Proceedings SPIE, Vol. 1965, April 1993

Neural net range image segmentation for object recognition

Leda Villalobos and Francis L. Merat

Electrical Engineering Department
Center for Automation and Intelligent Systems Research
Case Western Reserve University, Cleveland, OH 44106-7122

ABSTRACT

A technique for performing surface-based segmentation of range images using neural nets is introduced. In this approach, multilayered neural nets are used to classify local image patches according to the type of surface they belong to, based on features extracted from range and surface normal information. Central component to the efficiency and robustness is a near orientational invariant local data organization which takes place before features are extracted. This data organization reduces internal complexity by shifting the orientation invariance burden from the dimensionality of the feature spaces and/or from the internal architecture of the networks, to a much simpler sequencing of local data. The result is a well segmented image in which surfaces are properly labeled and delimited, without over segmentation. The approach shows to be robust in front of noise.

1. INTRODUCTION

A computer vision system attempts to construct explicit, symbolic descriptions of the objects appearing in an image. Since images are huge arrays of data, the process of building a symbolic description involves a series of procedures, each one of them progressively giving a more abstract and meaningful representation to the data. One of the first steps towards symbolic description is segmentation, whose objective is to identify image regions sharing similar properties of interest. Accurate image segmentation is a fundamental step in the development of a vision system, as the quality of the segmentation greatly affects the ultimate performance of the subsequent procedures.

Segmentation similarity criteria are numerous and usually tailored to the application at hand. In outdoor image understanding, where it is important to localize vegetation, roads, and the like, a common segmentation criterion is similarity in color and texture.[1,2] In images of assembly workcells, where the final goal is the identification and registration of objects, a criterion could be the similarity in parametric surface description, since surfaces normally play an important role in model-based object recognition.[3] In this paper, we focus our attention on surface segmentation of range images.

Two elements characterize a surface, geometry and shape. The geometry of a surface visible in a scene refers to its boundary, while the shape refers to the actual model of the surface. For instance, in a cube all surfaces have rectangular geometry and planar shape. Normally, surface segmentation requires the geometry to be identified; depending on the object recognition paradigm, shape estimation may or may not be necessary.

The boundary is the linked collection of the edges circumscribing a surface. In an image, an edge is composed by the pixels at which two different surfaces come into contact, or at which a surface and the background interact. Edges can be identified either with edge detection operators or through region growing techniques.

Intuitively, the regions on either side of an edge have dissimilar characteristics. The purpose of an edge detection operator is to highlight and detect these differences acting upon small neighborhoods of pixels.[4] For example, edges can be detected by identifying those pixels whose gradient magnitudes are larger than a threshold. Although significant work has been done in this area, edge detection continues to be a difficult task. It is typical to get missed pixels in an edge, which causes the boundary to break.

The idea behind region growing is that surfaces could be identified by grouping together spatially closed pixels that share similar intrinsic characteristics. Hence, if the different surfaces are identified, the pixels at which they interact -the edges- can easily be detected. Multiple region growing algorithms have been reported in the literature. In some of them, local invariant characteristics, such as Gaussian or mean curvatures, are first obtained and adjacent pixels with similar curvatures are grouped together and labeled as being part of the same surface.[3,5] In others, similarity in surface normals is used as the criterion guiding pixel grouping.[6,7]

Inherently, boundary detection by region growing is more robust than edge detection because surfaces involve more pixels than edges. Nevertheless, these techniques present several problems. Calculation of second order derivatives prompts curvature estimates to be highly sensitive to noise and to the image discretization process itself.[8,9] Rigid thresholds to define similarity among surface normals make delimitation of non-planar surfaces very difficult. Consequently, the result in these cases could be an overly segmented image.

In this research, an adaptive neural net technique to perform range data segmentation is introduced which overcomes the problems associated with curvature estimates and rigid similarity thresholds. In this approach, neural nets are trained to recognize trend characteristics of the range and surface normal information, and to identify different surfaces types such as planar, spherical, cylindrical, etc. The result is a well segmented image in which surfaces are properly labeled and delimited, without over segmentation. The approach is computationally efficient and does not rely on the tuning of parameters. Central component to the efficiency and robustness is a near orientational invariant local data organization which takes place before features are extracted and presented to the neural nets. This data organization reduces internal complexity by shifting the burden of orientation invariance from the dimensionality of the feature spaces and/or the internal architecture of the networks, to a much simpler sequencing of local data.

The remainder of this paper is organized as follows. In section 2, we present a very brief block diagram description of the system's organization. Detailed explanations of the individual procedures are given in section 3. Results and discussion appear in section 4, with conclusions and recommendations for future work in section 5.

2. SYSTEM DESCRIPTION

A block diagram of the segmentation system appears in Figure 1.

