A Genetic Algorithm to Segment Range Image by Edge Detection

I. DIAZ[1], J. BRANCH[2], and P. BOULANGER[3]

[1]Fac. de Ingeniería. Universidad de Medellín,
Medellín, Colombia
email : idiaz@udem.edu.co
[2] Escuela de Sistemas. Universidad Nacional de Colombia Sede Medellín, Medellín Colombia, email:
jwbranch@unalmed.edu.co
[3] Department of Computing Science. University of Alberta, Edmonton, Alberta, Canada,, email:
pierreb@cs.ualberta.ca

Abstract--**The following article presents a segmentation method of range images. This method is based on edge map detection by calculations of depth gradients and orientation gradients and a genetic algorithm. The objective is to delimit the planar patches contained in images to facilitate the labelling of each region. The genetic algorithm is guided by depth gradients and orientation gradients in order to find the edge map of the image.**

I. Introduction

The problem of computational reconstruction from range data has been receiving much attention in computer vision over the last decade. The interest arises from the multiple applications of three-dimensional vision, such as motion analysis, manufacturing systems, navigational robotics, geographic information systems, model construction, etc. To perform the reconstruction task, the computer vision system must be able to identify real world objects from sensed environmental data in digital form. Due to the large amount of data, direct interpretation of a range image is extremely costly in terms of both storage and computer time. Therefore, a segmentation step is usually carried out to group the range data with particular properties in regions suitable for the subsequent image analysis and interpretation. The segmentation task should preserve the object shapes and edge locations. However, processing digital images to segment a number of objects in different positions and with different sizes and shapes is often perturbed by noise and other sensor errors.

The segmentation algorithms known in the literature can be classified into two categories: edge-based and region-based segmentation. Region-based approaches, group pixels into connected regions based on similarity between data points. On the other hand, edge-based methods locate boundaries between regions. Both techniques have their strengths and drawbacks. Edge detection techniques often produce no connected boundaries, false edge points, and thick edges. Despite the guarantee of closed regions, region-based techniques, such as region growing and clustering, have several critical design issues to be dealt with. The performance of most region-growing approaches crucially depends on the selection of initial regions. In clustering-based methods it is difficult to adaptively determine the actual number of clusters in range images. Often, an over segmentation is achieved and a subsequent merge step is needed to provide the final segmentation. Also, the region boundaries are generally not well-defined. [5].

Performance of segmentation methods can be evaluated using Hoover's methodology. This methodology compares machine generated segmentation against a manually specified ground truth. Comparison is based on five metrics: over-segmentation or multiple detections of a single surface, under-segmentation; or insufficient separation of multiple surfaces, missed, noise, and correct detection. In [3], [6], and [8], the results of some popular segmentation methods are presented and evaluated using Hoover's methodology [3]. The results illustrate that the range image segmentation cannot be regarded as solved, even for simple surfaces. Most segmentation methods perform poorly when the required tolerance is 80% or higher.

In this article we propose a segmentation method of planar surfaces based on edge detection. This method uses a genetic algorithm to find the edge map that delimits the objects in the image. Then regions in the image are labelled. The task performed for the genetic algorithm is the elimination of false edges and the enclosure of non–continuous edges from an initial edge map obtained from the calculation of direction gradients and depth gradients of the image.

II. Related Work

A variety of methods is available for range image segmentation. Most methods can be classified into region–based or edge detection categories. Hoover [3] presents some range segmentations. The USF range segmentation algorithm works by computing a planar fit for each pixel and then growing regions whose pixels have similar plane equations. The WSU range segmentation algorithm traces its origin to the dissertation work of Hoffman, but contains many enhancements incorporated by Flyn. This method uses a squared error clustering algorithm called CLUSTER to partition the feature space into clusters. The UB range segmentation algorithm considers the segmentation of range images into planar regions by growing regions. This segmenter is based on the fact that the points on the scan line that belong to a planar surface form a straight 3–D line segment. Therefore, the method divides each scan line into straight line segments and subsequently performs a region growing process using the set of lines.

0-7803-9419-4/05/$20.00 ©2005 IEEE

The University of Edinburgh (UE) range segmentation algorithm calculates a Gaussian mean curvature at each pixel to label it as belonging to a particular surface and then a region growing procedure is applied through an iterative expand, refit, contract cycle, grouping pixels with similar labels.

Another approach to range image segmentation is Segmentation Through Variable-Order Surface Fitting by Besl and Jain [1]. This algorithm segments a large class of images into regions of arbitrary shape and approximates image data with bivariate functions so that it is possible to compute a complete, noiseless image reconstruction based on extracted functions and regions. Surface curvature sign labelling provides an initial coarse image segmentation, which is refined by an iterative region growing method based on variable-order surface fitting.

