

Merging Panoramic Intensity and Range Maps for Robot Localization

Dana Cobzas, Hong Zhang and Martin Jagersand

*Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada*

Abstract

We present a novel navigation map formed from a panoramic image-based model registered with depth data provided by a laser rangefinder. From the map a sparse geometric model is derived and used to perform incremental robot localization. The image information is used to disambiguate the feature matching problem. The novelty of this system is that it avoids the difficult problem of detailed 3D reconstruction from image information and performs localization using only a 2D camera on the robot. (The laser range is used only once in the model building step) Experimental results in indoor map building and localization demonstrate the performance of the algorithm.

1 Introduction

A challenging and important problem in mobile robotics research is how to use wide area sensing such as 3D laser range and 2D vision to model unstructured environments and then use those models for localization and navigation. The navigation models may contain different levels of detail, varying from a complete CAD model to a simple graph representing the inter-connectivity between key elements in the environment. Most of the early vision-based navigation systems rely on spatial geometry and precise metric measurements for modeling the space. Without an *a-priori* model it is in general difficult to create a precise metric map that contains the level of detail required for robot localization algorithms [8, 1].

Recently appearance-based solutions have emerged as an alternative to geometric maps. These involve “memorizing” the navigation environment using images or templates. By associating the templates with commands or controls the robot can be driven along a memorized route [5] or to a goal position [4, 6, 14]. One of the major drawbacks of the existing appearance-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.

Here we propose a new type of navigation map that combines an appearance map with a sparse geometric map. It is formed by panoramic image-based models augmented with depth and a sparse set of 3D line features. This model contains detailed information about the navigation environment without explicit full 3D reconstruction. We propose an incremental localization algorithm that uses image information for matching and geometric information for estimating robot position.

In a previous paper [2] we presented a model where the depth information was acquired using a trinocular vision system. The localization was performed by matching 3D features acquired by the robot at the moment of navigation with 3D features in the model. Here we present a new localization method which uses only a 2D camera on the robot. The depth map obtained in [2] had inaccuracies caused by the small baseline of the vision system (10 cm). The approach presented here generates a very accurate depth map but the separate sensors now require registration between depth and intensity data. We propose an image-based algorithm for range-intensity registration. The next section presents the model generation and the data registration algorithms, Section 3 describes the incremental-localization algorithm and Section 4 shows experimental results.

2 Model construction

2.1 Range data acquisition

The data acquisition system consists of a *laser rangefinder* mounted on a *pan-tilt unit* with a *camera* attached on top of it (see Figure 1). We use the rotating camera to build the panoramic image-based model and the laser to acquire range information. The two types of data have to be registered to generate the final model.

The laser rangefinder is a commercial AccuRange

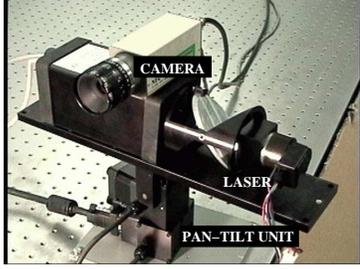


Figure 1: System configuration: the laser-rangefinder with the camera attached on top of it is mounted on a pan-tilt unit

4000-LIR from Acuity Research, Inc. We attach it to a pan-tilt unit (model PTU-46-70) from Directed Perception, Inc. The data returned for each rangefinder sample consist of range r , amplitude a and angular position of the rotating mirror θ . In addition to these we store the pan angle ϕ at which the scan was taken, so the raw data for each reading is (r, a, θ, ϕ) . We sample the range/amplitude data on spherical grids and represent them as spherical images, ignoring the small translation between the mirror center of rotation and the pan-tilt unit center. To clean (filter) the range data we eliminate all the values that are inconsistent with more than four neighbors or bigger than the dimension of the room. The range finder does not generate uniformly sampled data. To fill missing values in a uniform grid, we apply a 3×3 averaging filter over the neighboring samples. Figure 2 (top) shows the spherical range image representing 180° scan of the navigation environment.

2.2 Panoramic image-based model

A panoramic mosaic is constructed by composing planar images taken from a single viewpoint. This is geometrically corrected because there is no parallax between the input images.

The camera is mounted on top of the laser rangefinder. It rotates together with it and acquires images every 10° . The images are projected on a cylinder with radius equal to the focal length of the camera and then correlated in order to precisely 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 [12]. The corresponding 180° panoramic mosaic for the range scan is presented in Figure 2 (bottom).

