Mobile Robot Localization using Planar Patches and a Stereo Panoramic Model

Dana Cobzas and Hong Zhang
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
{dana, zhang}@cs.ualberta.ca

Abstract

This paper presents a new type of image-based map for robot navigation formed by panoramic models enriched with depth and 3D planarity information. We take advantage of the scene geometry implicitly contained in the model to localize a mobile robot that is moving in the same environment. The novelty of this model compared to the existing image-based maps is that the motion of the robot is not restricted to a predefined path or to locations close to the original images. Experimental results demonstrate this. We also present a new technique for extracting planar patches using both intensity and sparse disparity information provided by a trinocular vision system.

1. Introduction

One of the most difficult problems in mobile robot research is understanding the surrounding environment and navigating through it. A variety of world representations have been proposed depending on the robot sensors, intended application, and type and size of the environment. Some of the maps are built *a priori* by the human operator [9], while others are automatically built by the robot. In many cases, mobile robots utilize vision sensors, and the techniques can be classified into two major approaches: *geometrical-based maps* and *image-based maps*. The first approach represents the environment using geometrical features and the absolute relationship between them [14, 1]. Extracting and matching features is a difficult vision problem. Image-based maps overcome this problem by storing a collection of images that sample the navigation environment without explicitly extracting a geometric model.

1.1. Image-Based Maps

Images contain rich information about the surrounding environment. That is why a set of images organized in a meaningful way can be used as a map for robot navigation. There are two different approaches in image-based maps. One is to memorize images *along a path* that should then be repeated by the robot, and the other is to memorize images *at fixed locations* as reference points in the navigation environment.

One of the earliest works from the first category (route representation) was created by Tsuji [24, 10] where a panoramic representation of the route is obtained by scanning side views along the route. The robot uses the panoramic representation recorded in a trial move, and the current one for locating itself along the trial route. In [11] the model contains a sequence of frontal views along the route. The robot memorizes, at each position, an image obtained from a camera facing forward, and the directional relation to the next view. An interesting approach is presented in [23] where the route is memorized as 2D Fourier power spectrums of consecutive omnidirectional images at the horizon. The robot position is determined by comparing patterns from memorized Fourier power spectrum with the principal axis of inertia.

The problem with route representation approaches is that the robot has to move along the same pre-stored route. To overcome this problem, omnidirectional images are stored in fixed places of the environment [5, 4, 22, 6]. This representation is very suitable for homing applications where the robot has to move toward a target location. The omnidirectional images used to represent the space are very similar and require a lot of memory space, so they are processed and compressed. Ishiguro [5] transforms the images into the Fourier space; Hong [4] uses a one-dimensional signature of the image assuming that the robot is moving on a plane. Winters [22] and Jogan [6] use an eigenspace representation of panoramic images. Localization is done by projecting the representation of the current image into the eigenspace. Other techniques compare the current image with the stored ones, and find the optimal position. Ishiguro [5] uses a spring model to arrange the observation points according to the environment geometry.

1.2. Problem Formulation

One of the major drawbacks of the existing image-based maps is that the robot motion is restricted to either a predefined route, or to positions close to the original locations of the stored images. This is because they use the appearance of the stored images without considering the geometry behind them. We are proposing here a new type of image based map formed by panoramic models, enriched with depth and

