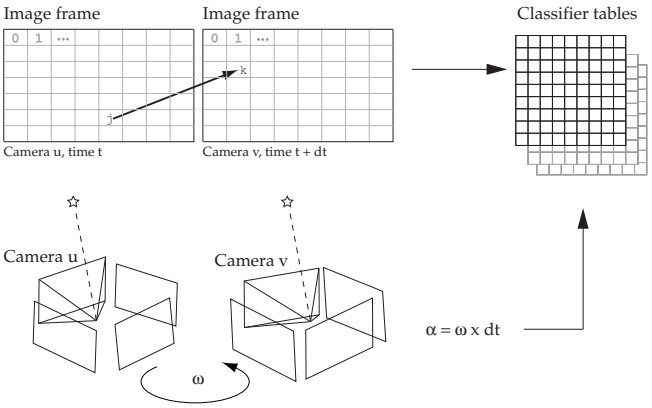

Fig. 3. Given a rotation speed ω, a feature tracked from camera u at time t to camera v at time $t+dt$ generates an entry $\alpha = \omega * dt$ in the matrix M^{uv}. There are $n_c * (n_c+1)/2$ matrices where n_c is the number of cameras on the rig.

During a training phase, the user rotates in place at approximately constant speed while the system captures data. Given the number of frames in the sequence and the number of turns executed during the sequence (typically, two), the system can estimate the rotation angle in the user's body frame between any two frames. Given a set of feature point matches between the two frames, the hash table is populated as shown in figure 3.

In practice, each camera image is subdivided into a grid of $k \times k$ bins (typically, $k = 10$). The hash table is represented by a set of matrices M^{uv} for each camera u and v ($u < v$). When multiple feature matches hit the same cell, the algorithm retains the average rotation value, in the following sense:

$$\bar{\alpha} = atan(\frac{\sum_{k=1}^{n} sin(\alpha_k)}{\sum_{k=1}^{n} cos(\alpha_k)}) \quad (1)$$

which corresponds to minimizing the square sum of the distances between each angle α_i represented as a point on the 2D unit circle and the point $(1, 0)$.

The algorithm proceeds as follows. First, it extracts a set of feature points $\mathcal{F}$ in the first image and a set $\mathcal{G}$ in the second image. Given a correspondence between a point $f_i \in \mathcal{F}$ on camera u and a point $f_j \in \mathcal{G}$ on camera v, it determines the rotation angle that would bring f_i into alignment with f_j. To do so, it computes r and s, the bins to which f_i and f_j belong respectively. Then, the desired rotation angle corresponds to M^{uv}_{rs}. This process is repeated for all feature matches and a voting algorithm returns an average rotation motion (Figure 4).

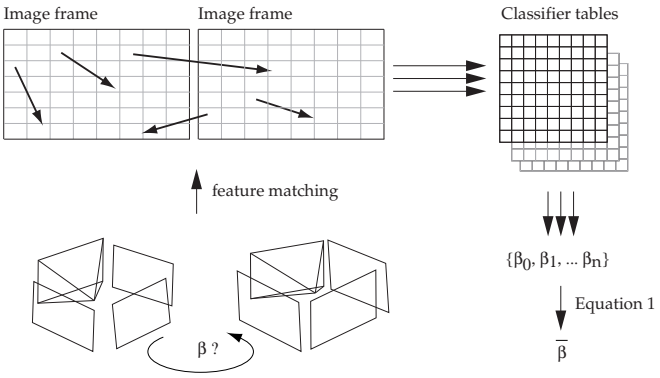

Fig. 4. Assuming pure rotation of the rig by an unknown angle β, each feature match is assigned a rotation angle by the classifier. The mean of all the angles (in the sense of equation 1) gives an estimation of β.

5.1 Method Resolution

Assuming a dense and homogeneous feature distribution and perfect feature matching, the accuracy of the algorithm does not depend on the grid resolution of the classifier. In order to prove this statement, we model the rig as four cameras, each covering a 90° FOV (Figure 5). We further assume that the classifier has been trained with a single bin per image. A feature match within the same

camera votes for a rotation of zero degree. A feature match between camera 0 and camera 1 votes for a rotation of 90°, and so forth.