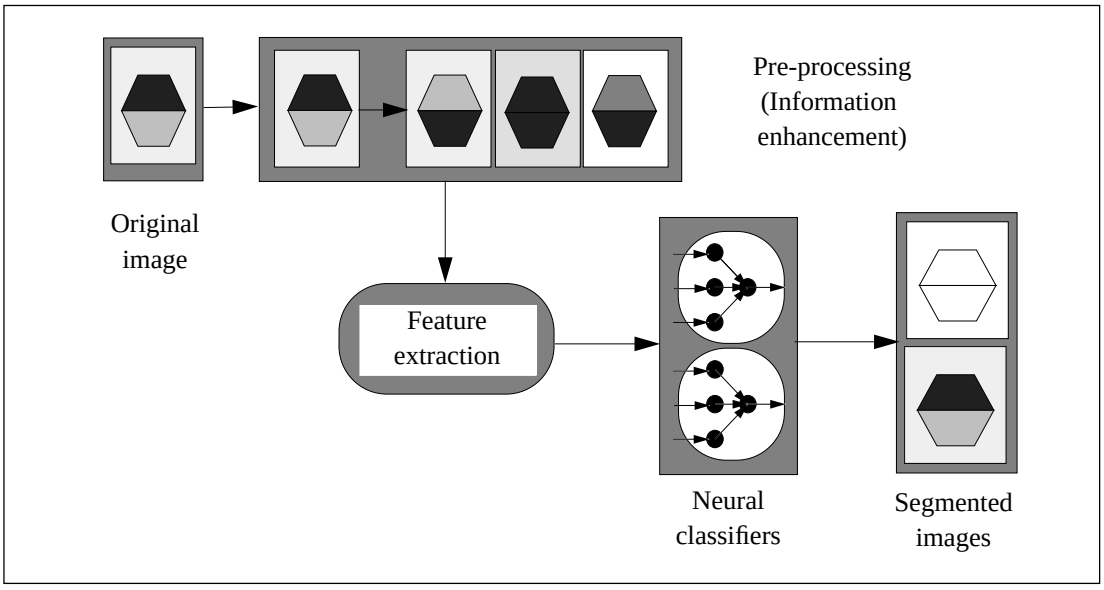

FIGURE 1. Neural segmentation scheme.

Initially, the original range image undergoes a pre-processing procedure which includes noise filtering and surface normal estimation at every pixel. The magnitude and direction cosines of the normals are then extracted to create four new images. After this, small surface patches are defined, and the local data of every patch is organized in a near orientational invariant fashion. Feature vectors are extracted from the reorganized data which are then presented to a pair of feedforward multilayered neural nets. The output of the first net indicates whether or not the feature vector was taken from a patch corresponding to an edge, while the output of the second net indicates the type of surface the patch belong to. The result is an edge and surface based segmentation of the range image.

3. IMPLEMENTATION

3.1. Range image

A range image is a function *f(u,v)*, where *u* and *v* are the row and column in the image, respectively, and *f(u,v)* is the range associated with the pixel *(u,v)*. The triple *(u,v,f(u,v))* can be converted into a world coordinate system with an appropriate transformation function which depends on the ranging device used to collect the image.

The research reported in this paper is restricted to the segmentation of range images obtained, for example, with a laser range finder. With this kind of sensor, the scene is scanned with a laser beam in incremental angular steps; *u* and *v* are related with the vertical and horizontal angular deflections of the beam, respectively. For any pixel *(u,v)*, *f(u,v)* represents the distance between the sensor and the point in the scene on which the beam impinges. Suppose that a Cartesian coordinate system is constructed at the sensor with the X and Y axes pointing in the horizontal and vertical directions, and Z pointing towards the scene. In such a case, the Cartesian coordinates for *(u,v,f(u,v))* will be given by *(g(u,v,f(u,v)),h(u,v,f(u,v)),f(u,v))*, where *g* and *h* are transformation functions derived from the sensor's geometric parameters.

3.2. Surface normal estimation

Consider a surface parametrically described by

$$z = f(u, v) \tag{1}$$

where *z* denotes range, and *u* and *v* are two indexing variables. The normal to this surface at a point $P_0=(u_0,v_0)$ is given by

$$\underline{\boldsymbol{n}} = f_u(u, v)\big|_{P_0}\hat{\boldsymbol{u}} + f_v(u, v)\big|_{P_0}\hat{\boldsymbol{v}} - \hat{\boldsymbol{k}} \tag{2}$$

To calculate the surface normal at a pixel in a range image, the partial derivatives f_u and f_v of (2) have to be estimated based on the variation of range in the neighborhood of the pixel of interest. A number of techniques have been proposed to approximate the partial derivatives. In some of these techniques, the derivatives are estimated through weighted discrete differences encoded in templates, simple examples being the Kirsh and the Sobel operators.[10] Templates are attractive because they offer the advantage of very fast computation. However, the discrete differences act as high-pass frequency filters over the image, producing a sensible loss of information. Other techniques fit a plane to the data surrounding the pixel under consideration, and calculate the derivatives for the plane's equation.[7] This procedure has the contrary effect, acting as a low-pass filter over the image. If the surface curvature is large, the normal's estimate degrade.