2.3 Range-intensity data registration

The registration of volumetric and intensity data is an important problem especially in the fields of model building and realistic rendering. Most of the proposed solutions are recovering the rigid transformation between the sensors using point or line features [11, 9]. This is in general a non-linear problem and requires an initial estimate to converge. In contrast, image-based techniques compute a direct mapping between the points in the data sets to recover the transformation. The accuracy of this method depends on the original assumptions about the physical system (e.g. affine camera or planar scene) but in general they are good for locally recovering a mapping between data sets. In a recent paper [3] we compared the two approaches and find that image-based methods are fast and adequate for application that does not require a high precision alignment. In the setup presented here the two sensors are very close to each other an image to image warp is suitable for aligning the two data sets.

The first step in the registration algorithm is to project the spherical range data into a cylindrical representation with the radius equal with the focal length of the camera (same as the panoramic mosaic). This mapping is given by

$$P(r, \theta, \phi) \mapsto P(r, \theta, f \tan \phi) = P(r, \theta, h) \quad (1)$$

where r represents the distance from the center of the cylinder to the point, h is the height of the point projected on the cylinder, θ is the azimuth angle and f the focal length of the camera. Again, we sample this data on a cylindrical grid θ, h and represent it as a cylindrical image. The same procedure is applied to the amplitude data and get the cylindrical amplitude image.

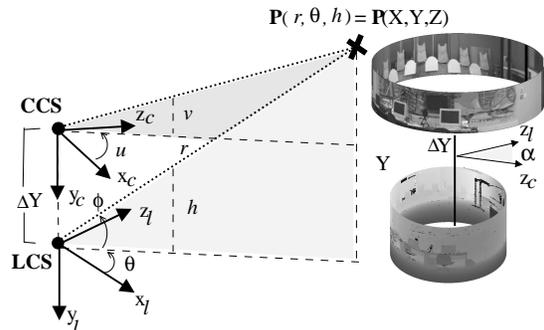


Figure 3: The projection of a space point P in the cylindrical image (θ, h) and the panoramic mosaic (u, v) . We approximate the laser-camera transformation with a translation ΔY and a rotation over y axis.

Having the data in similar cylindrical image rep-



Figure 2: (top) Spherical representation of the range data from an 180° scan after filtering (bottom) Corresponding 180° panoramic mosaic.

representations we compute a *global mapping* between them. We approximate the physical configuration of the sensors as in Figure 3 assuming only a vertical translation ΔY and a pan rotation between the two reference coordinate systems LCS (laser coordinate system) and CCS (camera coordinate system). For a point $p_l(\theta, h)$ in the cylindrical laser image its corresponding point in the panoramic mosaic $p_c(u, v)$ is

$$\begin{aligned} u &= a\theta + \alpha \\ v &= f \frac{Y - \Delta Y}{r} = f \frac{Y}{r} - f \frac{\Delta Y}{r} = bh - f \frac{\Delta Y}{r} \end{aligned} \quad (2)$$

where a and b are two warp parameters that will account for difference in resolution between the two images, α aligns the pan rotation and $Y = r \frac{h}{\sqrt{f^2 + h^2}}$ is the height of the 3D point $P(r, \theta, h)$. For our setup we have $f = 1000$ pixels, $\Delta Y = 5$ cm and the range of the points is $r = 5 - 8$ m, so $f \frac{\Delta Y}{r} = 5 - 8$ pixels and it can be approximated to a constant $-\beta$. The general warp equations are:

$$u = a\theta + \alpha, \quad v = bh + \beta \quad (3)$$

We compute the warp parameters (a, b, α, β) from two or more corresponding points in the two images using a least square approach.

After the global mapping the two data sets are only approximately aligned with a misalignment of 5 – 7 pixels. We perform a *local alignment* using a set of corresponding control points. The local map “stretches” the range data to fit the intensity data using cubic interpolation based on a 2D Delaunay triangulation of the control points.

For visualizing the performance of the registration algorithm, we re-render the navigation room from different positions than the one where the model was taken. We triangulate the registered range points on the panoramic image using a 2D Delaunay triangulation and render the corresponding image triangles

using OpenGL. In a complete mesh, some of the triangles might not represent physical planes but are artifacts of occluding contours, and in most of the cases this appears at silhouette edges [7] where points from the object are connected with background points. To avoid this phenomenon we eliminate all the triangles within a threshold that are parallel to the viewing direction. Figure 4 shows the rendered view after global mapping (top) and with additional local alignment (bottom). Notice that the misalignment of the texture on the computer monitor is corrected with the local mapping technique.