Let us assume that the rig rotates by 90° (second tile on Figure 5). Each feature in camera 0 now appears in camera 1, and so forth. Hence, all feature matches vote for a rotation of 90°, yielding a zero error in the rotation estimate. Let us assume that the rig rotates by 45° (thid tile on Figure 5). Half of the features in camera 0 now appears in camera 1 (and votes for a rotation of 90°), while the other half stays in camera 0 (and votes for a rotation of zero). The average rotation estimate is therefore 45°, which again yields a zero error.

Finally, let us assume that the camera rotates by an arbitrary angle α (fourth tile on Figure 5). Let us us put β the ratio of α over 90°. A ratio of β feature matches will move to camera 1 (and vote for a rotation of 90°) while the remaining will vote for a rotation of zero degree. The rotation estimate is equal to $\beta \times \pi/2 + (1 - \beta) \times 0 = \alpha$, which again yields a zerro error. This reasoning extends straightforwardly to any number of bins.

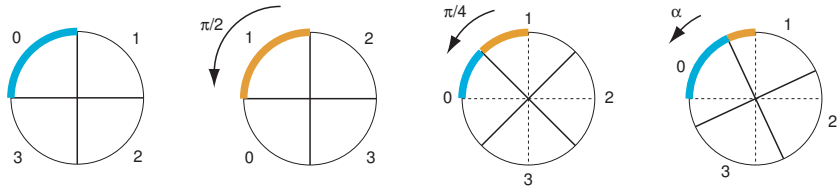

Fig. 5. Representing the rig as four cameras covering each 90° of field of view. Cameras are numbered from 0 to 3. *First tile*: reference position. Features on camera 0 are shown in blue. *Second tile*: after a rotation of $\pi/2$. All features have moved from camera 0 to camera 1 (in yellow). *Third tile*: after a rotation of $\pi/4$. Half of the features remain on camera 0 (blue) while the other half is now visible in camera 1 (yellow). *Fourth tile*: after an arbitrary rotation of α. The ratio of features which stay in camera 0 and move to camera 1 yield a perfect estimate of the rotation angle. In practice, subdividing images into bins helps support the non-density and non-homogeneity of feature distribution.

In practice however, the assumption of perfect feature matching and dense feature distribution does not hold. Assuming that feature mismatches spread evenly across cameras, a higher number of bins in the grid does not improve the method accuracy. However, a higher number of bins helps combat the uneven and non-dense nature of feature distribution. Indeed, if γ is the average angle subtended by any two features, the accuracy of a k-bin classifier has a minimum bound of γ/k. In addition, in the case of a one-bin classifier, if a large cluster of features located on the edge of a camera image moves in one block to the neighboring camera following a small rotation of the rig, the whole cluster will vote for a rotation of $\pi/2$ which will incur an over-estimation of the rotation of the rig. A higher grid resolution decreases the importance of this phenomenon.

On the other hand, the method may be subject to aliasing effects if the number of bins is too high. Given the finite number of feature matches during

training, a high-resolution grid may result in a sparse hash table from which queries return too few votes to obtain a significant result. As a conclusion, there exists an optimal grid size for the method driven on one hand by the sparsity of feature distribution in the image and on the other hand by the number of feature matches during training.

6 Generating Guidance from the Place Graph

To provide navigation assistance, the system tracks the motion of the user within the place graph graph. When the user is standing at a node, the system uses the *rotation guidance algorithm* to determine the relative orientation of the user at the node. This relative orientation is then converted into a user-centric navigation suggestion (Section 4). When the user is navigating along an edge, the system must determine the location of the user along that edge.

