Visually Guided Mobile Robots

James J. Little
Department of Computer Science
University of British Columbia, Vancouver, BC Canada
little@cs.ubc.ca

Abstract

Our mobile robots use trinocular stereo sensors to sense their environment, navigate around obstacles in a dynamic, unknown world and partner with humans in various tasks. Fundamental to these abilities is the capacity to build a map and locate themselves relative to it, while operating.

Here we review a collection of methods that permit mobile robots to acquire spatial and visual maps of their environment incrementally, to localize themselves in these maps, and to share these maps when collaborating. The resulting dense spatial and appearance data can be used for visualization, modeling and remote interaction.

Visually guided robots rely almost exclusively on passive light gathering sensors and so can be stealthy as well as small, since they need no emitters and need less power than active sensors. Progress in camera and processor miniaturization will permit embedded devices to see as they act and move about the world.

1 The Robots

1.1 Mobile robots: *José* and *Eric*

We use a Real World Interfaces (RWI) B-14 mobile robot, *José*, for vision-based robot navigation. *José* is equipped with a Pentium^{TM} PC running the Linux operating system as its onboard processing. *José* has a Compaq wireless ethernet modem for communication to a host computer, as well as a Matrox Meteor RGB frame grabber connected to a Triclops trinocular stereo vision camera module. The Triclops stereo vision module was developed at the UBC Laboratory for Computational Intelligence (LCI) and is being marketed by Point Grey Research, Inc. (*www.ptgrey.com*). *José*'s partner, *Eric*, is identically equipped.

The stereo vision module has 3 identical wide angle (90° degrees field-of-view) cameras. The system is calibrated, and corrected for lens distortion and camera misalignment in software to yield three corrected images that conform to a pinhole camera model with square pixels. The camera coordinate frames are co-planar and aligned so that the the epipolar lines of the camera pairs lie along the rows and columns of the images.

José and *Eric*, the mobile robots.

1.2 Trinocular Stereo Algorithm

The trinocular stereo approach is based on the multibaseline stereo developed by Okutomi and Kanade [OK93]. Each pixel in the reference image is compared with pixels along the epipolar lines in the top and left images. The comparison measure used is sum of absolute differences (SAD). The results of the two image pairs (left/right, top/bottom) are summed to yield a combined score. Multibaseline stereo avoids ambiguity because the sum of the comparison measures is unlikely to cause a mismatch—an erroneous minimum in one pair is unlikely to coincide with an erroneous minimum in another pair.

1.3 Mapping with occupancy grids

Occupancy grid mapping, pioneered by Moravec and Elfes [ME85, Elf89], is the most widely used robot mapping technique due to its simplicity and robustness and also because it is flexible enough to accommodate many kinds of spatial sensors. It also adapts well to dynamic environments. The technique divides the environment into a discrete grid and assigns each grid location a value related to the probability that the location is occupied by an obstacle. Initially, all grid values are set to a 50% value (i.e., equal probability for occupied and unoccupied). Sensor readings supply uncertainty regions within which an obstacle is expected to be. The grid locations that fall within these regions of uncertainty have their values increased while locations in the sensing path between the robot and the obstacle have their probabilities decreased. The system also maintains a map indicating which regions have been seen during exploration.

1.4 Constructing top-down views from stereo images

Although occupancy grids may be implemented in any number of dimensions, most mobile robotics applications (including ours) use 2D grids. Much of the 3D data is lost in the construction of a 2D occupancy grid map. The robot possesses 3 DOF (X, Y, heading) within a 2D plane corresponding to the floor. The robot's body sweeps out a 3D volume above this plane. By projecting all obstacles within this volume to the floor, we can uniquely identify free and obstructed regions in the robot's space.