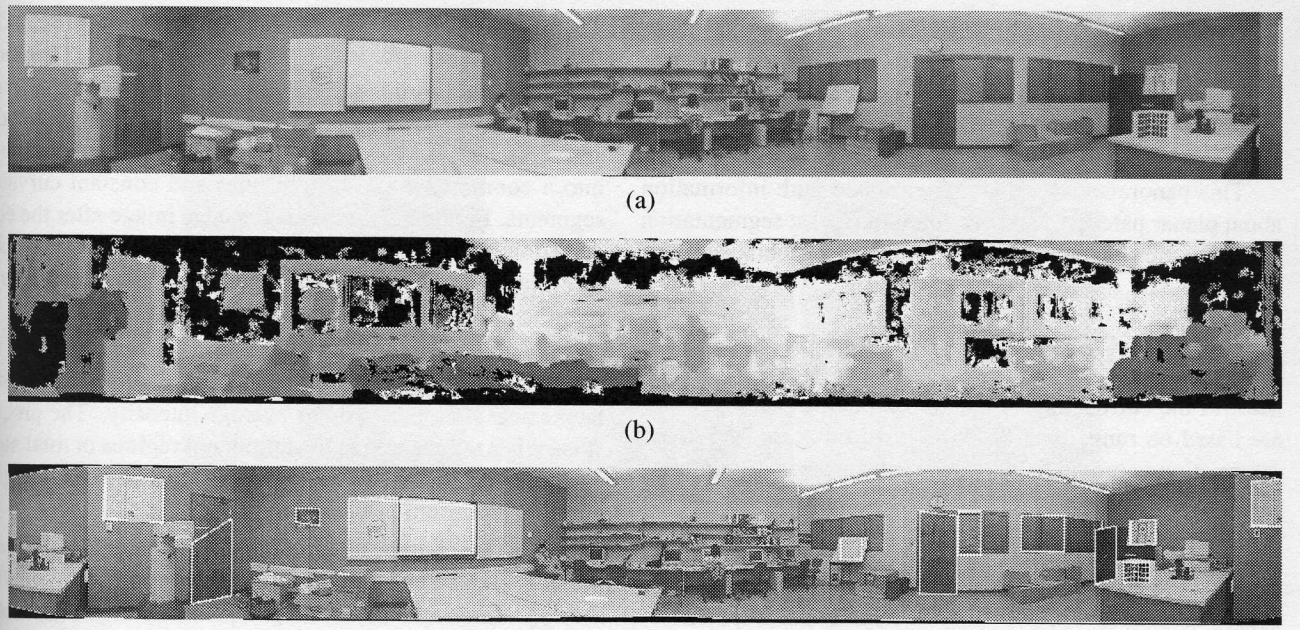

Figure 1: (a) Cylindrical panoramic model; (b) Depth map: dark - close objects; whither - far objects; black - no depth value; (c) Extracted vertical planar patches

3D planarity information, taken in the navigation environment. This model contains detailed information about the navigation space without explicit 3D reconstruction. We use geometric constraints applied to planar patches to localize a robot that is moving in the same environment. The motion is constrained to a plane but is not restricted to positions close to the models.

For acquiring the model we use a trinocular vision system [17]. As a result, one advantage of our model is that, because we use the same sensor to acquire both intensity and depth data, we do not need to register them as in a rendering system based on multiple sensors of different modalities [12].

Our system is mainly designed for indoor navigation where planar patches are naturally occurring and not changing significantly over time, so they will improve the robustness of the localization algorithm. We have designed a special segmentation algorithm that extracts planar patches from the "stereo" panorama.

The rest of the paper is organized as follows. Section 2 presents the panoramic model and Section 3 describes the segmentation algorithm that is extracting planar patches. The localization algorithm is presented in Section 4, and experimental results are shown in Section 5.

2. Panoramic Model with Depth

In this section we will present the process of building a panoramic image-based model by mosaicing. Image mosaicing means merging a collection of images into a larger one [20, 21, 13]. A panoramic mosaic contains a 360° view of the environment and is constructed by composing planar images taken from the same center of projection. This mosaic is geometrically correct because the input images are related by a 2D projective transformation (homography).

For acquiring the images we used a trinocular stereo system provided by *Point Grey Research* [17]. This system consists of three cameras and produces a real time disparity map. We use both intensity and depth information to produce a "stereo" panorama. A similar approach is presented in [8], with the difference that they used two panoramas to produce the depth map, while we use disparity information provided by the trinocular system for each of the images to be composed in the panorama. In this way we have significant quantizations errors in the generated disparity map because of the smaller baseline. This causes problems in the modeling process which we will address in Section 3.

The trinocular system is rotated around the optical center of the reference camera. The intensity images are projected on a cylinder with radius equal to the focal length of the camera, and then correlated in order to determine the amount of rotation between two consecutive images. In the cylindrical space, a rotation becomes a translation, so we can easily build the cylindrical image by translating each image with respect to the previous one. To reduce discontinuities in intensity between images, we weigh the pixels in each image proportionally to their distance to the edge [21].

Along with the intensity cylindrical panoramic image, we also build the corresponding depth map using the disparity values provided by Triclops Stereo System. Because of the

particular geometry of the image, instead of storing depth values, we store, for each pixel with disparity, the distance from the center of the cylinder to the corresponding 3D point. The result of the mosaicing technique is presented in Figure 1(a) and the corresponding "depth" map in Figure 1(b).