The segmentation algorithm proposed by Sappa [13] and the segmentation algorithm proposed by Silva [10], can be classified into edge detection categories. Sappa's technique is based on scan line processing. It consists of two stages. First, a binary edge map is generated by means of a scan line approximation technique. Then, the points of that map are linked, generating the boundaries that enclosed each region. Silva's algorithm does edge detection by locating depth and orientation gradients in the image. Then a region growing method is applied to segment the image. The main problem with this last approach is that the locating gradients cannot guarantee closed boundaries, because some edge points are not correctly detected.

III. Proposed Segmentation Method

The proposed segmentation method is based on edge detection. The purpose is to find a well-defined edge map to lead the image segmentation. The segmentation method considers three stages. First, the depth gradients and orientation gradients are calculated. Then, in the second stage a genetic algorithm is used to find an edge map from the obtained gradients. Finally, a fast labelling is carried out in the third stage. A brief introduction of each stage is presented.

In the first stage, to extract depth and orientation gradients, we use the procedure propose in [9]. The depth gradients are equivalent to significant changes between the values of depth of the pixels in the image. The changes are identified by thresholding the maximum variation in *z* between the central pixel and each one of the other pixels in an *N x M* neighborhood. If the largest z-deviation is equal or greater than a threshold, the pixel is considered as a depth gradient. On the other hand, the orientation gradients are equivalent to significant change between the orientations of two surfaces. A two-step process is used to compute the orientation gradients. First, surface normals are estimated at each range pixel bounded by a *K x K* window. Second, the center pixel in an *S x S* window is considered an orientation gradient if a significant change between the normal's pixel and other normals in the window is within a threshold.

In the second stage, an edge map is obtained by a genetic algorithm. The task performed by the genetic algorithm is to select gradients to form correct, continuous, and thin edges. This procedure works with a population of individuals conformed by two-dimensional binary arrays where a value of one (1) represents an edge pixel while a zero (0) represents a non-edge pixel. The individuals are transformed with a traditional single point crossover operator and a particular mutation operator in each generation. The mutation operator has the task of selecting gradient points to conform to the edges of the image. This operator selects a starting point which is united with other pixels to form an edge. The selected pixel can be an isolated endpoint edge pixel, or a non-isolated endpoint edge pixel, or an orientation gradient. The selected pixel is united with other pixels until the conformed edge is closed.

The fitness function to evaluate each possible solution is based on three criteria to increase the number of edge pixels located correctly in the image, and to reduce the number of small regions and the number of endpoint edge pixels in the image.

Finally, in the third stage, a fast labelling is carried out assigning the same label to the pixels bounded with the same edges.

A. Calculation of gradients

Our segmentation method extracts an edge map from an initial approximation of edges obtained by calculating orientation and depth gradients. Due to noise present in the worked range image, a 9 x 9 median filter is applied to clean the image. Then the following procedures are performed as in [9].

Depth gradients. For each pixel in the image the respective coordinates (*x, y, z*) are obtained, using:

$$\begin{aligned} x[i,j] &= (j-255)*(r[i,j]/scal + offset)/|f_k| \\ y[i,j] &= (255-i)/c*(r[i,j]/scal + offset)/|f_k| \quad (1) \\ z[i,j] &= (255 - r[i,j])/scal \end{aligned}$$

where scal, offset f_k and c, are parameters obtained through a calibration camera and depends on the used image. In this work we used the 40 ABW range images of the USF [11].

For each coordinate (x, y, z) in the image, a depth difference is calculated by:

$$D_p(i,j) = \max|w_{i,j} - W| \quad (2)$$

where $w_{i,j}$ is the *z* value of the point (*i, j*) in the image and W represents the *z* value of each neighbour in a 3 x 3 neighbourhood.

The point *i, j* is considered a depth gradient if the respective $D_p(i,j)$ value is upon a threshold.

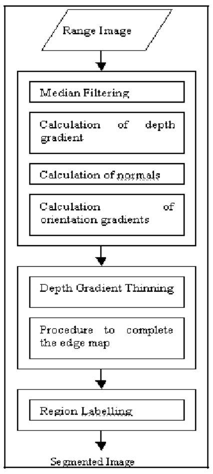


Figure1. Proposed segmentation method

Orientation gradients. To calculate the orientation gradients in a two-step process:

First, one normal is calculated for each point in the image. A 5 x 5 window is adjusted on each pixel and four points are taken in the north, south, west, and east directions in the window to obtain four vectors (Figure 2).

The normal for each point in the image is calculated by:

$$\vec{N}_{i,j} = \frac{\sum V_D \otimes V_{nextD}}{4} \qquad (3)$$
$$D = \begin{Bmatrix} North(N), South(S), \\ East(E), West(W) \end{Bmatrix}$$

where $V_D \otimes V_{nextD}$ is the cross product between the vector in direction D and the next vector in a clockwise direction (North–East, East–South, South–West, West–North).

The four direction vectors to calculate the respective normal must not be interrupted by a depth edge. If one of the four vectors is interrupted by a depth gradient, it is discarded in this procedure and the normal is calculated with the remaining vectors.

