# Planar Patch Extraction with Noisy Depth Data

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

## Abstract

*This paper presents an algorithm for extracting planar patches by integrating both intensity and range data provided by a stereo system. For dealing with noisy and sparse range data, the initial segmentation is based on intensity information, and then the resulted regions are thresholded using depth data. This new algorithm, different from the existing ones that use only range data in the segmentation process, produces accurate planar patches that are then used for building a panoramic image-based model for mobile robot navigation.*

## 1. Introduction

Modeling the world is important for many computer vision, robotics or virtual reality applications including robot navigation, object recognition, architectural modeling, flight simulation *etc*. In order to have a realistic model of the environment, data is usually collected using both vision and range sensors, and then integrated into a 3D model. Some of the above mentioned applications, like virtual reality, require a complete 3D reconstruction, whereas for others (*e.g.* robot navigation), a partial model of the surrounding world is enough. We describe in this paper our work on building image-based environment models augmented with sparse 3D planar patches.

### 1.1. Related Research

Most of the current algorithms for extracting planar regions are based on range image data. Several algorithms are known in the literature for segmenting range images in planar regions. They can be classified into three main categories: *region growing*, *split-and-merge* and *clustering* methods. An experimental comparison of the range image segmentation algorithms [2] has shown that this problem is far from being "solved".

Region growing approaches start with a fine segmentation of the initial image and then bigger regions are grown based on similar plane equations. The initial segmentation can be obtained by fitting a plane to each pixel based on an $N \times N$ window [2], or by generating a mesh from the original 3D points using traditional graphics techniques like Delaunay triangulation [3, 8]. The approach presented in [6, 9] is based on the observation that, in the ideal case, the points on a scan line that belong to a planar surface form a straight 3D line segment. Therefore one can first divide each scan line into straight line segments and then perform region growing using the set of line segments. In [8] the boundary of the final regions is further simplified in order to accommodate noise in the range data.

In [10], the split-and-merge approach originally proposed by Horowitz and Pavlidis [5] for intensity images was extended to range images. Schmitt and Chen [14] propose a new split-and-merge algorithm that uses a Delaunay triangulation as the basis for an adaptive surface approximation technique.

Clustering represents another class of algorithms for segmenting range data into planar regions. The feature space in which the clustering takes place differs from one algorithm to the next and can be a pixel [7], or a pixel and the estimated normal [2].

All these algorithms use a best fitted plane to range data to verify the planarity of a region. An interesting texture-based planarity constraint is presented in [11]. Having an initial triangulation of the scene in two images, the planarity of each triangle is determined based on the correlation between the texture of matched triangles after rectifying them to a common viewpoint.

### 1.2. Problem Formulation

Most of the current systems segment the range image data into planar regions, which are then integrated into a 3D model. They rely on dense and accurate depth information that is usually acquired using laser or structured light range finders. However what happens if the range data is *sparse* and *noisy* like the depth provided by a stereo system? This paper presents a system that detects planar patches based on both intensity and depth data provided by a trinocular stereo system. The trinocular system uses traditional correlation techniques for extracting the depth data, so it generates disparity map only for textured areas. Our system is designed for indoor robot navigation, where most of the planar patches, like doors, walls, cabinets, either have regular or do not have any texture, but they are visually distinctive in the intensity image. Our planar patch extraction algorithm first extracts rectangular regions based on intensity values, and then use depth information to derive the planar regions. The resulting 3D model is a partial reconstruction of the world, witch we use for robot localization.

Depth from stereo was used before [8] to segment planar regions, but in their case the depth data was more accurate because of the bigger baseline. In our case, because of the small baseline (10 cm), depth data is very noisy and accurate segmentation required by the robot navigation application is not possible by using only depth data. The main contribution of our work is the integration of both intensity and range information in the planar patch extraction algorithm.

The rest of the paper will be organized as follows. Section 2 presents the algorithm for extracting planar patches, and Section 3 describes the application of the algorithm for modeling a navigation environment for a mobile robot and the results of a localization algorithm that is using this model.

# 2. Planar Patch Extraction

In this section we will present the segmentation algorithm that extracts planar patches. For acquiring the depth and intensity data we used a trinocular system offered by *Point Grey Research* [12]. This system consists of three cameras and produces a real time disparity map. The main problem with stereo systems is that, because of the quantization errors, depth data can be very accurate for close objects and less accurate for far objects. The Triclops stereo software allows subpixel accuracy when calculating the disparity map, but in our experiments we found that the error in depth for a distance of approximately 10m is about 0.5m. Another problem is that they provide depth/disparity information only for regions with rich texture. A typical pair of image/disparity map is presented in Figure 1. All these problems make the extraction of planar patches based only on depth data almost impossible.