This panorama with depth is enriched with information about planar patches. Next section describes a segmentation algorithm that extracts planar features using both intensity and range data.

3. Planar Patch Extraction

Most of the current algorithms for extracting planar regions are based on range data [2, 7, 15]. In our case, the depth information is provided by a stereo system with a relatively small baseline (10 cm), so it is sparse and noisy. This makes the segmentation using only range information almost impossible. To overcome this problem we designed a new segmentation algorithm that is using both intensity and range data.

The main observation that led us to the current algorithm is that in a typical indoor environment, most of the planar regions have an intensity distinct from the surrounding regions. So the first step in the segmentation algorithm is a region growing approach based on average intensity. This algorithm is summarized in Subsection 3.1. Next we use depth information to segment the regions generated by the region growing algorithm based on a planarity test. To compensate the errors in depth data, we use a generalized Hough transform to eliminate the non-coplanar points. Subsection 3.2 describes this planar patch selection approach.

3.1. Intensity Based Segmentation

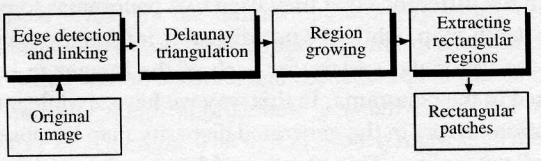

Figure 2: Flow chart for intensity-based segmentation algorithm

The flow chart of the segmentation algorithm is presented in Figure 2. The fundamental structure used by the global region growing algorithm is a triangular mesh. The segmentation algorithm takes place in the image domain so the mesh is also generated in pixel space. We choose a constrained Delaunay triangulation [19] based on edge segments to construct our 2D mesh because it generates a connected mesh with disjoint triangles. The edge segments input to the triangulation algorithm are edges of the resultant mesh. The segmentation algorithm extracts regions with distinct average intensity that should have also distinctive edges.

For edge extraction and linking we used code provided by Dr. S. Sarkar at University of South Florida [18]. Their edge detection algorithm is an adaptation of the optimality criteria proposed by Canny to filters designed to respond with a zero crossing. For edge linking, they segment an edge chain into a combination of straight lines and constant curvature segments. Figure 3(b) presents the edge image after the edge detection and linking algorithm is applied to the original image (3(a)), and Figure 3(c) presents the result of constrained Delaunay triangulation with the edge segments.

The global region growing algorithm starts with the triangular mesh and merges the initial triangular regions into larger ones that have similar average intensity. The process stops when a threshold in the number of regions or total mesh error is exceeded. We used a modified version of the region growing algorithm presented in [2, 7].

From the initial triangular regions, *region adjacency graph* is created, where the vertices represent the regions and the edges indicate that two regions are adjacent. Each edge is weighted by the error given by

$$E_{ij} = \sum_{u,v \in R_i} \frac{I(u,v)}{N_i} - \sum_{u,v \in R_j} \frac{I(u,v)}{N_j} \quad (1)$$

where R_i and R_j are the adjacent regions that share the edge, I is the initial image to be segmented, and N_i represents the number of pixels from region R_i. Larger regions are grown from the initial mesh by merging adjacent regions. At each iteration the two regions that produce the smallest error E_{ij} are merged. This guarantees that the total error grows as slowly as possible. After each merge the adjacency graph is updated.

There are two thresholds for stopping the region growing process. One is the total number of regions and the other is an upper bound for the total error. In our case the first one works better.

The resulting regions are presented in Figure 3 (d). They usually have irregular shapes that can be either concave or convex. We developed a heuristic algorithm that extracts the biggest trapezoid out of a region. The algorithm proceeds by first filling all the interior small holes and then finding the biggest rectangle included in the original region. For easily testing if a certain pixel belongs to the current region or not, we created a black and white image that contains only the current region. We then detect the bigger interior rectangle - R_h - by horizontally scanning the image. The initial rectangle is the longest vertical scan scan line of the current region. This rectangle is extended in both left and right directions till its area stops growing. The procedure is repeated for vertical scan to obtain R_v. The final rectangle is the biggest one between R_h and R_v. This is the expanded up and down into a trapezoid to fit the original region shape. The result of this algorithm is shown in Figure 3 (e).