Figure 1 shows the construction of the 2D occupancy grid sensor reading from a single 3D stereo image. Figure 1(a) shows the reference camera greyscale image (160x120 pixels), and (b) the resulting disparity image. White regions indicate image areas which were invalidated and thus not used. Darker areas indicate lower disparities, and are farther away from the camera. Figure 1(c) shows a column-by-column projection of the disparity image (a *radial map*), taking the maximum valid disparity in each column. Figure 1(d) shows these column values converted into distance, and (e) shows these distance values converted into an occupancy grid representation; black indicates the uncertainty region around the object, and white indicates clear regions. The two "spikes" on the left hand portion of Fig. 1(d) were caused by mismatches in the stereo algorithm; their causes and removal are discussed in [ML00]. The process illustrated in Figure 1 generates the input into our stereo vision occupancy grid. The mapping system then integrates these values over time, to expand the map and keep it current in the changing world.

2 Localization in Occupancy Grids

Mobile robot localization and mapping, the process of simultaneously tracking the position of a mobile robot relative to its environment and building a map of the environment, has been a central research topic for the past years. Accurate localization is a prerequisite for building a good map, and having an accurate map is essential for good localization. Therefore, Simultaneous Localization And Map Building (SLAMB) is a critical underlying factor for successful mobile robot navigation in a large environment, irrespective of the higher-level goals or applications. Vision-based approaches using stable natural landmarks in unmodified environments are highly desirable for a wide range of applications.

Great success has been achieved with Markov localization [NPB95, SK95, BFHS96]. RHINO [BCF+98, FBTC98] utilizes a metric version of this approach with laser sensors. Thrun *et al.* [TBF98] proposed a probabilistic approach for map building using the Expectation-Maximization (EM) algorithm. Unlike RHINO, the latest museum tour-guide robot MINERVA [TBB+99] learns its map and uses camera mosaics of the ceiling for localization in addition to the laser scan occupancy map. A Monte Carlo Localization method was proposed in [DBFT99], using sampling-based density representation. Given a visual map of the ceiling obtained by mosaicing, it localizes the robot using a scalar brightness measurement. Recently, Thrun *et al.* [TBF00] proposed a novel real-time algorithm combining the strengths of EM algorithms and