Instead of using templates or planar approximations, we explore a different approach in this research. To estimate the normal at the pixel *(u,v)*, a small patch centered at *(u,v)* is defined and the Cartesian world coordinates for every pixel in the patch are calculated. The result is a set $Q=\{(X_i,Y_i,Z_i) \mid X_i=g(u_i,v_i,f(u_i,v_i)),\ Y_i=h(u_i,v_i,f(u_i,v_i)),\ Z_i=f(u_i,v_i)\}$. Fourth-order Lagrange polynomials[11] are then fit to the data along the *u* and *v* directions, as functions of

X and Y respectively. Partial derivatives are calculated from the profile polynomials to obtain f_x and f_y, and the surface normal is estimated according to (2).

One of the highly desirable characteristics of Lagrange polynomials is the exact approximation of the function at the sampling points. Consequently, over a moderately smooth surface the approximation will be more faithful than planar approximations or template matching because low and high frequency variations can be accurately modelled. Figure 2 shows a few representative results.

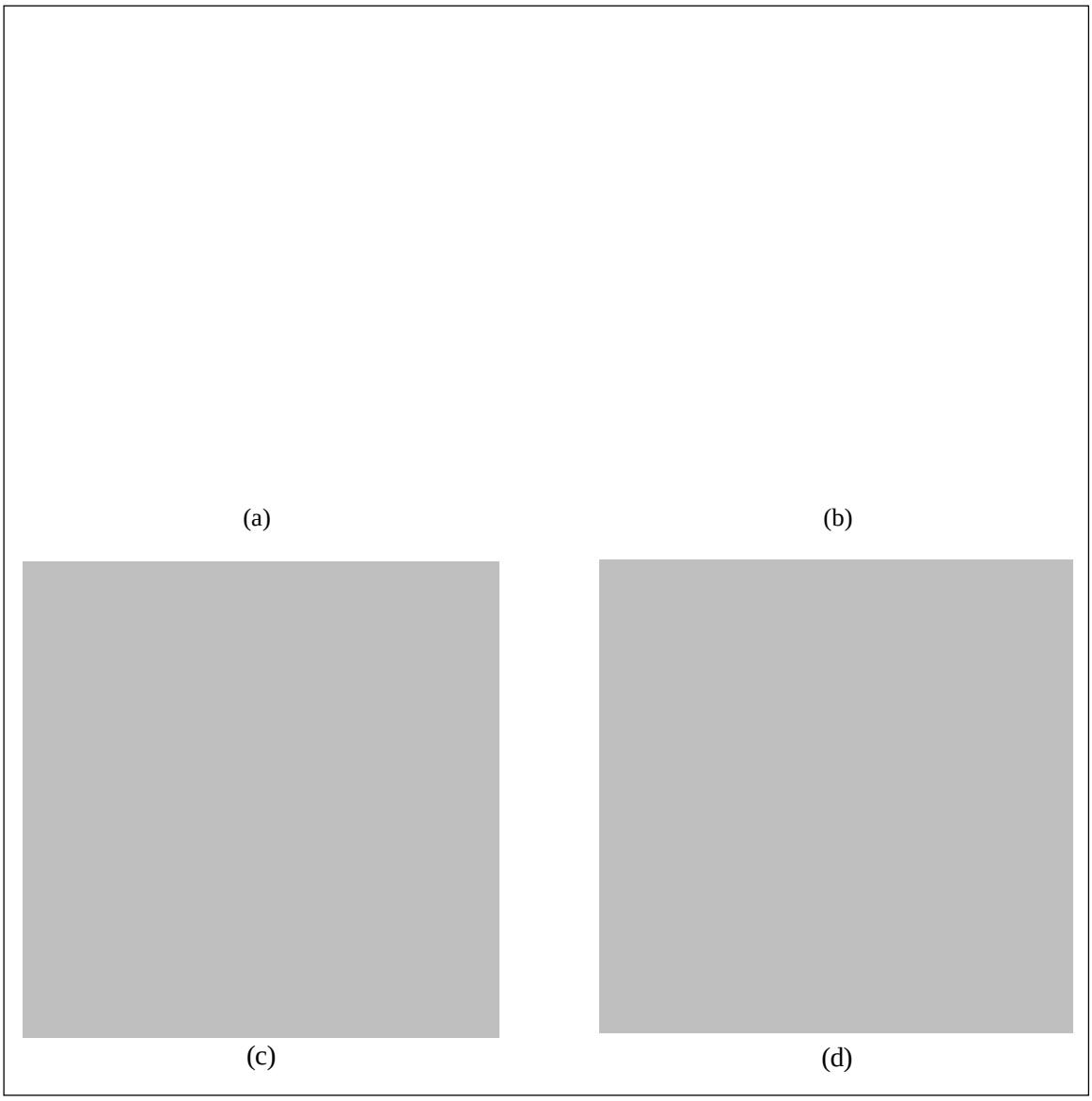

FIGURE 2. Surface normal estimates. In each group, the upper left image corresponds to the original range image. The normals direction cosines along the X, Y, and Z directions appear at the upper right, bottom left, and bottom right, respectively. No estimates are calculated in the regions surrounding depth discontinuities. Normal estimates for images of (a) a block with a blind hole, and (b) a block with a boss were obtained with the proposed technique. Note how the normals of cylindrical and planar surfaces are well defined. For comparison purposes, the corresponding estimates obtained using templates appear in (c) and (d). The filtering effects of the template make the images dull.

3.3. Near orientation invariant data organization