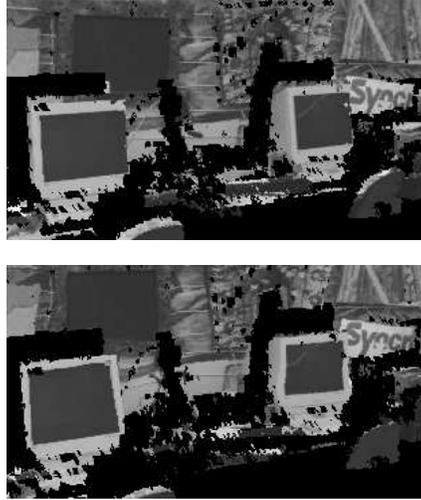


Figure 4: (top) Rendered image using only the global mapping (bottom) Rendered image with local alignment. Notice how misalignment from the top edge of the left monitor is compensated.

2.4 Model vertical lines

The proposed localization algorithm is using vertical line features. We choose vertical lines because when

projected on the cylindrical image they remain vertical and are not transformed into curves as would horizontal or arbitrary lines. Consequently a standard edge detection and linking algorithm can be used to detect vertical line segments.

A set of vertical line segments are manually selected in the cylindrical amplitude image. The segment points are projected on the panoramic mosaic using the above mentioned registration algorithm and then a line segment is fitted to them. Figure 5 shows the selected lines in the amplitude image (top) and the projected lines on the panorama (bottom). The small misalignment is the error of the registration algorithm. The navigation image-based map consists of the panoramic mosaic, the 3D vertical line segments and the corresponding lines segments in the panorama.

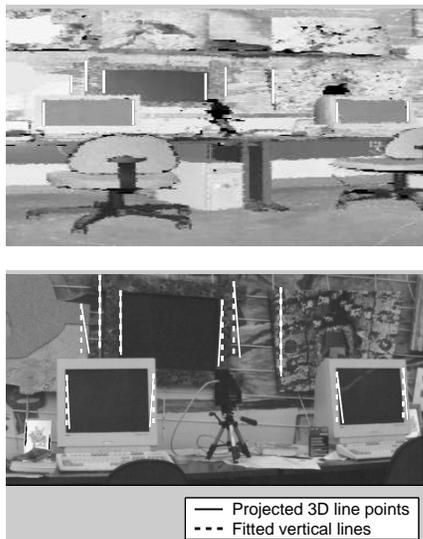


Figure 5: Selected vertical lines in the laser amplitude image (top) and Projected 3D lines in the panoramic mosaic and the fitted vertical lines (bottom)

3 Robot localization

Having the image-based model with the detected line features, the localization problem involves finding the position and orientation of the robot with respect to the center of the model using the current robot view. We assumed planar motion, which is reasonable for indoor environments where motion takes place on the floor.

We are doing an incremental localization where the current position is approximately known either from the previous position assuming motion continuity or from other sensors (odometry). An initial position and the height difference between the model loca-

tion and the robot has to be estimated at the beginning using for example manual selected corresponding feature points. The first step in the localization algorithm is to detect vertical line segments in the current image. Using the approximate position of the robot, model 3D segments are projected on the current view. The *matching* algorithm will look in the vicinity of each projected segment for a vertical edge with similar intensity characteristics. The current position is updated from the corresponding model-image segments (*localization* step). The incremental localization approach assumes that the localization position is approximately known, limiting the matching search region, which contrast it from a global localization approach.

3.1 Localization using vertical line segments

A 3D line can be represented in term of a unit vector \mathbf{v} which indicates the direction of the line and a vector \mathbf{d} that represents a point on the line. For a vertical line $\mathbf{v} = [0, 1, 0]^T$ and \mathbf{d} can be chosen as the intersection of the line with the horizontal plane $\mathbf{d} = [X, 0, Z]^T$. Any point on the line can be expressed as $\mathbf{P}_l = [X, k, Z]^T$ where k is a real parameter that is restricted to an interval in case of a line segment. A vertical image line can be characterized by the column coordinate u , and a point on the line has the form $\mathbf{p}_l = (u, q)$, where q is a parameter similar to the 3D case.