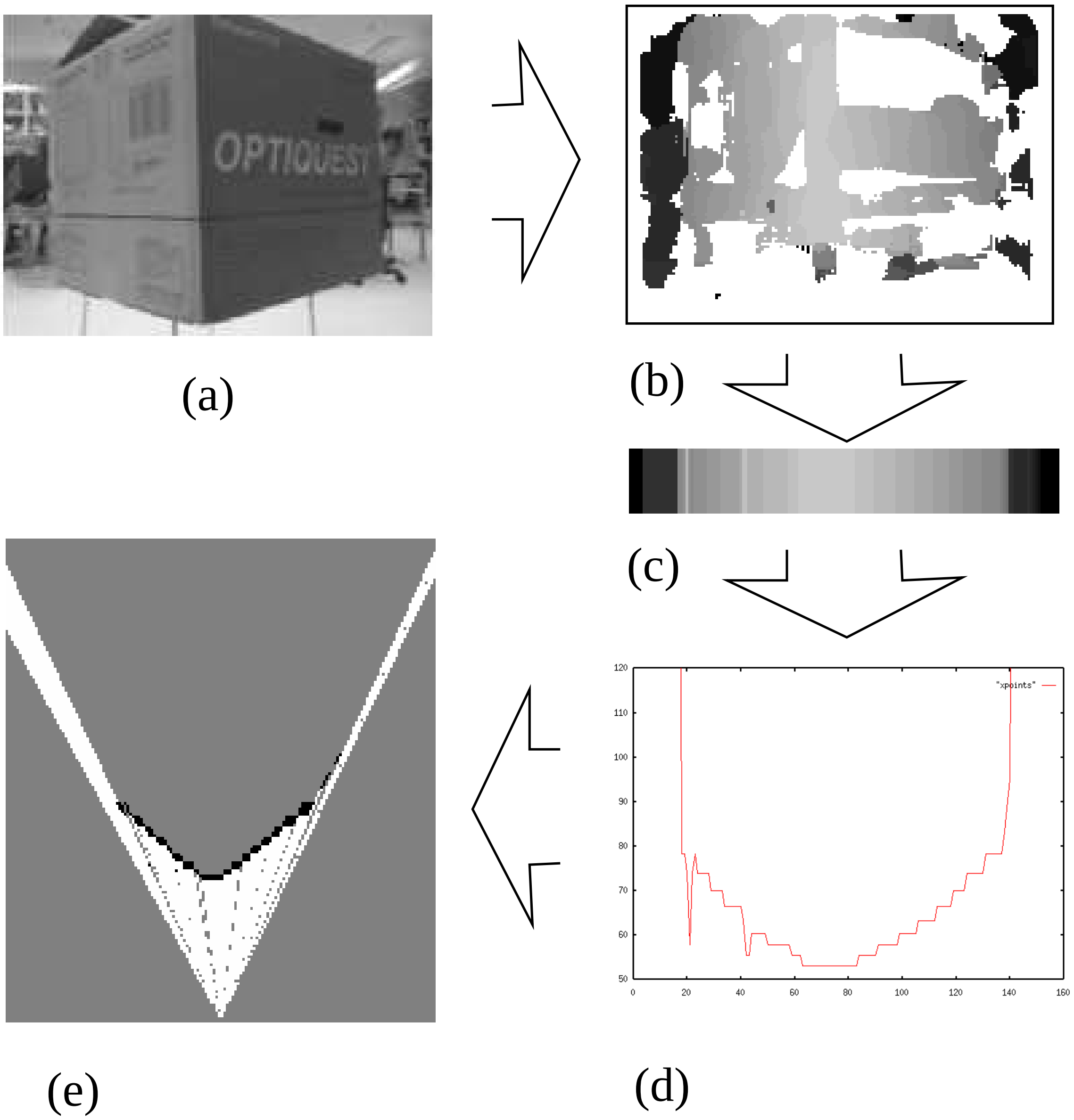
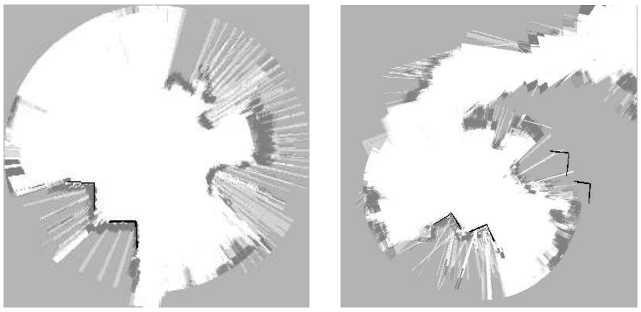


Figure 1: From stereo images to top-down views: (a) greyscale image (b) disparity image (white indicates invalid, otherwise brighter indicates closer to the cameras) (c) the maximum disparity in each column of the disparity image (d) depth versus column graph (depth in centimetres) (e) the resultant estimate of clear, unknown and occupied regions (white is clear, black is occupied and grey is unknown)

incremental algorithms. Concurrent mapping and localization is achieved in 2D using a forward-looking laser and an upward-pointed laser is used to build a 3D map of the environment