We formulate the problem as a general state estimation problem using a recursive estimator. The state is defined as a vector which length is the number of frames recorded during the first visit of the current edge. The value of each element in the vector is the probability of the user to be located at the corresponding frame. Let b_t be the belief state at time t after incorporating actual observations $\{o_1, \cdot, o_t\}$. We proceed in two steps:

Transition update

$$b'_{t+1}(s_{t+1}) = P(S_{t+1}|O_1 = o_1, \cdot, O_t = o_t) = \sum_{s_t} P(S_{t+1} = s_{t+1}|S_t = s_t) b_t(s_t) \tag{2}$$

Observation update, given a new observation o_{t+1}:

$$b_{t+1}(s_{t+1}) = P(S_{t+1} = s_{t+1}|O_1 = o_1, \cdot, O_{t+1} = o_{t+1}) \tag{3}$$

$$= \frac{P(O_{t+1} = o_{t+1}|S_{t+1} = s_{t+1}) b'_{t+1}(s_{t+1})}{\sum_{s_j} P(O_{t+1} = o_{t+1}|S_{t+1} = s_j) b'_{t+1}(s_j)} \tag{4}$$

The belief state is initially set to $[1, 0, \ldots, 0]$ as the user leaves the node. The probability distribution $P(S_{t+1} = s_j|S_t = s_i)$ used in the transition update is defined as a Normal distribution centered on s_i. The probability $P(O_{t+1} = o_{t+1}|S_{t+1} = s_{t+1})$ used in the observation update is defined as a quadratic function of the variance of the *rotation guidance algorithm* run between the features observed during the first visit of the edge (s_{t+1}) and the current features o_{t+1}.

The current location of the user along the edge corresponds to the maximum element in the belief state vector. The relative orientation of the user is determined by running the *rotation guidance algorithm* against the corresponding set of features in the database.

7 Application Scenarios

There are several applications of our system for the visually impaired. First, the *homing* feature allows the user to navigate within an unknown environment with the ability to be guided back to the starting point at any time. This is particularly useful for people who explore their neighborhood for the first time.

As a second application, a non-impaired person can use our device to first visit a new environment, such as a public building. Robust to small changes in relative camera position, the rig can then be used to guide a visually impaired user who is navigating the space for the first time.

As demonstrated in section 5, the system calibrates itself in a few minutes and does not require the assistance of a non-impaired person. We hope to leverage future advancements in camera technology and portable computers to decrease the size and weight of the system.

8 Validation and Experimental Results

8.1 System Details

The system includes four IEEE-1394 PointGrey Firefly MV cameras. Each camera holds a wide-angle 2.2 mm Tamron lens. The rig's overall field of view is 360° degrees horizontally and 90° vertically. The cameras are connected to a dual-core, 1.7-GHz laptop through a firewire hub, enabling video recording at full resolution ($4 \times 752 \times 480$) at 8 Hz. The laptop performs real-time Scale Invariant Feature Transform (SIFT [15]) feature computation on half-size images at 4 Hz. The user interacts with the system using a handheld PDA connected via Bluetooth. A $12\,V$ battery pack supports several hours of untethered operation.

8.2 Classifier Resolution

We evaluate the accuracy of our system by comparing the output of the *rotation guidance algorithm* on a set of video sequences for which ground-truth is known using an Inertial Measurement Unit (IMU). The sequences correspond to the user rotating in place in a different environment than the one used during training. Each sequence is approximately 200 frames long. The algorithm is run on every pair of frames, yielding to about $20,000$ data points per measurement.

Table 1 shows the standard deviation against ground-truth for various classifier resolutions. The accuracy of the method for a one-bin classifier is 13.2°, which is comparable to the average subtended angle γ between any two features in the image. Indeed, given an average number of $n_f = 70$ features per image and a horizontal field of view $h_{fov} = 120°$, we estimate γ as $\gamma = h_{fov}/\sqrt{n}_f = 14.3°$. As the number of bins increases, the standard deviation decreases slightly since the feature distribution in images is not dense and homogeneous. The two-bin classifier represents an outlier for which we have no clear explanation.