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 2.1. 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

Figure 1: Intensity and disparity image provided by Triclops. In the disparity image: white - no disparity value; dark - far objects; light - close objects

transform to eliminate the ”bad” points. Subsection 2.2 describes this planar patch selection approach. Subsection 2.3 presents some experimental results for evaluating the robustness of the algorithm.

## 2.1. Intensity Based Segmentation

The flow chart of the segmentation algorithm is presented in Figure 3. 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 constraint Delaunay triangulation [15] 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.

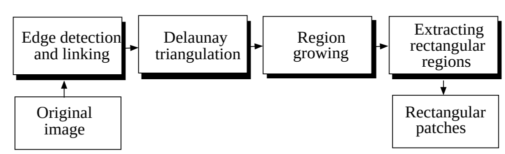



Figure 3: Flow chart for intensity-based segmentation algorithm

For edge extraction and linking we used code provided by Dr. S. Sarkar at University of South Florida [13]. 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 2(b) presents the edge image after the edge detection and linking algorithm is applied to original image (Figure 2(a)), and Figure 2(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

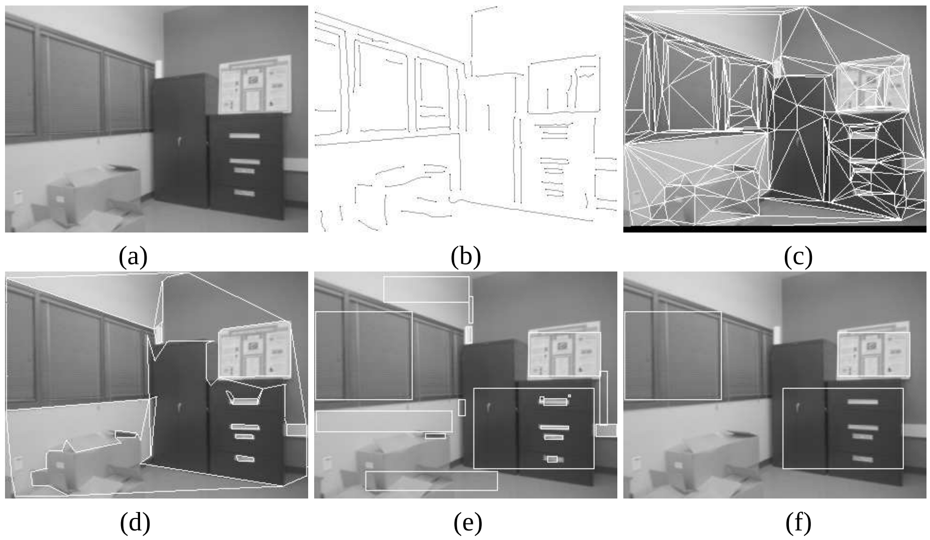



Figure 2: Planar region segmentation (a) Original image; (b) Edge detection and linking; (c) Constraint Delaunay triangulation; (d) Region growing (e) Extracted rectangular regions; (f) Vertical planar regions;

stops when a threshold in the number of regions or total mesh error is passed. We used a modified version of the region growing algorithm presented in [3, 8].

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

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

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 2(d). They usually have irregular shapes that can be either concave or convex. We developed a heuristic algorithm that extracts the biggest rectangle 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 is growing. The procedure is repeated for vertical scan to obtain $R_v$. The final rectangle is the biggest one between $R_h$ and $R_v$. The result of this algorithm is shown in Figure 2(e).

## 2.2. Planar Regions Selection

The rectangular 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 planarity error and properties of the corresponding 3D plane.

The trinocular system provides 3D information for some of the interior points in each rectangular region. This depth data is very noisy, so, before calculating best fitted plane to region points, we eliminate the outliers using a generalized Hough transform [4].