The most important issue of surface based segmentation is the definition of a robust mapping function from the image data to the discrete set of possible surface shapes. From differential geometry, it is known that surfaces with the same shape have the same mean and Gaussian curvatures, independent of viewpoint under orthographic projection.[12] Consequently, mean and Gaussian curvatures are two strong candidate features for defining the mapping function. There has been substantial research on the segmentation and characterization of range surfaces using curvatures.[3,13] However, the estimation of curvature requires the calculation of second order derivatives, a task that is very sensitive to the high frequency noise introduced by the image discretization process.

Figure 3 presents an example of mean and Gaussian curvature estimation for a cube and a boss. It can be seen that differentiating between the planar and cylindrical surfaces is difficult, since the difference in curvatures between both surfaces is small. This low contrast makes necessary a fine tuning of the thresholding processes that classifies surfaces with different local shapes.[3]

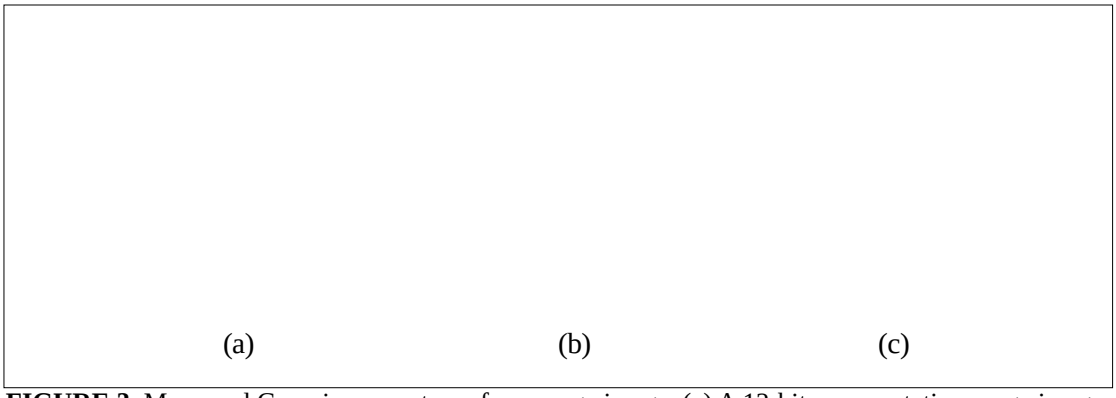

FIGURE 3. Mean and Gaussian curvatures for a range image. (a) A 12-bit representation range image; (b) Corresponding mean curvature, and (c) Gaussian curvature images. Mean and Gaussian curvatures were obtained using the procedure outlined by Besl.[3] Note how difficult it is to differentiate between surface types.

The concept behind surface curvature is the directional rate of change of the normals. The variation of the normals over a patch of surface is indeed a signature of the surface's shape. Since the normals are expressed with respect to a fixed reference frame and the surfaces can have any orientation with respect to that frame, the signature is orientation dependent. Given that no restrictions are imposed on the surface's orientation, a mechanism to deal with orientation dependency is necessary. In this research, we rely on neural nets to generate the orientation invariant mapping function.

