A Proximity Compatibility Function among 3-D Surfaces for Environment Modelling

Liscano, Ramiro; Elgazzar, Shadia; Wong, A.K.C.

A PROXIMITY COMPATIBILITY FUNCTION AMONG 3-D SURFACES FOR ENVIRONMENT MODELLING*

RAMIRO LISCANO Ramiro.Liscano@nrc.ca
SHADIA ELGAZZAR elgazzar@iit.nrc.ca
Institute for Information Technology
National Research Council
Ottawa, Ont. K1A 0R6
CANADA

ANDREW. K. C. WONG akcwong@watnow.uwaterloo.ca
Department of Systems Design
University of Waterloo
Waterloo, Ont. N2K 3G1
CANADA

Abstract

This article defines a method for computing a proximity compatibility function among fragmented 3-D surfaces for environment modelling. Fragmented surfaces are a common occurrence after the segmentation process has been applied to 3-D sensory data, in particular for data taken from large indoor environments. This proximity compatibility function among surfaces gives an indication on how close the surfaces are to each other based on a common gap defined between the boundaries of the surfaces. This particular approach performs most of the computations in the 2-D image plane and when required will use the 3-D information in the data. This is simpler than tackling the whole problem in Euclidean space and as an added bonus, sections of the algorithm pertaining to the 2-D image plane can be applied to classical 2-D intensity images.

Keywords: Surface proximity, Surface grouping, Environment modelling

1 Introduction

Research in the domain of computer modelling from 3-D data has primarily focussed on the extraction of 3-D surfaces and volumetric primitives for the purpose of either object recognition or creating more precise models from 3-D sensory data of machined parts [7, 13]. These type of objects can easily be carried and placed in a controlled environment and scanned using a high resolution active sensor. This is significantly different from the modelling of large indoor environments where it is necessary to bring the sensor to the environment, changing the characteristics of the sensed data dramatically. The result is that nearly all scans taken in these environments consist of fragmented data and it becomes necessary to develop algorithms that can reason among the fragmented surfaces.

Research in the modelling of indoor environments has primarily focussed on the incremental synthesis of sensor views and/or position estimation of the sensor [15, 14, 1]. For these systems to become viable tools for Computer Aided Design (CAD) it is necessary to develop approaches that hypothesize the formation of more composite features from the surfaces.

Recently, there has been interest in the interpretation of scenes taken from 3-D data primarily for the creation of models of indoor environments [9, 12, 5]. All of these approaches rely on the ability to determine the connectivity among the surfaces. In some circumstances the sensory data appears relatively clean with little or no gaps [12] and connectivity can easily be determined. While in the other examples [9, 5] gaps were present but little is said about the algorithms used for determining surface adjacency and proximity.

This article presents an approach for computing a proximity compatibility function among 3-D surfaces. This proximity compatibility function can be used by other algorithms as a measure of confidence in grouping the surfaces. In particular to this research, the proximity compatibility function was used in the design of a Bayesian Network for the grouping of 3-D surfaces [10]. The approach presented can be separated into 2 distinct operations; the determination of adjacent surfaces along with a common area of interest between them, and the calculation of a proximity factor that can be used for determining how close the surface boundaries are to each other.

2 Determining adjacent surfaces

It is necessary to define the context in which the proximity measure is to be applied since it is possible to have several different types of proximity measures among 3-D surfaces. This particular proximity measure takes advantage of the fact that the range data

*NRC Number 40179.

is stored as a 2-D image of the 3-D points taken from the sensor's view point. This implies that the data points are placed in a particular order determined primarily from the scanning characteristics of the sensor, i.e. they are not a cloud of 3-D points. When the data is viewed with respect to the camera there cannot be any overlap among the surfaces, and therefore any proximity measures that rely on surface to surface overlap cannot be applied. For this type of sensory data a proximity measure based on the surface boundaries is more appropriate. The result of this is that the majority of the computations required to determine adjacent surfaces and their respective gaps are performed in the 2-D image plane, and only at the end of the process are the 3-D values used for computing the proximity measure.