However, the standard deviation starts increasing beyond a threshold number of bins. Given $n_{f,t}$ the number of frames in the training sequence and $n_{k,t}$ the

average number of features per frame, we estimate the number of hits in the hash table as $n_{hits} = n_{f,t} \times (n_{f,t} - 1)/2 \times n_{k,t}$. For $n_{f,t} = 200$ and $n_{k,t} = 300$, $n_{hits} =$ 6 millions. On the other hand, given n_c the number of cameras and n_{bins} the grid resolution, the number of cells in the hash table n_{cells} is $n_{cells} = n_c^2 \times n_{bins}^4$. For $n_c = 4$ and $n_{bins} = 20$, $n_{cells} = 2.5$ millions. In conclusion, using a higher resolution than 20 bins is not reasonable, which shows in Figure 1.

Table 1. Standard deviation of the classifier (in degrees) and memory requirement with respect to the grid resolution.

Grid resolution	1	2	4	8	12	16	20	30
Standard deviation (°)	13.2	10.4	11.8	10.5	7.1	12.9	33.7	45.2
Memory usage	4.0 K	8.0 K	76 K	1.2 M	5.6 M	18 M	43 M	217 M

8.3 Datasets

We tested our system on two datasets. The INDOOR dataset was collected during a 15 minute-long exploration across several building of a campus, for a total distance of approximately 1.5 kilometers. The OUTDOOR dataset was collected during a 15 minute-long walk across a dense urban environments, for a total distance of approximately one kilometer. Both datasets were collected in unprepared environments involving dynamic scenes and lighting changes. We first present the topological maps generated by the system for the two datasets. Table 2 presents summary data.

Table 2. Datasets information

Name	Duration	Path length	Frame rate	# frames	# nodes
INDOOR	12 min	1.5km	4 Hz	6,000	120
OUTDOOR	12 min	1km	4 Hz	2,900	43

Generating the Place Graph Figure 8 shows the ∇f function described in Section 3 for the INDOOR dataset. Maxima in the function correspond to nodes in the map. Figure 6 shows the corresponding nodes manually overlaid on a 2D map. Figure 7 shows the topological map for the OUTDOOR dataset. Figure 9 shows sample images at node locations. As expected, nodes correspond to places of high variability in the appearance of the scene. They often correspond to places of interest in human-made environments.

Guidance From the Place Graph We performed a *retrace* scenario for each dataset. In each case, the user started from the first node in the map and requested navigation assistance from the system to the last node in the map. For

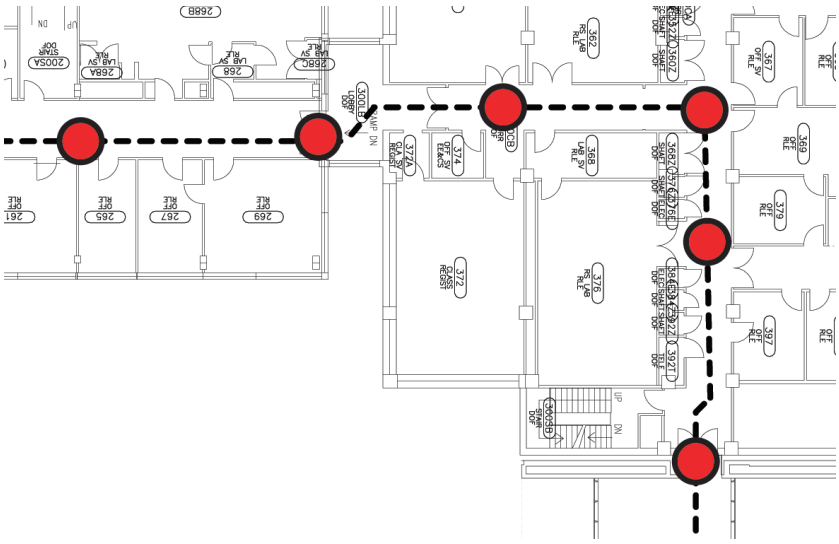

Fig. 6. Topological map for the INDOOR dataset. The map is built online and automatically by analyzing the variance in scene appearance. Nodes are represented as red circles. The notional path of the user is shown in dotted line.