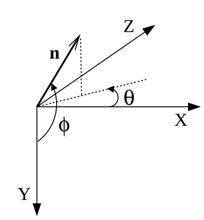
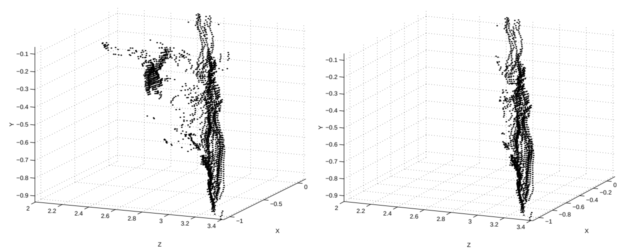



Figure 4: Spherical coordinates for normal to the plane

Consider the plane equation

$$X \sin\phi \cos\theta + Y \cos\phi + Z \sin\phi \sin\theta + \rho = 0 \tag{2}$$

where $\rho$ is the distance to the plane from the origin, and $\phi$ and $\theta$ are the spherical coordinates of the normal (see Figure 4). There is an infinite number of planes that pass through the point $(X, Y, Z)$, but they all satisfy Equation 2. By fixing $X, Y, Z$ and considering $\phi, \theta, \rho$ as the parameters of Equation 2, we will have a surface in the parameter space corresponding to the point $(X, Y, Z)$. All the points belonging to a certain plane will have the corresponding surfaces in the parameter space intersecting at one point $(\phi, \theta, \rho)$ which represents the parameters of that plane. We subdivide the parameter space in cells according to expected ranges of $\phi, \theta, \rho$ and then compute the corresponding cells for each 3D point by incrementing $\phi$ and $\theta$ and calculating $\rho$ from equation 2. The range for $\phi$ and $\theta$ is $[0°, 180°]$.



(a) (b)

Figure 5: Point cloud for a planar region before (a) and after (b) the Hough Transform

The parameters of the cell that contains the largest number of points give an approximation of the plane fitted to the region points, so we will keep only these points for computing the exact equation of this plane. Figure 5 shows a point cloud 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

$$E = \sum_{i=1}^{N} (\mathbf{n}^T \mathbf{M}_i + d)^2 \qquad (3)$$

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. This is a classical non-linear optimization problem [3] 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 $\lambda$, 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 3), number of points with disparity relative to patch size, and patch dimension in the image space. For the robot navigation application we are interested in extracting only planes that are almost vertical so another threshold criterion is the normal angle with the horizontal plane. Figure 2(f) shows the extracted vertical planar patches.

## 2.3. Results and Plane Equation Evaluation

To compare the performance of our algorithm with others, we implemented Kang and Sziliski's range-based segmentation method [8]. The result is presented in Figure 6 that shows the algorithm's inability to handle noisy range data. For example the poster from the right part of the image is considered to belong to the same plane as the cabinet below that is about 0.5 m in front of it. This is because of the noise present in the depth data resulting from the trinocular system. Figure 2(f) shows the output of our segmentation method where the poster and the cabinet are correctly separated into two distinctive regions.

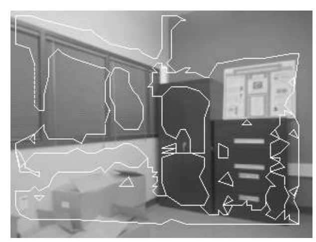


Figure 6: Results of a range data segmentation algorithm

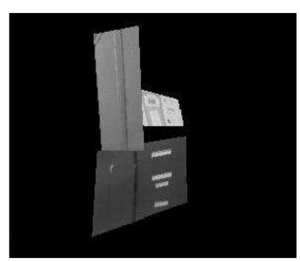


Figure 7: Image with the planar patches rendered from a different point of view

After applying the segmentation algorithm, for evaluating the correctness of the plane equation for the extracted planar patches, we produce an image rendered from a different view point than the original one. For producing the rendered image, we first project the corresponding 3D points for the corners of each rectangular patch on the evaluated plane, and then re-project them from a new viewpoint. All the interior points of the rectangular region from the original image belong to the same plane so, by texture mapping the interior of each patch we obtain a geometrically correct image. Figure 7 shows an example of a rendered image for the three extracted planar patches from Figure 2(f).

# 3. Application to Panoramic Image-Based Models

To demonstrate the usefulness of our planar patch region extraction algorithm, we applied it to the construction of a panoramic image-based model for indoor robot localization. We created the image-based model of the navigation environment, and then compute the position and orientation of a robot by matching the current captured image with the model. We used the planar patches as features in the localization process. The next subsection describes the formation of the cylindrical panoramic model, and Subsection 3.2 shows the experimental results.

## 3.1. Panoramic Model with Depth

For panoramic model construction, we use the trinocular system mentioned before [12] that 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 [16].

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 mosaicking technique is presented in Figure 8(a) and the corresponding "depth" map in Figure 8(b).

We used the algorithm described in Section 2 to extract rectangular 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 8(c).