The approach is rather simple in theory but in practice smoothing operators and approximations to the boundaries must be applied to reduce the effect of noise on the image. The procedure for determining a common gap is as follows: For any surface in an image extract a boundary that represents the border of that surface. For each pixel on the boundary determine its adjacent surface. Collect the boundary points that share a common adjacency to each other and reorganize them so as to define a common polygon between the surfaces.

2.1 Extracting a surface's boundary

The data was acquired using a compact laser camera called BIRIS [2]. Figure 1 is an intensity image associated with the 3-D scan of a corner of the laboratory. Planar surfaces are extracted from the 3-D points using an algorithm based on a hierarchical segmentation procedure developed by Boulanger [3]. The result of that algorithm is an image of data points grouped into a set of planar surfaces as shown in figure 2.

An edge tracking algorithm is applied to each of the surfaces depicted in the labeled image shown in figure 2. The edge tracking algorithm is an extension of the one developed by Gao et al [8] in that it computes an estimate for the curvature of the edge as it is tracking the boundary of a surface. The curvature along the edge is computed as a difference in a running average of the gray level gradient values along the boundary. Currently this filter uses the average of 3 pixel gradient values and appears to be able to filter the majority of large changes in the gradient values which are due primarily to the discretization of the image. The high curvature points are used to define a polygon which can be used as another form of representing the 3-D polygons, in particular for virtual reality modelling.

Not every high curvature boundary point is used in defining the polygon representation for the surfaces.

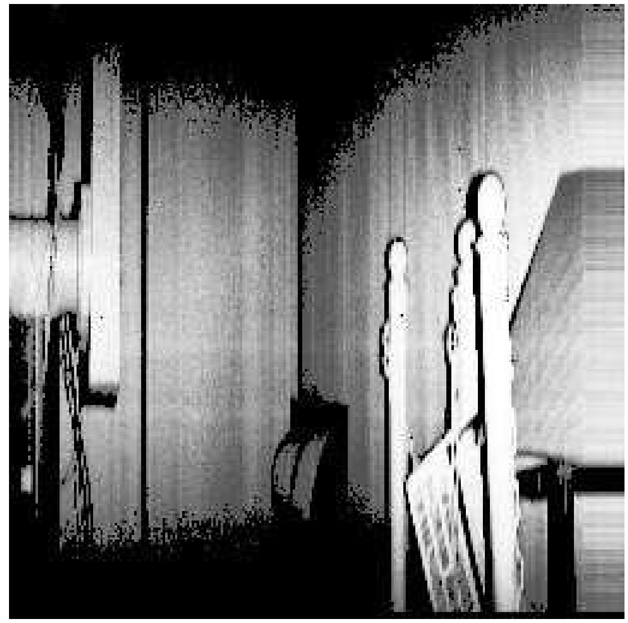

Figure 1: Intensity image of a scan from a laboratory.

Any sequential set of high curvature points is replaced by a straight line defined by the first and last high curvature points in that segment. The result of this is a polygon, as depicted for the surfaces in figure 3, whose corners are high curvature points (represented as white pixels) connected by straight lines or low curvature edges (represented as gray pixels). This procedure of by-passing a set of consecutive high curvature points can be justified by the fact that in most cases these set of points are associated with virtual boundaries caused by weak sensor values and so little information is lost replacing one virtual boundary with another straighter one.

This reduced set of high curvature points are used to define a 3-D polygon which can be used to represent the surfaces for virtual reality modelling. Figure 4 is a view of the surfaces, taken from a VRML (Virtual Reality Markup Language) browser, that were defined using the reduced set of high curvature points. For this display the number of points used in defining the polygons was reduced by a factor of 10 over using all the high curvature points. This is a significant improvement in the performance of the VRML browser.