Orientation invariance is a particular type of transformation invariance problem. Transformation invariance with neural nets can be attained in four different ways: by architecture invariance, by exhaustive training, by feature invariance, and by feature constraint.[14] Invariance by architecture is obtained by imposing a structure on the connectivity of the net which forces the same output to occur whenever the input corresponds to a transformation of the same data. Extensive work in this area has been done by van der Malsburg[15] and Fukushima.[16] Conceptually, this is a straight forward approach to invariance inspired on the mammalian visual path. Nevertheless, it has the serious disadvantage of poor scaling. Invariance by training is obtained by presenting the net with a substantial number of training patterns which virtually cover all possible transformations of the data.[17] Eventually, exhaustive training becomes computationally impractical. Invariance by feature definition is obtained by selecting feature spaces which are themselves invariant to the transformations. This is a very efficient approach, but requires robust, invariant features.[18] The concept of invariance by feature constraint has been recently introduced by Barnard.[14] In this approach, raw features are extracted from the data and a family of constraints is imposed upon them to generate new features. These new features are invariant to the transformations.

We deal with orientation invariance in a way slightly different from that proposed by Barnard.[14] We attempt to introduce invariance directly at the local data organization level, such that the directions at which the normals are examined in a surface patch remain near orientation invariant.

Suppose the surface type of a small neighborhood of pixels has to be estimated. To organize the local data, the gradient, its perpendicular, and the normal vector are used to construct a reference frame at the center of the patch. This construction represents a compromise. Strictly speaking, an invariant frame would have to be constructed along the directions of minimum and maximum curvature, but the computational cost of locating them is high. Furthermore, as it was mentioned before, the second derivatives involved in the process would prompt noise sensitivity. After defining the frame, the angle θ between the gradient and the horizontal is used to index an organization template. Local data is reorganized by applying this template to the patch, as shown in Figure 4.

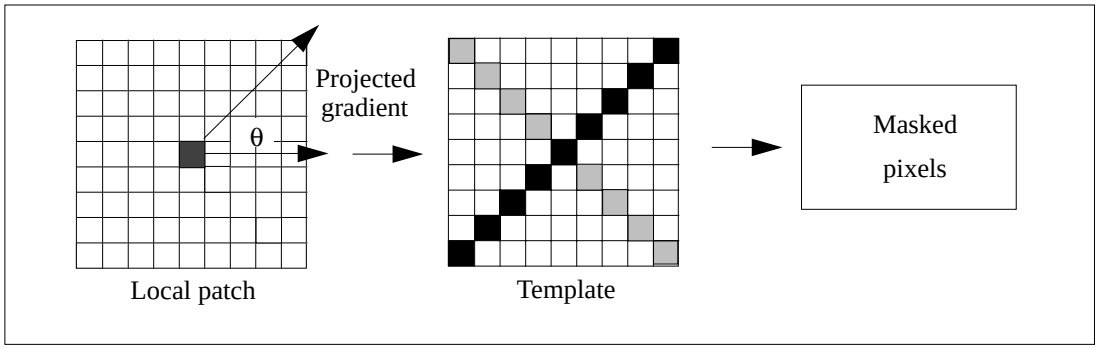

FIGURE 4. Local data organization process.

For data organization purposes, a template has four branches, B1-B4, 45º apart from each other. Pixels are selected in order, starting with those selected by B1 and ending with those by with B4. The center pixel is considered to be part of all four branches. Two criteria control the definition of the templates: first, the same number of pixels has to be selected on each one of its four branches with respect to the center pixel; and second, the selected pixels should lay as faithfully as possible along the desired directions despite image discretization. The templates corresponding to angles θ falling in the first quadrant are shown in Figure 5. Only the templates for angles falling in the first quadrant need to be pre-calculated. For the remaining quadrants the only variation corresponds to the labeling of every branch, as shown in Figure 6.

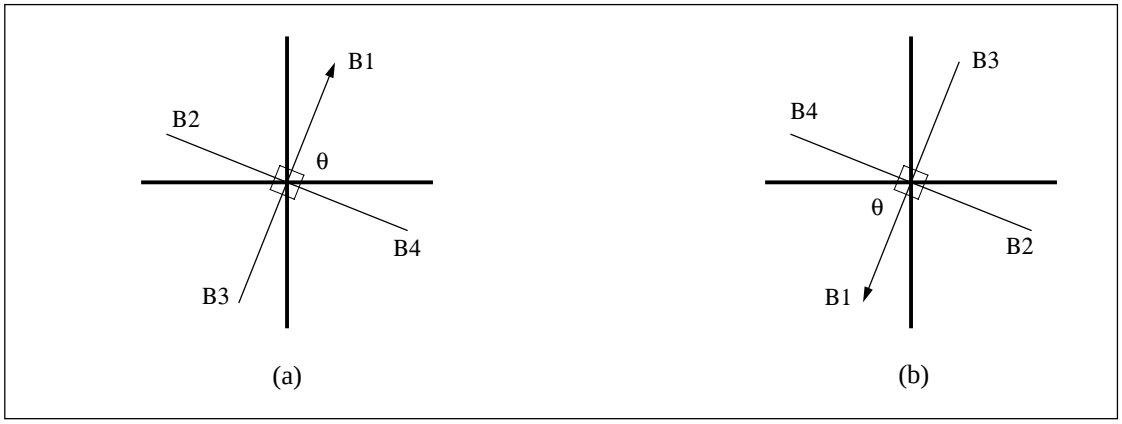

FIGURE 6. Branches for an angle falling in (a) the first quadrant, and (b) the fourth quadrant. Branches remain in the same directions but their labeling differs.