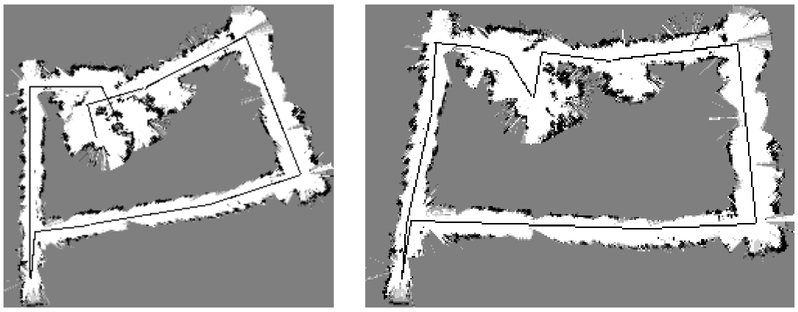
In [LJM98, JML99], Jennings, Murray and Little find landmarks in the occupancy grid itself. To localize visual information in the occupancy grid, we must detect features in the map. Corners are appealing because they provide 2 degrees of freedom of constraint, occur frequently in indoor environments, and stand out well in occupancy grids, despite quantization. We use a least square model matching approach because it is capable of subpixel resolution, fast, and trackable. Moreover, we can augment our occupancy grid with several height levels in which we can find corners, for example in rooms, above the clutter on the floor.

First we find edges: boundaries between empty and solid space in the occupancy grid. Next, we find possible locations for corner models and use these as initial estimates. Third, we do a least-squares fit of a



(a) (b)

Figure 2: (a) shows the initial map with the landmarks located (b) shows the map after returning to home base with the re-localized landmarks. This shows the amount of drift in the odometry readings: $2.4m$ in x and $1.1m$ in y. The total length of the traverse is approximately $82m$.



(a) (b)

Figure 3: The result of a long traverse out the lab door, along the corridor, and back into the lab (a) before correction (b) After correction. The robot path is drawn in by hand. In the original map the drift has caused two areas to merge into one. In the corrected map, the corridors appear more rectilinear.

corner model to the edge data. Finally, we pick the best one.

Once we have found the corner feature, we can use it and its original reference location to update the position and orientation of the robot. The orientation of a single corner is not accurate, so we use two corners and the angle between them. We fit the model to the data by an iterative technique derived from [Low91]. Figure 2(a) shows the occupancy grid created by rotating the robot in place near a pair of corners, and the corners found.

Each cooperating robot maintains a local map and then updates a shared map after observing home base and correcting its odometry. If both robots were to explore and contribute their updates to a common shared map, their independent errors in pose would corrupt the common map.

In this method, each robot initializes itself to a corner landmark (home base). During the traverse, it generates radial maps from stereo and updates its own occupancy grid, while recording its radial maps, as well as its pose when the map is acquired. At the end of a traverse, it has returned to home base, and can then determine its pose relative to the home base. The difference between its actual pose and its pose estimated from the odometry of the vehicle is an error that must be corrected before its data can be merged with the common map. Figure 2(b) shows the map with the corners as seen after the traverse. Although this strategy avoids corrupting the shared map, information is shared only after periodic updates.

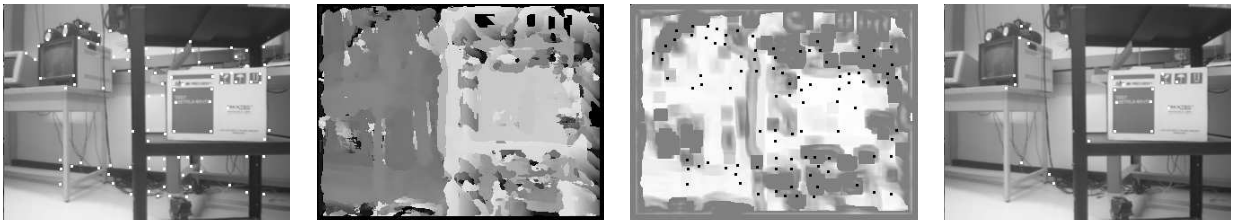
2.1 Rebuilding the map after localization

We adjust the position and orientation of all measurements acquired during a traverse using the robot's pose error. We make the simple assumption that all errors occur uniformly distributed by path length along the traverse. We recompute new poses for each radial map and then add the corrected observations to the shared map. The robot then re-imports the shared map (to receive all correct improvements from other robot partners) and begins a new traverse. Our assumption is overly simple. Odometry errors occur both as systematic drift and incidental slips, distributed randomly. For map correction, we assume that translation errors occur during translation and that all orientation errors occur only during rotation. This last is a bit strong: during translation, orientation can drift, typically 2.5° over a translation of 8m.