2.2 Determining adjacent surfaces

Being able to jump across the gaps results in determining more possible adjacent segments than were originally computed using direct contact as a criterion. This added information is important when dealing with fragmented surfaces. For example, surface S_{17} is an isolated surface but using this particular

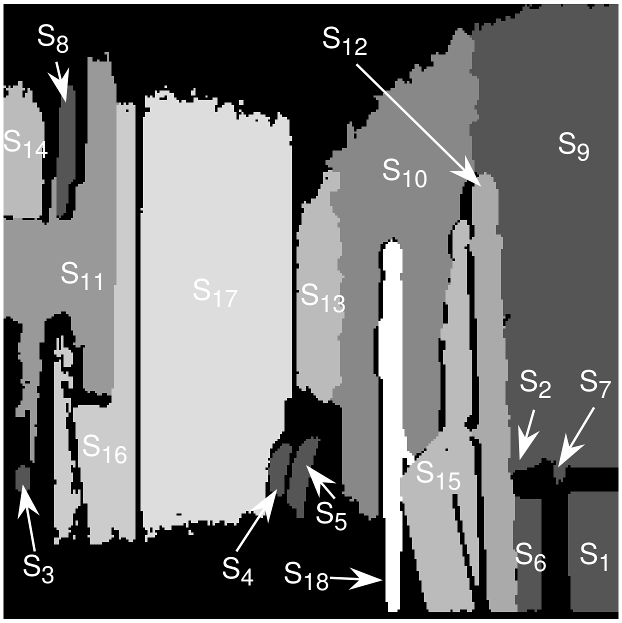

Figure 2: Surface label map of a scan from a laboratory.

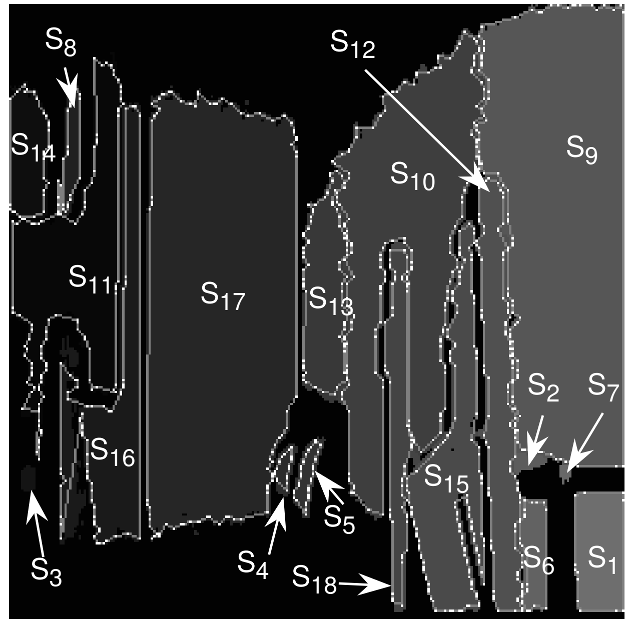

Figure 3: Surface boundaries of a scan from a laboratory.

projection algorithm resulted in 5 other possible adjacent surfaces S_4, S_5, S_{10}, S_{13}, and S_{16} and therefore a greater possibility of being able to join the surfaces.

Adjacent surfaces can be determined by projecting from each of the boundary's pixels in a direction away from the surface. At a global perspective, the boundaries of a surface that were formed due to a weak sensor signal maintain some directional information about the continuation of that surface. This can be observed in the top and bottom edges of surface S_{17} in figure 3. This direction of the boundary is affected by the perspective transformation of the sensor but their effect is fairly minimal since their projection tends not to result in an intersection with another surface. Boundaries that are real boundaries, formed by intersections or jump edges, maintain their directional information and appear as edges with less noise. At a local perspective, direction of the segments that make up a boundary can vary dramatically. This effect is reduced substantially by choosing to represent the boundary as a polygon since the large deviations in the gradient that occur between sequential high curvature points are removed.