We denote the unknown displacement between the model coordinate system (MCS) and the current image position (CIP) by (R_y, \mathbf{t}) , where R_y is a rotation matrix over Y axis, and $\mathbf{t} = [t_x, H, t_z]^T$ (H is the height difference between the model and the robot). The vertical line points $\mathbf{P}_l = (X, k, Z)$ expressed in MCS are projected on the image using:

$$\mathbf{p}_l = C(R_y \mathbf{P}_l + \mathbf{t}) \quad (4)$$

where the camera matrix C has the form:

$$C = \begin{bmatrix} a_u & 0 & u_c \\ 0 & a_v & v_c \\ 0 & 0 & 1 \end{bmatrix}$$

Using Equation 4, the column coordinate of the projected line can be derived as:

$$u_{proj} = a_u \frac{X \cos \alpha + Z \sin \alpha + t_x}{-X \sin \alpha + Z \cos \alpha + t_z} + u_c \quad (5)$$

where α is the pan rotation angle.

If N lines are available, we compute the motion parameters (α, t_x, t_z) by minimizing the displacement between the projected and detected lines in the image.

$$(\alpha, t_x, t_z) = \min_{\alpha, t_x, t_z} \sum_{i=1}^N (u_{proj}^i - u^i)^2 \quad (6)$$

We solve the non-linear least square problem using Levenberg-Marquardt non-linear minimization algorithm.

3.2 Matching vertical lines

For detecting the image vertical lines we apply an edge extraction and linking algorithm. We used code provided by Dr. S. Sarkar at University of South Florida [10]. From the extracted line segments only the vertical ones are further selected.

Having the approximate position of the robot we project the model 3D line points in the current image using equation 4. The matching algorithm finds for each model line the corresponding image vertical line by searching the vicinity of the projected line. Together with the 3D line points the map has the corresponding lines in the panoramic model. We formulate a matching criterion that picks the image line closest in intensity with the model line. More precisely the match of model line j is

$$\Delta I_{ij} = |I_L^i - \Upsilon_L^j| + |I_R^i - \Upsilon_R^j|$$

$$match_j = \min_i(\Delta I_{ij}) \quad (7)$$

where I_{LR} and Υ_{LR} represents the average intensity at the left and right side of the edge for a 5 pixel strip respectively in the current image and the panoramic mosaic and i indexes the extracted image lines from the vicinity of the projected model line. In our experiments we considered a 80×60 search window for a 320×480 image. If ΔI_{match_j} is above a threshold we consider to be no match for model line j . Figure 3.2 shows the detected vertical line segments, the projected model lines and the matched lines.

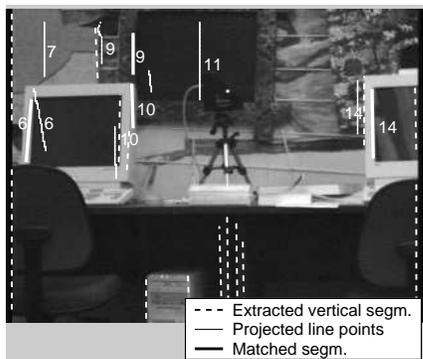


Figure 6: Illustration of matching algorithm: detected vertical lines, projected lines and detected matches. Numbers indexes lines in the model and show the matched pairs.

4 Experimental results

To evaluate the model accuracy and the performance of the localization algorithm we took 26 images along

two lines 10 cm apart from each other and recovered their positions using the localization algorithm. The camera was calibrated using the Tsai algorithm [13]. We considered two methods for selecting the corresponding feature lines in the current image a *manual* one and the *automatic matching* algorithm presented in Section 3.2. We used the values determined by the manual approach to feed the approximate position in the matching algorithm. The manual method will evaluate the accuracy of the model and localization algorithm and the automatic one the overall performance of the localization system. All the measurements are relative to the model reference system. For initializing the position we manually selected corresponding feature points in the model and first image and perform a point-based localization algorithm [2]. As mention before this will also recover the difference in height between the model and the camera. Figure 4 presents the recovered positions using manual (top) and automatic (bottom) matching.

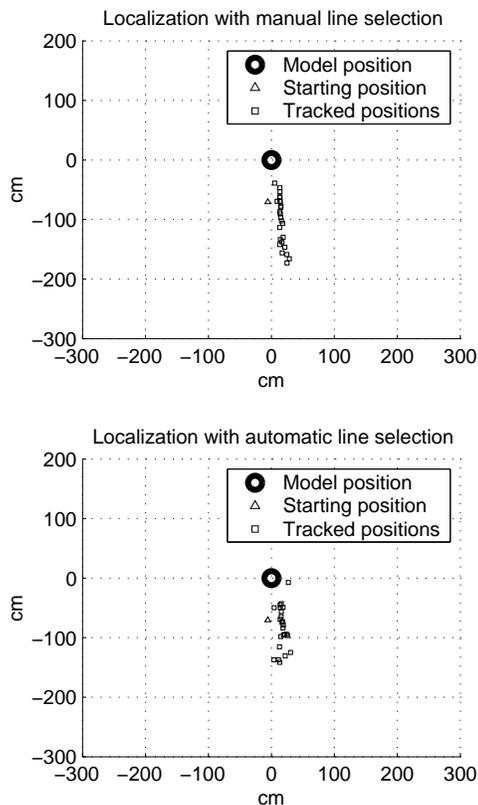


Figure 7: Recovered positions using localization algorithm with manual (top) and automatic (bottom) selection of line features. The dimension of the room is marked in cm.