Fig. 3(a) shows the occupancy grid map created by a long traverse ($82m$) about our lab. (b) shows the map created with compensated robot positions. The odometry error determined from landmark localization was $2.64m$ and 25.4°. The corrected map has improved, but correction has not straightened the corridors, due to the assumption that orientation errors do not occur during translation. The data could at this point be corrected by the algorithm of [TBF98].

3 Selecting Stable Image Features for Localization Using Stereo

Finding suitable image features for landmarks is complicated by accidental alignments. In [LML98] Little, Murray and Lu show how to analyse dense stereo data to determine scene locations that are locally almost planar. So we reject brightness corners where the surface is not planar; we can with confidence assume that the corners result from surface markings on a nearly planar surface, not the accidental projection of brightness edges separated in space. These selected corners are easier to identify from widely differing viewpoints. The task of matching the corners and then determining pose is greatly simplified. Fig. 4 shows a stereo image with corners (a), and disparities (b). Fig. 4(c) shows the residual errors for planar fits to depth data, and (d) the results of pruning features that do not lie on planar surfaces. Without such pruning, the average number of stable corners (that show over a sequence of frames) is 30.9%. Using stereo to eliminate corners formed by accidental alignments increases the number to 73.8%.



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

Figure 4: (a) Corners (white) on the right image (b) disparity: brighter points are nearer (c) planar fitting error: brighter is lower error. Corners shown in black. (d) right image: corners (white) in planar regions

4 Feature-based Localization

Most existing mobile robot localization and mapping algorithms are based on laser or sonar sensors, as vision is more processor intensive and stable visual features are more difficult to extract. In [SLL00], Se, Lowe and Little propose a vision-based Simultaneous Localization And Map Building (SLAMB) algorithm by tracking SIFT features, which can be detected at a variety of scales. Trinocular stereo obtains the 3D position of the landmarks, so a 3D map can be built and the robot can be localized simultaneously in 3D. The 3D map, represented as a SIFT landmark database, is updated over frames and adaptive to dynamic environments.

SIFT (Scale Invariant Feature Transform) was developed by Lowe [Low99] for image feature generation in object recognition applications. The features are invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine or 3D projection. Key locations are selected at maxima and minima of a difference of Gaussian function applied in scale space.

When mobile robots move in an environment, landmarks may be observed over time, but from different angles, distances or under different illumination. The characteristics of the SIFT features make them suitable landmarks for robust SLAMB. At each frame, we extract SIFT features in each of the three images, and stereo match them. Matched SIFT features are stable and will serve as landmarks for the environment.

Figure 5: (left) SIFT features found, with scale and orientation indicated by the size and orientation of the squares. (right) Stereo matching results: the horizontal line indicates its horizontal disparity and vertical line indicates its vertical disparity. Closer objects will have larger disparities.

Figure 5(left) shows the SIFT features in a stereo image. Scale, orientation and a subpixel location are associated with each SIFT feature. The size of the square surrounding each feature is proportional to the scale at which the feature is found, and the orientation of the squares shows the orientation of the SIFT features. In addition to the epipolar constraint and disparity constraint, we also employ the SIFT scale and orientation constraints for computing stereo matches. Subpixel disparity is obtained for each match. Matches are made horizontally and vertically, and matches whose disparities do not agree are discarded. Knowing the camera intrinsic parameters, we can compute the 3D world coordinates (X, Y, Z) relative to the robot for each feature. They can subsequently serve as landmarks for map building and tracking. Figure 5(right) shows each matched SIFT feature marked with a corner: the length of the horizontal line indicates the horizontal disparity and the vertical line indicates the vertical disparity for each feature.