The points at which the normal projections of a boundary intersect with another surface's boundary are labeled as the neighbour points of the first boundary. These are recorded and are used to compute a common gap between the 2 adjacent surfaces. While the points along a particular boundary are represented as a linked list of points, their respective neighbouring points have no particular ordering. This is demonstrated in figure 5 where all the ordered points between $P_i^{S_1}$ to $P_m^{S_1}$ have neighbouring pixel values on surface S_2, but the order of those corresponding neighbouring points is not apparent and must be determined.

2.3 Determining the common gap

An approximation for the shape of the gap between the surfaces is done by collecting the neighbour points and ordering them such that a polygonal surface is defined common to both surfaces S_1 and S_2. This is done in the following manner: Determine the

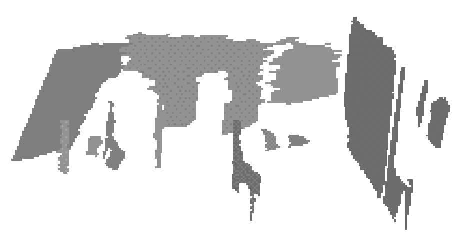

Figure 4: Polygonal representation of the segments.

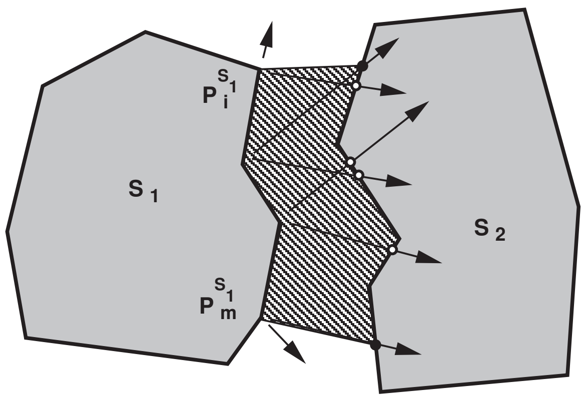

Figure 5: Neighbour points of S_1 with S_2.

extrema points (black points in figure 5) of the neighbour points on S_2 that correspond to S_1. Determine the segment of the boundary on S_2 that contains all the neighbour points. Extract from this segment the corner points, since all that is required to define the segment are the extrema points and the corner points within the segment. The gap is defined by joining this segment with another segment from the boundary of S_1 that contained the neighbour points that are within the segment from S_2, see figure 5.

There does not exist one unique gap between the surfaces, in fact at least 2 common gaps can be defined; One with respect to the first surface connecting it to another second surface and the reverse, one from the second surface back to the first. This is demonstrated in the gaps defined in figure 6 (a) and (b) where figure 6 (a) are the gaps defined with respect to surface S_1 and figure 6 (b) those with respect to surface S_2. This will result in 2 values of proximity as defined in section 3.

As well as these 2 possible cases multiple gaps can be defined with respect to one surface towards the other, as shown in figure 6 (a) as the hatched polygons. One gap is defined from the continuous set of points starting at $P_i^{S_1}$ to $P_m^{S_1}$ and another from $P_q^{S_1}$ to $P_u^{S_1}$. This split is a result of the direction of the line defined from $P_m^{S_1}$ to $P_q^{S_1}$ whose projection does not intersect with the S_2. In this situation one can sum up the effect of these 2 gaps when computing the proximity measure.

3 Computing a proximity measure

The objective is to define a compatibility function among the surfaces that quantifies the distance between the boundaries of 2 surfaces. Our approach is similar to the proximity approach used by Fan et al. [6] for determining the proximity between the end points of 2 lines. That approach compared the proximity between 2 lines by computing a ratio of the distance of the gap between the 2 closest end points of the lines to the sum of the lengths of the 2 lines. In this particular case the 3-D surfaces replace the lines and the distance between the 2 closest end points is replaced by a common polygon shaped gap. Instead of using a linear distance as a measurement of the proximity it is more appropriate to use the area of the surfaces and of the gap. This results in the following measure of proximity,

$$f_{prox}(S_i, S_j) = \frac{A_g^{\mathbf{s}_i\mathbf{s}_j}}{A_{\mathbf{s}_i} + A_{\mathbf{s}_j}}, \quad (1)$$

where $A_g^{\mathbf{s}_i\mathbf{s}_j}$ is the area of the gap between surfaces $\mathbf{s}_i$ and $\mathbf{s}_j$., and $A_{\mathbf{s}_i}$ and $A_{\mathbf{s}_j}$ are the area of the surfaces S_i and S_j. The closer the surfaces are to each other the smaller the value of $f_{prox}(S_i, S_j)$.