We then used this panoramic image-based model to find the position and orientation of a robot with respect to the model coordinate system from the current image observed by the robot and its corresponding disparity map. We first extract the rectangular planar patches from the current image using the same algorithm (Section 2). The localization algorithm uses pairs of corresponding planar patches in the model and the image to be localized. We assume that the motion takes place in a plane (the floor). For indoor environments this is normal because the floor is almost flat. Details of the localization algorithm are described in another paper [1]. Next subsection presents the results of the localization algorithm.

## 3.2. Experimental Results

For evaluating the localization algorithm and demonstrate the model accuracy we took images in four locations around the panoramic model and then recover their position using the localization algorithm. Figure 9 represents the original locations and computed locations and the position of the planes that were used for localization. The dimension of the room is 10 m $\times$ 8 m. The average error was about 30 cm in position and $5^{\circ}$ in orientation. Both are small for navigation process.

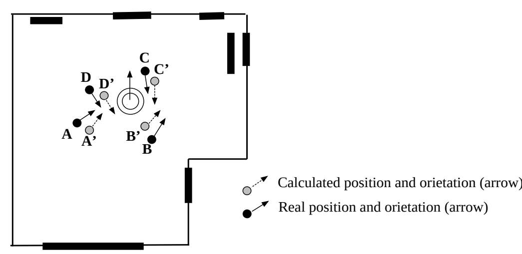



Figure 9: Results for localization experiments.

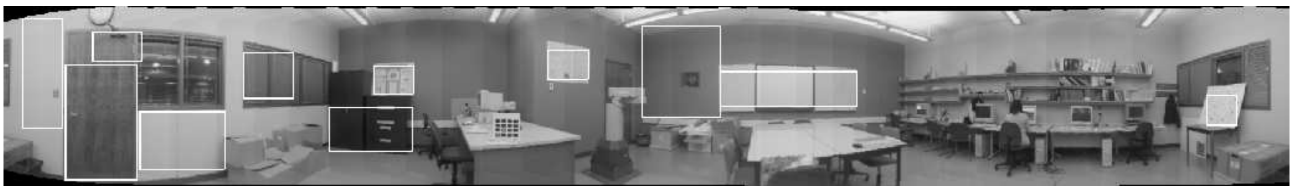
# 4. Conclusions and Future Work

This paper addressed the problem of extracting planar patches by integrating intensity and depth data provided by a stereo system. For compensating the sparse and noisy depth data, we first segment the intensity image and then we used depth information to threshold the resulted regions. This new algorithm, different from the existing ones that use only range data in the segmentation process, produces accurate planar patches that are used in a mobile robot localization application.

In the future, we want to improve the intensity based segmentation algorithm in order to extract any type of quadrilateral in the image space, not only rectangles, for better handling planar rectangular surfaces in the 3D space. We

(a)

(b)



(c)

Figure 8: (a) Cylindrical panoramic model; (b) Depth map: dark - close objects; whither - far objects; black - no depth value; (c) Extracted vertical planar patches

also want to improve the matching process by compensating light changes and make use of multiple image-based models for improving the accuracy of the extracted planes.

# References


[1] D. Cobzas and H. Zhang. Mobile robot localization using planar patches and a stereo panoramic model. *Submitted to Vision Interface 2001 (VI2001)*, 2001.

[2] A. Hoover *et al*. An experimental comparison of range image segmentation algorithm. *IEEE Trans. PAMI*, 18(7):637–689, 1996.

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

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

[5] S. L. Horowitz and T. Pavlidis. Picture segmentation by a direct split and merge procedure. In *Proc. of 2nd International Conference on Pattern Recognition (ICPR'74)*, pages 424–433, 1974.

[6] X. Y. Jiang and H. Bunke. Fast segmentation of range images into planar regions by scan line grouping. *Machine Vision and Applications*, 7(2):115–122, 1994.

[7] J.-M. Jolion, P. Merr, and S. Bataouche. Robust clustering with application in computer vision. *IEEE Trans. PAMI*, 13(8):791–802, 1991.

[8] 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.

[9] E. Natonek. Fast range image segmentation for servicing robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 406–411, 1998.

[10] 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.

[11] J. S. Perrier, G. Agam, and P. Cohen. Physically valid triangulation of scarcely matched images using texture information: application to view-synthesis. In *Proc. of Vision Interface (VI'2000)*, pages 233–240, 2000.

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

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

[14] F. Schmitt and X. Chen. Fast segmentation of range images into planar regions. In *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 710–711, 1991.

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

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