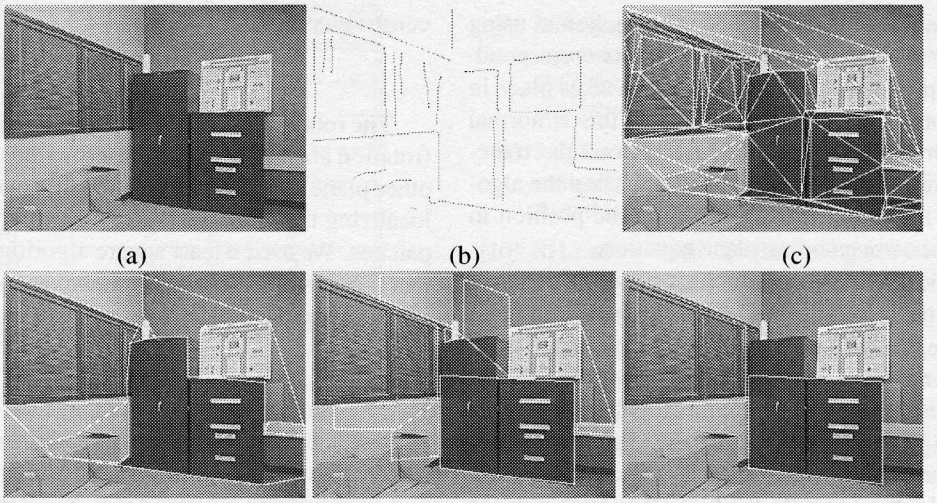

Figure 3: Planar region segmentation (a) Original image; (b) Edge detection and linking; (c) Constraint Delaunay triangulation; (d) Region growing (e) Extracted trapezoidal regions; (f) Vertical planar regions;

3.2. Planar Regions Selection

The trapezoidal regions that result from the intensity based segmentation algorithm are distinct regions not necessary planar. This section describes the algorithm that thresholds these regions based on a planarity error measure and properties of the corresponding 3D plane.

The trinocular system provides 3D information for some of the interior points in each trapezoidal region. This depth data is very noisy, so, before calculating the best fitted plane to region points, we eliminate the outliers using a generalized Hough transform [3]. Figure 4 shows a point cloud near a planar region before (a) and after (b) the Hough transform.

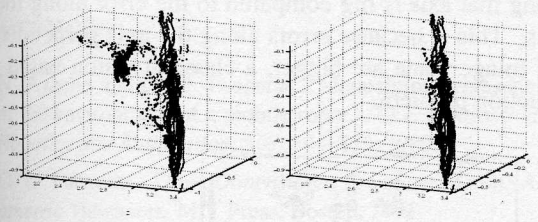

Figure 4: Point cloud for a planar region before (a) and after (b) the Hough Transform

For computing the plane equation of a planar patch, we compute the best fitted plane that approximates the points selected by Hough transform. The plane parameters $(\mathbf{n}, d)$ are determined by minimizing the error measure

$$E = \sum_{i=1}^{N} (\mathbf{n}^T \mathbf{M}_i + d)^2 \qquad (2)$$

where $\mathbf{M}_i$ are the points in the region, N is the number of points, $\mathbf{n}$ in the unit normal of the plane, and d is the distance from the origin to the plane (all in the camera-based coordinate system). This is a classical non-linear optimization problem [2] and the solution for the plane normal $\mathbf{n}_{min}$ is an eigenvector of length one of the covariance matrix $\mathbf{\Lambda}$ associated with the smallest eigenvalue λ, which is also the minimum error. The covariance matrix is given by

$$\mathbf{\Lambda} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{A}_i \mathbf{A}_i^T, \ \mathbf{A}_i = \mathbf{M}_i - \mathbf{M} \ and \ \mathbf{M} = \frac{1}{N} \sum \mathbf{M}_i$$

The minimum distance to the best fitted plane is given by

$$d_{min} = -\frac{1}{N} \sum_{i=1}^{N} \mathbf{n}_{min}^T \mathbf{M}_i$$

After computing the plane that best approximates the points in each region, we decide if this is a real planar patch by thresholding the plane error - Equation (2), the number of points with disparity relative to patch size, and patch size in the image space. Since a typical indoor environment is dominated by vertical walls, and these walls are quite useful for the localization, we extract only planes that are almost vertical and add them to the image model. Figure 3 (f) shows an example of the extracted vertical planar patches.

We used this segmentation algorithm to extract planar patches out of the cylindrical panoramic model. For each planar patch we store its position in the panoramic image and the corresponding plane equation. The result is presented in Figure 1(c).