Up to this point all computations where performed utilizing the 2-D image representation of the 3-D sensory points. To get a proper estimate for the proximity measure between the surfaces it is necessary to compute the area of the 3-D surfaces. Note that the approach could be used on 2-D surfaces by simply computing the area of the 2-D polygon, in this manner this approach can easily be applied to surfaces in a 2-D image plane. The areas for the surfaces are computed using an algorithm for computing the area of a polygon. An estimate of the area of the gap is performed by approximating it as a triangular mesh [11] and summing the areas of the triangles in 3-D space. This triangular mesh approximation is necessary because the shape of the gap can be an irregular shaped surface and not planar. An example of this is shown in figure 7 (a) where the gap defined between surfaces S_{17} and S_{13} has been triangulated in the 2-D image. The result of this triangulation in 3-D is shown in figure 7 (b).

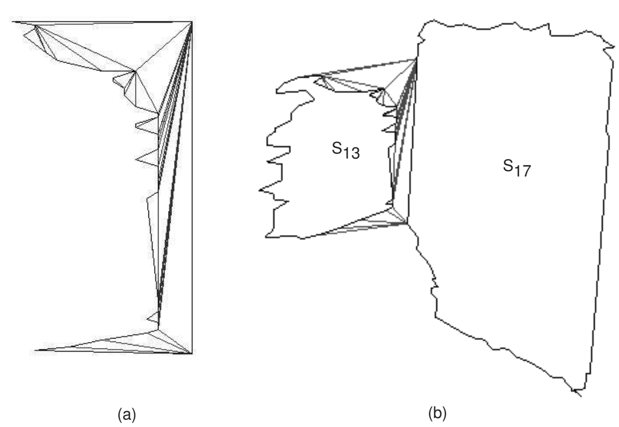

Figure 7: Triangulated gap between S_{17} and S_{13}.

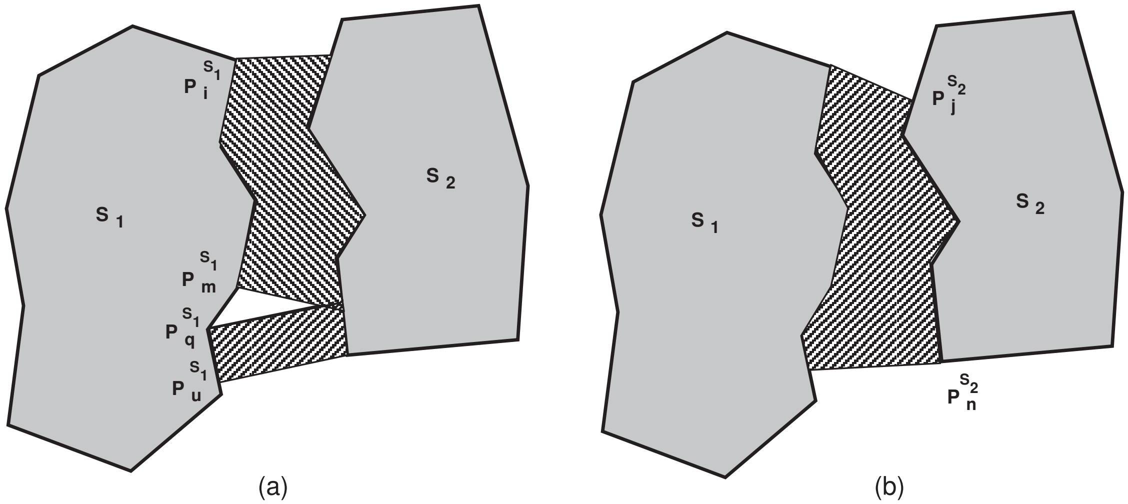

Figure 6: Proximity gaps between surface S_1 and surface S_2 (a) and between surface S_2 and surface S_1.