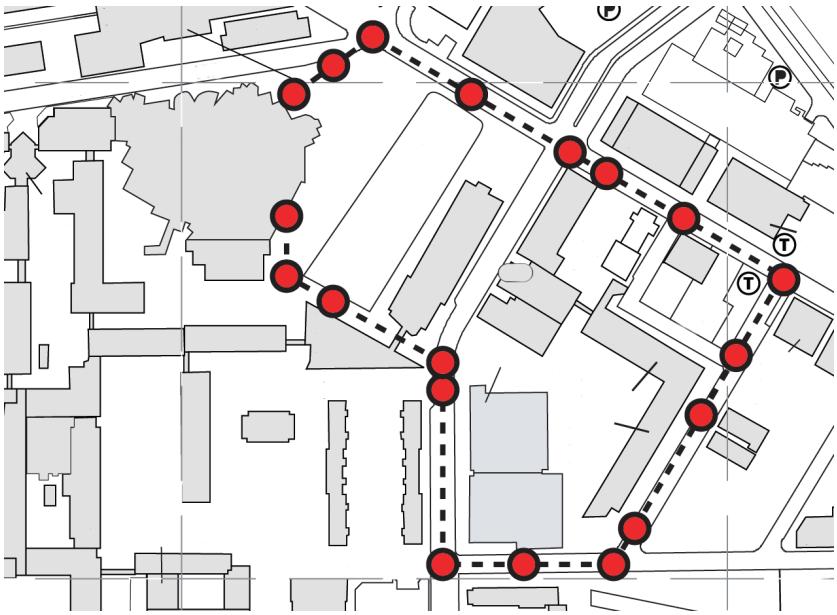

Fig. 7. Topological map for the OUTDOOR dataset. Nodes are represented as red circles. The notional path of the user is shown in dotted line.

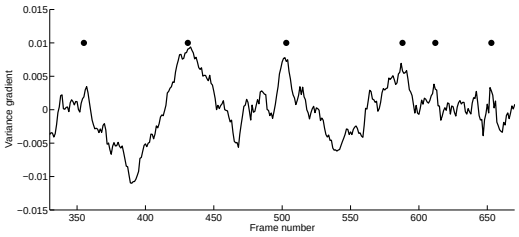

Fig. 8. Gradient of the variance in the *rotation guidance algorithm* over time (INDOOR dataset). Maxima in the function (shown in dots) correspond to nodes in the map.

Fig. 9. Sample node images for the INDOOR dataset (*top*) and OUTDOOR dataset (*bottom*). Nodes correspond to places of high variability in the visual appearance of the scene.

the INDOOR environment, the mission was run seven days after the first visit. For the OUTDOOR dataset, the mission was run at 4pm in the afternoon, while the first visit had happened at 10am in the morning. From a node to the next, the system automatically estimates the user's relative position on the map using the belief state estimator shown on Figure 10. Each time a new node is reached, the system plays back the audio file that was captured at the corresponding location during the first visit and gives rotation guidance to the user in a human-oriented fashion (Section 4). Note that the belief state estimator is reset every time a new node is encountered, which bounds the uncertainty on the user's location over time. Figure 11 shows the relative progress of the user along several consecutive edges on the INDOOR and OUTDOOR dataset. The average standard deviation from ground truth estimated assuming constant speed during the first visit is 3.3 frames for the INDOOR dataset (i.e approximately one second or 1.5 meters) and 7.7 frames for the OUTDOOR dataset (i.e approximately two seconds or 3 meters). The degradation in performance on the OUTDOOR dataset can be explained by the drastic change in lighting between the morning and the afternoon, making the feature matching more challenging.

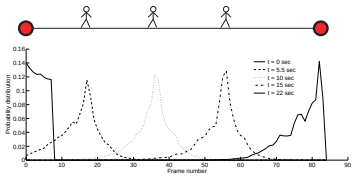

Fig. 10. Propagation of the belief state as the user navigates across the map. The uncertainty in the position estimate is reset every time a node is encountered. Notional user motion on the top.