We measured the relative accuracy along the two lines - the error in distance between consecutive recovered positions δ assuming we approximately

marked positions every 10 cm, and the deviation from a straight line ρ along each of the two path segments. The following table presents the computed measurements, both in cm. The results for the localizing positions on the first line segment are superior to the ones for the second one because the accuracy of the 3D line features is better.

	Distance $\delta(cm)$		Deviation $\rho(cm)$	
	Manual	Autom.	Manual	Autom.
1 st line	3.2	4.06	0.72	1.5
2 nd line	3.92	5.35	0.95	2.98
Mean err	3.56	4.7	0.83	2.24

Table 1: Localization accuracy for manual and automatic selection: error in distance between consecutive positions $\delta(cm)$ (ground truth 10 cm) and deviation from straight line $\rho(cm)$

To test the algorithm sensitivity with respect to position of the features in 3D and current image, we added different levels of uniform noise to the 3D lines and the manually selected feature points. We compute the average distance from the originally detected locations. The results are presented in Figure 4 show that a bad estimation of the 3D line has more influence on the reconstructed position than the error in the located image feature.

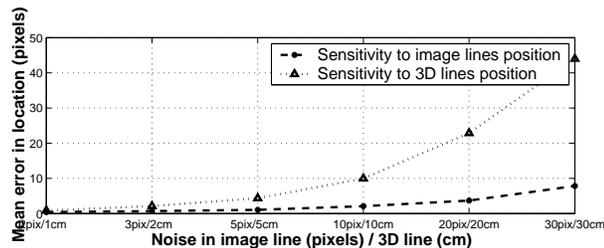


Figure 8: Sensitivity of the localization algorithm with respect to features position in image vs. 3D.

5 Conclusions and future work

We have presented a new type of navigation map that combines appearance information with sparse geometric information about the navigation environment. The map is formed by a panoramic image mosaic with the corresponding depth map and 3D line features. The range data acquired using a laser rangefinder is registered with the intensity data using an image-based approach. We proposed an incremental localization algorithm that uses image information for feature matching and geometric information for estimating robot position. The localization method is performed using only a camera on

the robot and its accuracy is superior to the previous approach from [2]. The mean position error was reduced to about half.

In the future we want to improve the localization algorithm by embedding a statistical model of the uncertainty in the robot location. We also want to extend the algorithm to arbitrary line features.

Acknowledgments

We appreciate the help from Steve Sutphen in setting the acquisition system.

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] D. Cobzas and H. Zhang. Cylindrical panoramic image-based model for robot localization. In *Proc. of IROS*, 2001.
- [3] D. Cobzas, H. Zhang, and M. Jagersand. A comparative analysis of geometric and image-based volumetric and intensity data registration algorithms. In *Proc. of ICRA (to appear)*, 2002.
- [4] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. Rseman. Image-based homing. *IEEE Control Systems*, pages 38–44, 1992.
- [5] S. Li and S. Tsuji. Qualitative representation of scenes along route. *Image and Vision Computing*, 17:685–700, 1999.
- [6] 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.
- [7] 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.
- [8] H. P. Moravec. The stanford cart and the CMU rover. *Autonomous Robot Vehicles*, pages 407–419, 1990.
- [9] M. D. Wheeler R. Kurazume and K. Ikeuchi. Mapping textures on 3d geometric model using reflectance image. In *Proc. of the Data Fusion Workshop in IEEE, ICRA*, 2001.
- [10] S. Sarkar. Lola edge detection and linking code. http://marathon.csee.usf.edu/sarkar/vision_html/lola_code/.
- [11] I. Stamos and P. K. Allen. 3D model construction using range and image data. In *Proc. of CVPR*, 2000.
- [12] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment map s. In *Computer Graphics (SIGGRAPH'97)*, pages 251–258, 1997.
- [13] R. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-self tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.
- [14] 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.