It should be emphasized that, since our preliminary experiments were conducted with a reduced number of different surface types, the pattern of the templates is fairly simple. For more complex surfaces, these patterns would have to include additional number of pixels.

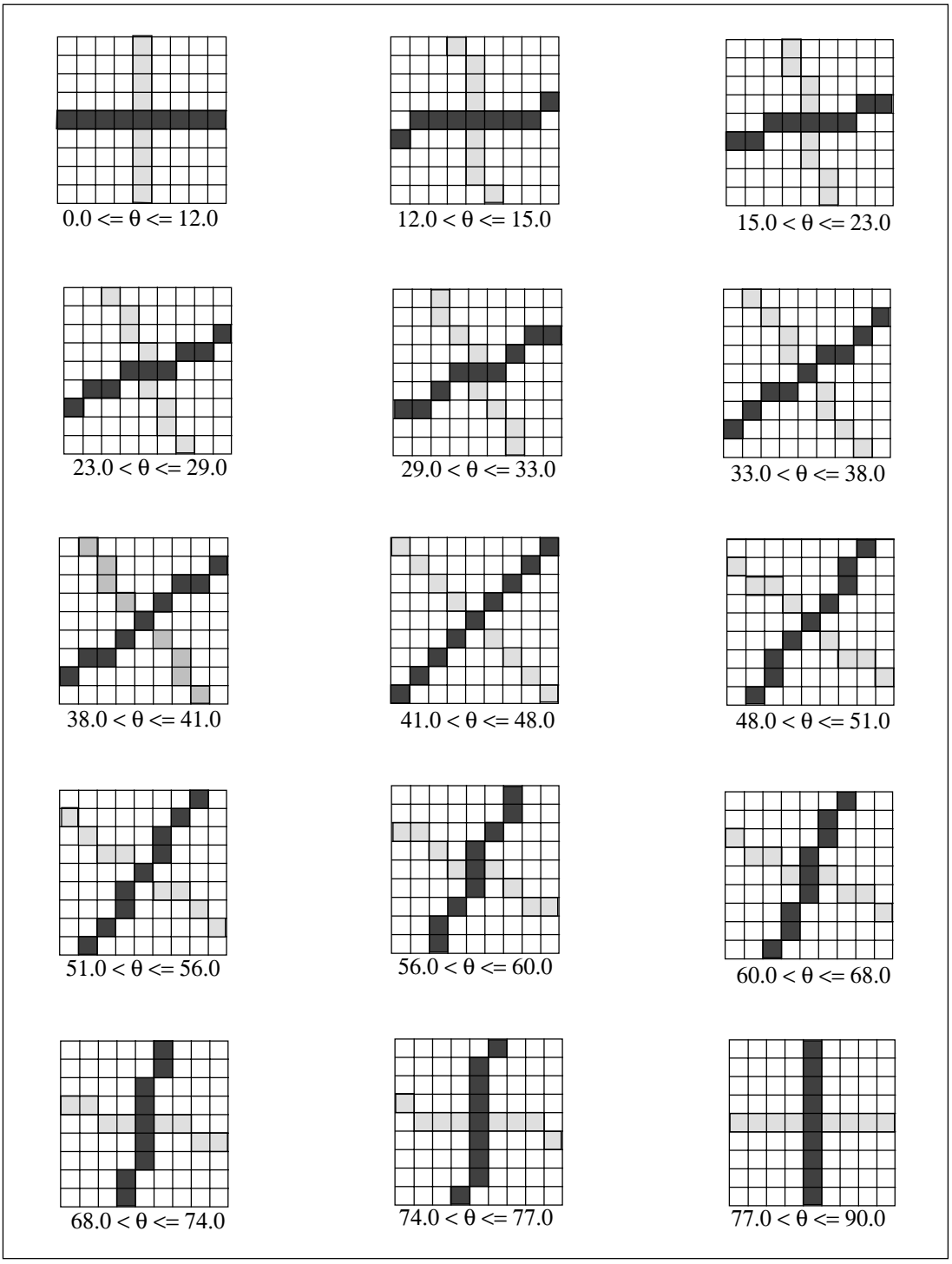

FIGURE 5. First quadrant templates.

3.4. Feature extraction and neural net training

We have trained neural nets for edge recognition and for surface type labeling. The 1200 training patterns were extracted from several synthetic images of planar, cylindrical, and spherical surfaces. Unlike real data, these images did not include any noise, i.e. were exact in 3-D spatial coordinates.