4. Localization

Having a panoramic image-based model with extracted planar patches, we want to find the position and orientation of

a robot with respect to the model's coordinate system using the current image observed by the robot and its corresponding disparity map. We assume that the motion takes place in a plane (the floor). For indoor environments this is normal because the floor is almost flat. We first extract the trapezoidal planar patches from the current image using the algorithm (Section 3). For each patch, we store the position in the image and the corresponding plane equation.

The localization algorithm uses at least two pairs of corresponding planar patches in the model and the image captured by the robot to be localized. For matching the planar patches, we compare their average intensities and then, if the difference in intensities is below a certain threshold, we correlate a middle portion of the trapezoidal regions. For each planar patch in the current image, we choose as the corresponding patch in the model the one with the highest correspondence score and the lowest difference in intensity, if it exists. If the light has not changed since the model was taken, this algorithm gives very good results.

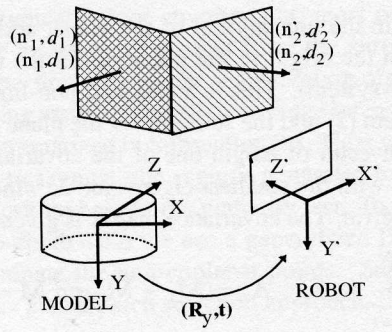

Figure 5: Localization using planar patches

The localization problem proceeds as follows (see Figure 5). Given two or more planes of known positions and orientations with respect to different coordinate frames, compute the position and orientation of one coordinate system relative to the other, assuming motion in the (X, Z) plane only.

Let us consider a plane that has parameters in the model coordinate frame $(\mathbf{n}, d)$ and in the image coordinate frame $(\mathbf{n}', d')$. The equations of this plane in the two coordinate frames and their relationship are the following:

$$\begin{aligned} \mathbf{n}^T\mathbf{P} + d &= 0 \\ \mathbf{n}'^T\mathbf{P}' + d' &= 0 \end{aligned} \tag{3}$$

and,

$$\mathbf{P} = \mathbf{R}_y\mathbf{P}' + \mathbf{t} \tag{4}$$

where $\mathbf{R}_y$ is a rotation about Y axis, and $\mathbf{t} = [T_x, 0, T_z]^T$ is a translation in (X, Z) plane. By substituting Equation (4) into the plane equations and comparing them we get the following constraints:

$$\begin{aligned} \mathbf{n} &= \mathbf{R}_y\mathbf{n}' \\ \mathbf{n}^T\mathbf{t} + d - d' &= 0 \end{aligned} \tag{5}$$

The rotation $\mathbf{R}_y$ can be computed from the first constraint (rotation about Y axis), but for the translation $\mathbf{t}$ we need another plane non-parallel with the first one. So for completely localizing the robot we need at least two non-parallel planar patches. We used a least square algorithm to solve this problem if more than two planar patches are available. The results are further refined by minimizing

$$E = \sum_{i=1}^{N} dist(\mathbf{n}, \mathbf{n}') + \sum_{i=1}^{N} |d - d'|$$

where N is the number of planar patches, and $dist(\mathbf{n}, \mathbf{n}')$ is the distance between the normal vectors $\mathbf{n}, \mathbf{n}'$. We solved the minimization problem using a Levenberg-Marquadt nonlinear minimization algorithm [16]. The next subsection presents the results of the localization algorithm.

5. Experimental Results

For evaluating the localization algorithm we took images in seven locations around the panoramic model. In order to demonstrate that our system works for positions that are not necessarily close to the model, four of the chosen locations are at more then 3 m away from the model. Figure 6 represents the original locations and computed locations, together with the positions of the planes that were used for localization. The errors in X, Z and rotation angle α for each position are listed in Table 1. The dimension of the room is 10 m x 8 m. The average error was about 30 cm in position and 5° in orientation. We have noticed that if the orientation of the robot is close to one of the principal axes, the position error along that axis is big compared to the error along the other axis. This is because errors along the axis perpendicular to the image plane are bigger, a classical vision-based reconstruction problem.

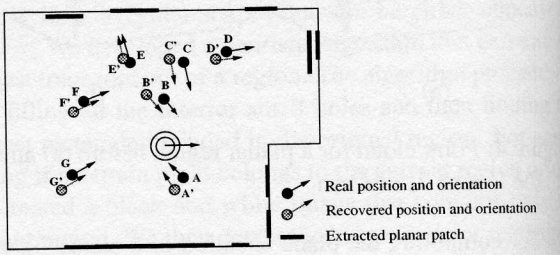

Figure 6: Results for localization experiments.