Fig. 11. Relative progress along several consecutive edges ((a): INDOOR dataset, (b): OUTDOOR dataset). Ground truth estimated using constant speed assumption shown in dotted line. The standard deviation is 3.3 frames for the INDOOR dataset (i.e approximately one second or 1.5 meters) and 7.7 frames for the OUTDOOR dataset (i.e approximately two seconds or 3 meters).

9 Discussion

We described a vision-based navigation guidance system designed to assist a person in navigating through complex spaces. Our method is hybrid. It builds a topological map representation by analyzing the visual appearance of the environment and provides user-centric guidance in a human-oriented way within the map. Our method requires no calibration of the cameras, generalizes to any number of cameras and learns the correlation between user motion and evolution of image features. We have presented the performance of our system on both indoor and outdoor datasets under real conditions.

We are currently working on solving the global localization problem in a scalable way. Also, the current mapping strategy does not scale well since it requires storing all features observed during exploration. Methods based on a vocabulary tree [16] for instance may help build a scalable mapping algorithm. Finally, we are working on a solution to detecting when the user leaves the map unexpectedly and helping them getting back on track.

10 Acknowledgments

We thank Ed Olson for his suggestion of the algorithm underlying Equation 1. Olivier Koch is supported by a University Research and Development award from Draper Laboratory.

References

1. Golledge, R.G., Klatzky, R.L., Loomis, J.M., Speigle, J., Tietz, J.: A geographic information system for a GPS based personal guidance system. Int. J. Geographical Information Science **12** (1998) 727–749
2. Chen, Z., Birchfield, S.T.: Qualitative vision-based mobile robot navigation. In: ICRA, IEEE (2006) 2686–2692
3. Horn, B.: Motion fields are hardly ever ambiguous. International Journal of Computer Vision **01**(3) (April 1987) 259–274
4. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. Computer Vision and Pattern Recognition **1** (2004) 652–659
5. Gluckman, J., Nayar, S.K.: Ego-motion and omnidirectional cameras. In: ICCV '98: Proceedings of the Sixth International Conference on Computer Vision, Washington, DC, USA, IEEE Computer Society (1998) 999
6. Stratmann, I., Solda, E.: Omnidirectional vision and inertial clues for robot navigation. J. Robot. Syst. **21**(1) (2004) 33–39
7. Koenig, S., Simmons, R.: Unsupervised learning of probabilistic models for robot navigation. In: Proceedings of the 1996 IEEE International Conference on Robotics and Automation (ICRA '96). (1996) 2301 – 2308
8. Davison, A.J.: Active search for real-time vision. In: ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, Washington, DC, USA, IEEE Computer Society (2005) 66–73
9. Wang, J., Zha, H., Cipolla, R.: Coarse-to-fine vision-based localization by indexing scale-invariant features. IEEE Transactions on Systems, Man, and Cybernetics, Part B **36**(2) (2006) 413–422
10. Davison, A., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. Pattern Analysis and Machine Intelligence, IEEE Transactions on **29**(6) (June 2007) 1052–1067
11. Sim, R., Elinas, P., Little, J.J.: A study of the Rao-Blackwellised particle filter for efficient and accurate vision-based SLAM. Int. J. Comput. Vision **74**(3) (2007) 303–318
12. Goedemé, T., Nuttin, M., Tuytelaars, T., Gool, L.V.: Omnidirectional vision based topological navigation. Int. J. Comput. Vision **74**(3) (2007) 219–236
13. Cummins, M., Newman, P.: FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. The International Journal of Robotics Research **27**(6) (2008) 647–665
14. Modayil, J., Beeson, P., B. Kuipers, B.: Using the topological skeleton for scalable global metrical map-building. Intelligent Robots and System, 2004. IEEE/RSJ International Conference on **2**(28) (September 2004) 1530 – 1536
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2) (2004) 91–110
16. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, IEEE Computer Society (2006) 2161–2168