For edge detection, the features emphasize changes in normal directions and range. They include the variance of each direction cosine, the variance in range, and the rate of change of the direction cosines per unit of length. For surface type classification, the features emphasize the trends of range and normals. They include the variances along each branch for the magnitude of the normal and the direction cosines; the variances along each branch for the range; and the rate of change of the normal magnitude and direction cosines magnitude per unit of length. Every feature was linearly normalized; the maximum allowed value was 0.9 and the minimum 0.1.

4. RESULTS

We have tested the neural nets with a number of synthetic range images. The images are 256x256 pixels, with 12 bit representation, and have been corrupted with an additive gaussian noise of 4 gray levels; they simulate a range finder with a maximum valid range of 1.0 m., and a 60^ox60^o field of view. Two representative segmentation examples are shown in Figure 7.

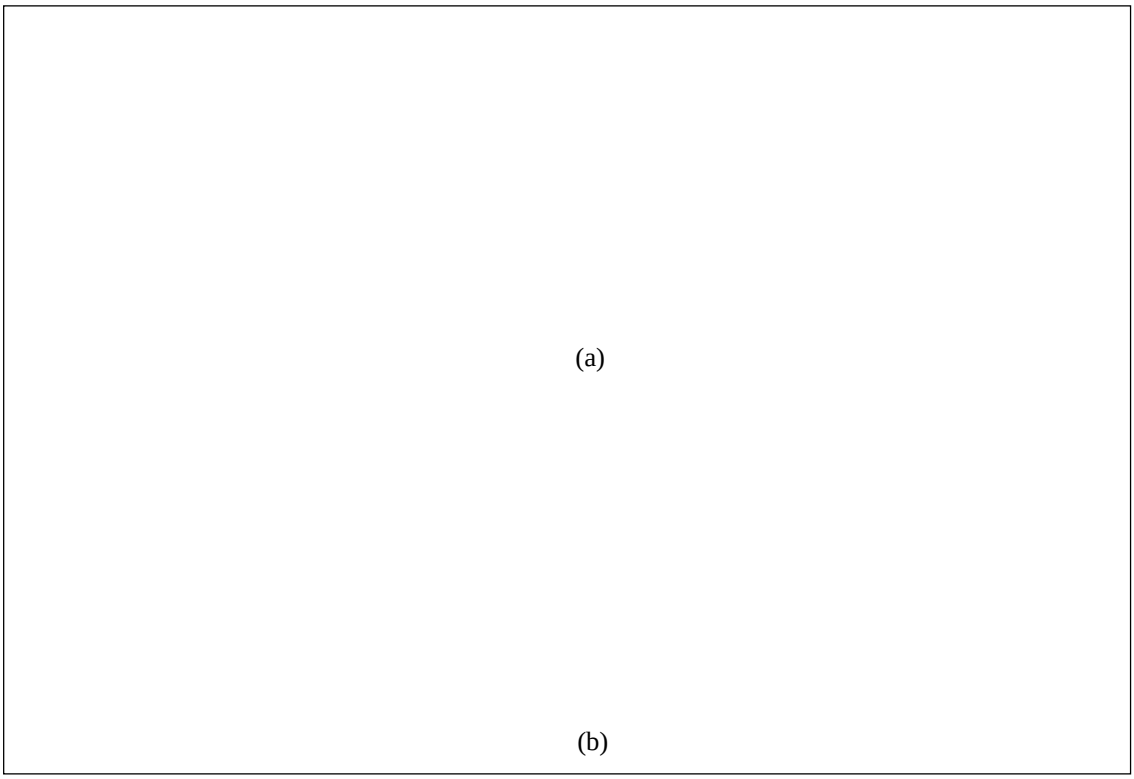

FIGURE 7. Segmentation results. In each set, the left, middle and right images correspond to the range, neural edge segmented, and neural surface segmented images, respectively. (a) Segmentation of a block with a through slot; (b) segmentation of a block with a boss.

Note how all the edges present in the image, including creases, are properly detected by the neural net. The surface types are also detected. It is interesting to see that in the surface segmentation, the crease edges are normally classified as being part of a cylindrical surface. This occurs because of the smoothing effects introduced by the Lagrange polynomials at and near the edges. These smoothing effects make the region look like a cylinder with a small radius.

In an object recognition system, these segmentation images would have to be filtered to suppress undesirable artifacts introduced for example by smoothing. The procedure would include a thinning of the edge image, and a dilation of the surface image.

5. CONCLUSIONS

We have introduced a connectionist technique for segmenting range images. The technique consists of four major steps: 1) image filtering and estimation of surface normals; 2) estimation of the maximum gradient at every pixel and near orientation invariant organization of the image data; 3) feature extraction; and 4) neural network surface classification.