4.1 Ego-motion Estimation

We obtain $[r, c, s, o, d, X, Y, Z]$ for each stereo matched SIFT feature, where (r, c) is the measured image coordinates in the reference camera, (s, o, d) are the scale, orientation and disparity associated with each feature, (X, Y, Z) are its 3D coordinates relative to the camera. To build a map, we need to know how the robot has moved between frames in order to put the landmarks together coherently. The robot odometry data can only give a rough estimate and it is prone to errors such as drifting, slipping, etc. The odometry allows us to predict the region to search for each match more efficiently for matches.

Once the SIFT features are matched, we can use the matches in a least-squares procedure to compute a more accurate camera ego-motion and hence better localization. This will also help adjust the 3D coordinates of the SIFT landmarks for map building. From the 3D coordinates of a SIFT landmark and the odometry data, we can compute the expected 3D relative position and hence the expected image coordinates and disparity in the new view. The expected scale is computed accordingly as it is inversely related to the distance. We then search for the appropriate SIFT feature match in the next frame, using the disparity constraint together with the SIFT scale and orientation constraints.

Figure 6 shows the match results visually where the shift in image coordinates of each feature is marked. The white dot indicates the current position and the white cross indicates the new position; the line shows how each matched SIFT feature moves from one frame to the next, analogous to sparse optic flow. Fig. 6(a) is for a forward motion of 10cm and Fig. 6(b) is for a clockwise rotation of 5°.

Figure 6: The SIFT feature matches between consecutive frames: (a) for a 10cm forward movement (b) for a 5° clockwise rotation.

4.2 Least-Squares Minimization

Once the matches are obtained, the ego-motion is determined by finding the camera movement that would bring each projected SIFT landmark into the best alignment with its matching observed feature. To minimize the errors between the projected image coordinates and the observed image coordinates, we employ a least-squares minimization [Low91] to compute this camera ego-motion. Although our robot can only move forward and rotate, we use a full 6 degrees of freedom for the general motion. We pass the SIFT features matches in Figure 6 to the least-squares procedure with the odometry as the initial estimate of ego-motion. For between-frame movement over a smooth floor, odometry is quite accurate and can be used to judge the accuracy of the solution. The following results are obtained:

Fig	Odometry	Mean E	Least-Squares Estimate
6(a)	z=10cm	1.328 (pixels)	[1.353cm,-0.534cm,11.136cm, 0.059°,−0.055°,−0.029°]
6(b)	θ=5°	1.693 (pixels)	[0.711cm,0.008cm,-0.9890cm, 4.706°,0.059°,−0.132°]

Each SIFT feature has been stereo matched and localized in 3D coordinates. Its entry in the database is $[X, Y, Z, s, o, l]$, where (X, Y, Z) is the current 3D position of the SIFT landmark relative to the camera, (s, o) are its scale and orientation, and l is a count to indicate how many consecutive frames this landmark has been missed. We maintain this database, add new entries to it, track features and prune entries when appropriate. The landmarks in the database are updated by maintaining a Kalman filter of its pose.

4.3 Experimental Results

We manually drive the robot on a loop in the lab. 3590 SIFT landmarks, with 3D positions relative to the initial robot position, are gathered in the SIFT database. The accuracy of ego-motion estimation depends on the landmarks and their distribution. Here there are sufficiently many matches in each frame, ranging mostly between 40 and 60. When the robot comes back to the original position (0,0,0), judged visually:

SIFT estimate: X:-2.09cm Y:0cm Z:-3.91cm and θ:0.30° α:2.10° β:−2.02°

Compare this ($0.04m$ and 0.30° over $8m$) with the $2.64m$ and 25.4° error due to odometry in a path of $82m$ reported above.

To acquire permanent landmarks, we must distinguish between volatile features and stable features. Discriminating between them remains challenging, because of occlusion by static and moving objects in the environment. To predict occlusion and allow for expected variation in appearance, we associate a view vector with each SIFT feature at a landmark and allow new SIFT features for a landmark dependent on view direction. Over time, if a landmark is observed from various directions, much richer SIFT information is gathered.