S_i	S_j	Pr.	S_i	S_j	Pr.	S_i	S_j	Pr.
1	6	0.23	9	12	0.23	13	10	0.09
1	9	0.04	10	17	0.03	13	5	0.01
4	17	0.12	10	9	0.03	14	8	0.31
4	5	0.39	10	12	0.15	14	11	3.47
5	4	0.24	10	15	0.52	15	18	0.85
5	17	0.23	10	5	0.03	15	10	0.55
5	10	0.05	10	13	0.10	15	12	1.46
6	12	11.3	11	14	2.29	16	11	1.10
6	9	0.00	11	8	13.5	16	17	0.01
6	1	0.27	11	16	6.5	17	16	0.00
8	11	13.4	12	15	1.51	17	10	0.01
8	14	0.30	12	10	0.51	17	13	0.05
9	1	0.02	12	9	0.60	18	15	0.86
9	6	0.17	12	6	3.44	18	17	0.48
9	12	0.00	13	17	0.03	18	10	0.48

Table 1: Proximity measures among adjacent surfaces.

4 Discussion

Table 1 presents calculated proximity values among the adjacent surfaces. Not all the surfaces have been included in the calculations, a number of surfaces below a certain area were eliminated at the beginning of the process. In particular surfaces S_2, S_3, and S_7 were all considered too small in size to process. Note, a small value for proximity implies that the surfaces are closer.

Adjacent surfaces will not have proximity values of 0 for several reasons. First, the process of defining a surface boundary results in the creation of a small gap between the surfaces. Even if the surfaces were originally connected there will always be a small gap created from the boundary extraction process. Secondly, irregular boundary shapes can result in gaps that extend beyond the length of the edges originally shared between the adjacent surfaces. For example, surfaces S_{10} and S_{13} in figure 1 (a) share a common edge. Small gaps will be detected at the top and bottom of the common edge due to the irregular shape of this edge. Thirdly, the common edge between the surfaces may be a jump edge, so the size of the gap can be rather large in depth. This is the situation depicted between the surfaces S_{12} and S_6 where they appear as joined surfaces in the image but are at different distances from the sensor.

At most 2 proximity measures may exist between adjacent surfaces and should be nearly of the same value. Table 1 (b) should be interpreted as the proximity measure between S_i and S_j relative to S_i, therefore another may exist between S_i and S_j relative to S_j. In some circumstances only one proximity measure will be available. This may occur when the polygon that defines the gap between the surfaces has self-intersecting edges. A unique algorithm was developed to automatically try to correct for this but will eventually fail if too many self-intersections are discovered. Self-intersection may seem theoretically impossible but in practice can occur when the boundaries are very close to each other.

The proximity measure computes one value to be used for determining if adjacent surfaces can be considered proximal to each other. The problem still exists of how one would choose a reasonable threshold value for proximity.

Equation 1 is a ratio of surface areas, those of the area of the gap to the sum of the areas of the adjacent surfaces. It is important to keep in mind that the gap is an approximation of a 3-D surface defined by propagating a surface's boundary, in a 2-D image, until it collides with another surface. Reasonable proximity values have more to do with the psychology of perceptual grouping and do not follow any hard rules. In general it appears that surfaces separated by gaps that are about the same size as themselves do not give the appearance of being proximal. If the 2 surfaces are about the same size then this results in a threshold value of 0.5, so that any ratio above 0.5 can be considered as non-proximal surfaces.

Instead of using the above threshold rule as a crisp value it is also possible to map the proximity values to a fuzzy set by using a declining S-curve [4]. This type of mapping was used by Liscano et al. [10] in the design of a Bayesian Network for the grouping of 3-D surfaces. The use of a Bayesian Network allowed the proximity value to be propagated and combined with other measures of evidence without having to make a decision on proximity at an early stage of the grouping operation.