To estimate the surface normal at a pixel, fourth order Lagrange polynomials are fit to the Cartesian range data surrounding the pixel. These polynomials are functions of the absolute coordinates (X,Y,Z). Partial derivatives with respect to X and Y are calculated, and their cross product used to obtain the magnitude and direction cosines of the normal.

Once the normals are estimated, the direction of the gradient at each pixel is used to reorient the local image data and create a near rotational invariant representation for each pixel's local region. This reorientation is done by applying appropriate templates for pixel selection along four branches of interest that are 45^{o} apart from each other. The templates are generated keeping two criteria in mind: first, the total number of pixels in every branch has to be independent of the gradient; and second, the selected pixels should lie as faithfully as possible along the desired branch directions despite image discretization.

After the appropriate pixels have been selected, feature vectors are extracted and presented to the neural nets for classification. The selected features emphasize surface and normal trends, and are good surface discriminants. Training has been conducted by extracting representative patterns from several planar, cylindrical concave, cylindrical convex, and spherical surfaces, and presenting them to the networks in a supervised training fashion.

In the final step, the outputs of the networks are used to identify the different surface regions, the surface types, and the edges existing at the boundaries of the surfaces. This approach has been tested with multiple range images. The results obtained with noisy, computer generated range images, show good performance with the edges and different surfaces being properly identified and labeled.

As part of future research, we plan on expanding the list of surface types the nets can recognize. We anticipate this expansion will require the modification of the current templates to include complete local regions instead of simple line branches.

6. ACKNOWLEDGMENTS

This research was supported with a graduate fellowship from the Center for Automation and Intelligent Systems Research. The center's support is greatly appreciated.

7. REFERENCES

1. C. Bouman, and B. Liu, "Multiple Resolution Segmentation of Textured Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 2, pp. 99-113, 1991.
2. M. Celenk, "A Color Clustering Technique for Image Segmentation," *Computer Vision, Graphics, and Image Processing*, Vol. 52, No. 2, pp. 145-170, 1990.
3. P. Besl, and R.C. Jain, "Segmentation Through Variable-Order Surface Fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 2, pp. 167-192, 1988.

4. E. Davis, “A Survey of Edge Detection Techniques,” *Computer Graphics and Image Processing*, Vol. 4, pp. 248-270.
5. T.-J. Fan, G. Medioni, and R. Nevatia, “Segmented Descriptions of 3-D Surfaces,” *IEEE Journal on Robotics and Automation*, Vol. RA-3, No. 6, pp. 527-538, 1987.
6. B. Sabata, F. Arman, and J.K. Aggarwal, “Segmentation of 3-D Range Images Using Pyramidal Data Structures,” *Proceedings of the International Conference in Computer Vision*, Osaka, pp. 662-666, 1990.
7. R. Hoffman, and A.K. Jain, “Segmentation and Classification of Range Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 5, pp. 608-620, 1987.
8. P. Flynn, and A.K. Jain, “On Reliable Curvature Estimation,” *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pp. 110-116, 1989.
9. E. Trucco, “On Shape-Preserving Boundary Conditions for Diffusion Smoothing,” *Proc. 1992 IEEE International Conference on Robotics and Automation*, Nice, pp. 1690-1693, 1992.
10. R. Haralick, and L.G. Shapiro, Computer and Robot Vision, Vol. 1, Addison-Wesley, Reading, Massachusetts, 1992.
11. R.L. Burden, and J.D. Faires, Numerical Analysis, Prindle, Weber, and Schmidt Publishers, Boston, 1985.
12. W.L. Burke, Applied Differential Geometry, Cambridge University Press, New York, 1985.
13. S.S. Sinha, and P. Besl, “Principal Patches: A Viewpoint-Invariant Surface Description,” *Proceedings 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, pp. 225-231, 1990.
14. E.Barnard, and D. Casasent, “Invariance and Neural Networks,” *IEEE Transactions on Neural Networks*, Vol. 2, No. 5, pp. 498-508, 1991.
15. C. von der Malsburg, “Pattern Recognition by Labeled Graph Matching,” *Neural Networks*, Vol. 1, No. 2, pp. 141-148, 1988.
16. K. Fukushima, “Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition,” *Neural Networks*, Vol. 1, No. 2, pp. 119-130, 1988.
17. D.E. Rumelhart, G.E. Hinton, and RJ. Williams, “Learning internal representations by error propagation,” in Parallel Distributed Processing, Chapter 8, pp. 318-362, MIT Press, Cambridge, 1986.
18. S. Perantonis, and P.J.G. Lisboa, “Translation, Rotation, and Scale Invariant Pattern Recognition by High-Order Neural Networks and Moment Classifiers,” *IEEE Transactions on Neural Networks*, Vol. 3, No. 2, pp. 241-251, 1992.