4.4 Building a Multiscale 3D Range and Appearance Map

The stereo information obtained by the Triclops sensor is accompanied by rectified aligned greyscale images. This rich source of visual information allows us to build incremental shape and appearance maps. The challenge is to seamlessly merge data gathered at close range with less reliable data acquired at long range; a pixel at 7.5m occupies $25cm^2$ while at 0.5m the surface covered is $0.04cm^2$. Likewise the visual texture sampled at the pixel changes markedly with distance.

Don Murray is building a multiscale surface representation to manage the registered range and appearance we accumulate while the mobile robot's explore our lab. Feature-based localization permits us to identify the robot's pose. His system uses *patchlets* (Fig. 7, top right), small surface areas whose orientation, distance, size, appearance and uncertainty are acquired through stereo, as well as the viewpoint from which it has been acquired. The multiscale patchlet representation will adapt to input resolution and allow local iterative surface construction, as well as resolution-based rendering. Figure 7 shows a rectified image of a hall from Triclops, the patchlet representation, and the cloud patchlets resulting from the stereo view of the hall.

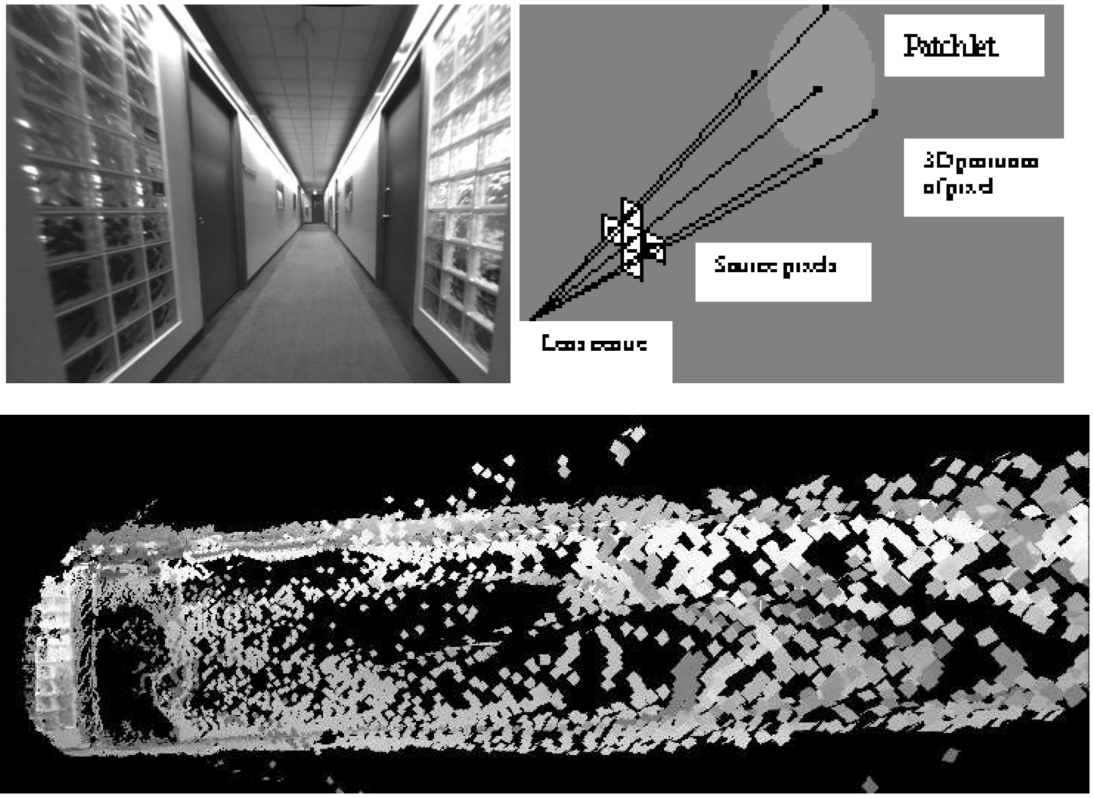


Figure 7: (top) Rectified hall image from Triclops and patchlet representation, showing the lens, source pixels and the 3D position of the pixel. (bottom) Cloud of patchlets (as rectangles) from hall image and registered depth data.