5 Conclusions

This article presented an approach for computing a proximity compatibility function between 2 surfaces in 3-D space based on a ratio of the area of a common gap between the surfaces to the sum of the area of the surfaces. The approach taken was to determine adjacency between fragmented surfaces and compute a measure of proximity between 2 adjacent surfaces by computing a common gap between their boundaries. The difficulty is in estimating the shape of the gap between the 2 surfaces. The presented methodology computes this shape using the image plane acquired from a sensor view, thus reducing the complexity of the problem down to 2-D image analysis instead of 3-D geometrical analysis. After the shape of the gap is determined the actual surface areas are estimated in 3-D space and the gap is triangulated so that an estimate of its surface area can be performed.

The proximity measure was applied to 3-D sensory data acquired from an indoor environment that was segmented into planar surfaces. The approach could be applied to any other type of ordered 3-D sensory points, but it is better suited for data that contains fragmented surfaces. If the sensory data is dense (with relatively small gaps), then surface adjacency is easily determined and it is simpler to extend the surfaces until they join. The surfaces do not necessarily have to be planar, because the algorithm operates on the surface's boundary in the 2-D image plane and not on the actual 3-D surface.

References

[1] N. Ayache. *Artificial vision for mobile robots: stereo vision and multisensory perception.* (Cambridge: MIT Press, 1991).

[2] F. Blais, M. Rioux, and J. Domey. Optical range image acquisition for the navigation of a mobile robot. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, volume 3, pages 2574–2580, Sacramento, CA., April 9-11 1991.

[3] P. Boulanger. Hierarchical segmentation of range and color images based on bayesian decision theory. In *Proceedings of the Fifteenth International Workshop on Maximum Entropy and Bayesian Methods*, Santa-Fe, NM., July 31- August 4 1995.

[4] E. Cox. *The fuzzy systems handbook.* (New York: AP Professional, 1994).

[5] M. Devy, J. Colly, P. Grandjean, and T. Baron. Environment modelling from laser / camera multisensory system. In *The 2nd Workshop on Sensor Fusion and Environment Modelling*, Oxford, UK, September 2-5 1991. International Advanced Robotics Program.

[6] T.-J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-d objects using surface descriptions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(11), Nov. 1989, pages 1140–1157.

[7] H. Freeman, editor. *Machine vision for three-dimensional scenes.* (New York: Academic Press, 1990).

[8] Q.-G. Gao and A. K. C. Wong. Curve detection based on perseptual organization. *Pattern Recognition*, 26(7), 1993, pages 1039–1046.

[9] J. Hoshino, T. Uemura, and I. Masuda. Reconstruction of 3d scene using active and passive sensors. *IEICE Trans. on Communications, Electronics, Information, and Systems*, E74(10), Oct. 1991, pages 3444–3450.

[10] R. Liscano, S. Elgazzar, and A. K. C. Wong. Use of belief networks for modeling indoor environments. In *Proceedings of Vision Interface 97*, page Submitted for Review, Kelowna, Canada, May 19-23 1997.

[11] R. Seidel. A simple and fast randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Computational Geometry: Theory and Applications*, 1(1), July 1991, pages 51–64.

[12] V. Sequeira, J. G. M. Goncalves, and M. I. Ribeiro. 3d scene modelling from multiple views. In *Proceedings of the SPIE: Videometrics IV*, volume 2598, pages 114–127, Philadephia, Pa., October 25-26 1995.

[13] M. Suk and S. M. Bhandarkar. *Three-dimensional object recognition from range images.* (New York: Springer-Verlag, 1992).

[14] H. Yao, R. Podhorodeski, and Z. Dong. A cross-section based multiple-view range image fusion approach. In *Proceedings of the SPIE: Sensor Fusion and Aerospace Applications*, volume 1956, pages 212–223, 1993.

[15] Z. Zhang and O. D. Faugeras. *3D Dynamic Scene Analysis: A Stereo Based Approach.* (Berlin: Springer-Verlag, 1992).