6. Conclusions and Future Work

This paper presents a new type of image-based map for robot navigation formed by panoramic models enriched with

	ErrX (cm)	ErrZ (cm)	Err(α) (deg)
A	-39	8	3
B	-9	50	2
C	-4	-23	8
D	-11	25	9
E	-3	-9	6
F	-20	33	6
G	-40	34	5

Table 1: Error for recovered position and orientation

sparse depth and 3D information about planar patches. We used geometric constraints applied to planar patches to localize a robot that is moving in the same environment. The novelty of this approach over the existing image-based maps is that, by considering the geometry contained in the images, we do not constrain the motion of the robot to a predefined path or to locations close to the models. We also developed a new type of segmentation algorithm that uses both intensity and depth information for extracting planar patches.

One of the major limitations of the current approach is that the intensity-based segmentation algorithm is very sensitive to illumination changes. We want to overcome this problem by using a precise depth sensor (laser range-finder) and more efficiently extract planar surfaces only from depth data. This might also solve the time problem currently caused by the slow Hough transform algorithm.

In the future we want to make use of multiple image-based models for improving the accuracy of the extracted planes. We also want to consider vertical lines as features and compare the performance of a line-based localization algorithm with the current one.

References

[1] N. Ayache and O. D. Faugeras. Maintaining representation of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, 1989.

[2] O. D. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Boston, 1993.

[3] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1993.

[4] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. Rseman. Image-based homing. *IEEE Control Systems*, pages 38–44, 1992.

[5] H. Ishiguro and S. Tsuji. Image-based memory of the environment. In *Proc. of IEEE International Workshop on Robots and Systems (IROS'96)*, pages 634–639, 1996.

[6] M. Jogan and A. Leonardis. Panoramic eigenimages for spatial localization. In *Proc. Computer Analysis of Images and Patterns (CAIP'99)*, 1999.

[7] S. B. Kang, A. Johnson, and R. Szeliski. *Extraction of Concise and Realistic 3-D Models from Real Data*. Technical Report CRL 95/7, Cambridge Research Laboratory, 1995.

[8] S.B. Kang and R. Szeliski. 3D scene data recovery using omnidirectional multibaseline stereo. In *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'96)*, pages 364–370, 1996.

[9] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.

[10] S. Li and S. Tsuji. Qualitative representation of scenes along route. *Image and Vision Computing*, 17:685–700, 1999.

[11] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 83–88, 1996.

[12] D. K. McAllister, L. Nyland, V. Popescu, A. Lastra, and C. McCue. Real-time rendering of real world environments. In *Proc. of Eurographics Workshop on Rendering*, Spain, June 1999.

[13] L. McMillan and G. Bishop. Plenoptic modeling: Am image-based rendering system. In *Computer Graphics (SIGGRAPH'95)*, pages 39–46, 1995.

[14] H. P. Moravec. The stanford cart and the CMU rover. *Autonomous Robot Vehicles*, pages 407–419, 1990.

[15] B. Parvin and G. Medioni. Segmentation of range images into planar surfaces by split and merge. In *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR'86)*, pages 415–417, 1986.

[16] W. H. Press, B.P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipies in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 1992.

[17] Point Grey Research. *http://www.ptgrey.com*.

[18] S. Sarkar. Lola edge detection and linking code. *http://marathon.csee.usf.edu/ sarkar/vision_html/lola_code/*.

[19] J. R. Shewchuk. Triangle-a two-dimensional quality mesh generator and delaunay triangulator. *http://www.cs.cmu.edu/ quake/triangle.html*.

[20] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, pages 22–30, March 1996.

[21] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *Computer Graphics (SIGGRAPH'97)*, pages 251–258, 1997.

[22] N. Winters and J. Santo-Victor. Mobile robot navigation using omnidirectional vision. In *Proc. 3rd Irish Machine Vision and Image Processing Conference (IMVIP'99)*, 1999.

[23] Y. Yagi, S. Fujimura, and M. Yachida. Route representation for mobile robot navigation by omnidirectional route panorama fourier transform. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1250–1255, 1998.

[24] J. Y. Zheng and S. Tsuji. Panoramic representation for route recognition by a mobile robot. *International Journal of Computer Vision*, 9(1):55–76, 1992.