5 Discussion

There are many sources of identifiable landmarks in a visual environment, including corners in depth found in occupancy grids, brightness corners on planar regions and multiscale image features. We have developed several systems to use these landmarks. The most promising is the SIFT-based multiscale landmarks, which occur densely in a typical indoor environment and are not restricted to built environments. Our mobile robots accumulate these landmarks while exploring the environment and use them to localize themselves and correct for odometry errors. We are considering the problem of initializing localization (the kidnappedrobot problem), i.e., re-using the map at any arbitrary robot position by matching the rich SIFT database landmarks.

Maintaining a dense, detailed 3D environment map, provides accurate navigation. Real-time stereo sensors allow safe operation in cluttered, dynamic, unstructured environments. Our visually guided mobile robots have operated for many hours, safely, at crowded venues. Moreover, the dense registered depth and brightness images gathered in real-time while traversing the environment can support resolution-based visualization.

5.1 Acknowledgements

Our work has been supported by the Institute for Robotics and Intelligent Systems (IRIS III), a Canadian Network of Centres of Excellence, as Robot Partners – Collaborative Perceptual Robotic Systems. The research discussed here is joint with Cullen Jennings, David Lowe, Jiping Lu, Don Murray and Stephen Se.

References

[BCF+98] W. Burgard, A.B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI)*, Madison, Wisconsin, July 1998.

[BFHS96] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI)*, pages 896–901, Menlo Park, CA, 1996. AAAI Press/MIT Press.

[DBFT99] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, Fort Collins, CO, June 1999.

[Elf89] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[FBTC98] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers. Position estimation for mobile robots in dynamic environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI)*, Madison, Wisconsin, July 1998.

[JML99] Cullen Jennings, Don Murray, and James J. Little. Cooperative robot localization with vision-based mapping. In *ICRA-99*, pages 2659–2665, 1999.

[LJM98] James J. Little, Cullen Jennings, and Don R. Murray. Vision-based mapping with cooperative robots. In *SPIE-98, Sensor Fusion and Decentralized Control in Robotic Systems*, pages 2–12, November 1998.

[LML98] James J. Little, Don Murray, and Jiping Lu. Identifying stereo corners for mobile robot localization. In *IROS-98*, pages 1072–1077, 1998.

[Low91] D. G. Lowe. Fitting Parameterized Three-Dimensional Models to Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[Low99] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 1150–1157, Kerkyra, Greece, September 1999.

[ME85] H. Moravec and A. Elfes. High-resolution maps from wide-angle sonar. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, St. Louis, Missouri, March 1985.

[ML00] Don Murray and James J. Little. Interpreting stereo vision for a mobile robot. *Autonomous Robots*, 8(2):161–171, 2000.

[NPB95] I. Nourbakhsh, R. Powers, and S. Birchfield. Dervish: An office-navigating robot. *AI Magazine*, 16:53–60, 1995.

[OK93] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, 1993.

[SK95] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1080–1087, San Mateo, CA, 1995. Morgan Kaufmann.

[SLL00] Stephen Se, David Lowe, and James J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. submitted to ICRA-00, 2000.

[TBB+99] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva: A second-generation museum tour-guide robot. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'99)*, Detroit, Michigan, May 1999.

[TBF98] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, 31(5):1–25, 1998.

[TBF00] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *IEEE International Conference on Robotics and Automation (ICRA 2000)*, San Francisco, CA, April 2000.