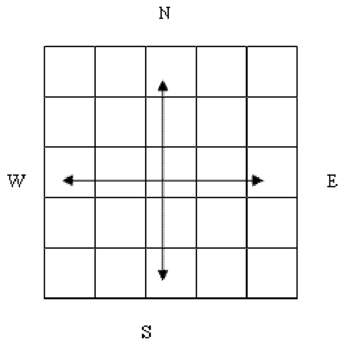


Figure 2. Direction vectors

Second, we calculate angular differences. Once a normal is calculated for each point in the image, then an angular difference is obtained for each normal. This procedure is performed with the following equations:

$$D_o(i,j) = \max(\theta_{k,l}) \qquad (4)$$

$$\theta_{kl} = arc\left(\frac{\left\|\vec{N}_{i,j}\right\| \cdot \left\|\vec{N}_{k,l}\right\|}{\vec{N}_{i,j} \bullet \vec{N}_{k,l}}\right) \qquad (5)$$

where $N_{k,l}$ represents the normals within a 15 x 15 window around a (*i, j*) pixel. If the $D_\theta(i,j)$ value is greater or equal to a threshold, then the (*i, j*) pixel is considered an orientation gradient.

c. Edge map

The gradients calculated define thin and fragmented edges and many false edge points. This can be seen in Figure 3.

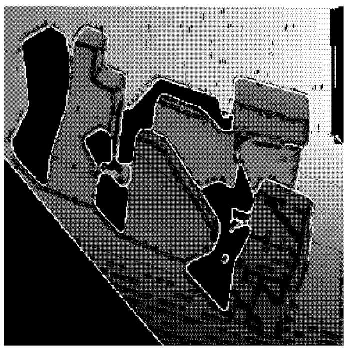


Figure 3. Orientation and depth gradients

The white pixels around the objects are the depth gradients and, the dark pixels are the orientation gradients. In Figure 3, one can see that the depth edges in the image are well-defined by the depth gradients. Therefore we use the depth gradients to obtain the first edges of our map. Hilditch's method [2] is used to thin out the depth gradients and to obtain the depth edges, however the depth edges obtained are not closed. A genetic algorithm is implemented to complete them and the orientation edges are generated from the orientation gradients.

B. Implemented genetic algorithm

In this work we use a genetic algorithm to find an edge map that delimits the surfaces of the range image to facilitate the segmentation.

In order, to implement a genetic algorithm the programmer must define: a representation scheme for each individual, the genetic operators, the fitness function, and a set of parameters necessary to control the algorithm. We have defined the initial population too. The implemented genetic algorithm is detailed below.

Representation scheme**.** Each individual in the population represents an edge map of 512 x 512 pixels, same size of the used images [11]. The edge maps in the population are two−dimensional binary arrays, where the value 1 represents an edge pixel and 0 a non−edge. Consequently our search space is too large. In our representation, $2^{512x512}$ combinations of 0 and 1 can occur.

Initial population. In order to reduce the amount of possible combinations, the genetic algorithm uses the obtained gradients to create an initial configuration of the edges for each individual of the population. In this stage of the genetic algorithm, we copy the depth gradient in each individual of the initial population after thinning them. Once the copy is made the mutation operator is applied to generate orientation edges or to complete a fragmented depth edge, so that the configuration of each individual is different.

The genetic operators. Our genetic algorithm uses the traditional crossover in which one point of a cut is selected at random in two individuals which are also taken at random, and the genetic information of both individuals is exchanged.

The mutation operator is very important in our genetic algorithm because the mutation constructs new edges in the configuration of each individual. The procedure made by the mutation operator selects random points in the image which are united with other points forming a closed edge. The selected point can be an orientation gradient or an end point of a fragmented depth edge. This procedure can be divided into the following steps:
First, the operator selects a starting point to be extended. The starting point must satisfy one of the following conditions:

- It is an end point in a fragmented edge;
- It is an isolated depth edge point;
- It is an orientation gradient that has not been selected before and without near edge points.

The following figure illustrates the three conditions.

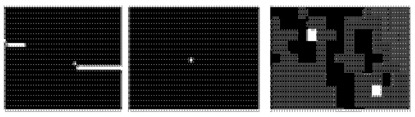


Figure 4. Possible starting point to mutate: (a) end point in fragmented edges; (b) isolated point; (c) gradients

Second, an objective point near the starting point is chosen to be united through a line segment. The objective point is chosen in a 13 x 13 window around the starting point in the following preference order:

- A near edge point;
- An orientation gradient in the same direction of a line segment already traced;
- A random orientation gradient in the windows not necessarily in the same direction of a line segment already traced;
- The point with the greatest angular difference without a direction gradient in the window.

Third, once the first line segment is traced, the objective point is taken as the starting point, and a new objective point is located according to the second step. This procedure is repeated in each mutation until the objective point is an edge point previously established.

Fitness function. The fitness function evaluates each individual in the population to measure how much it contributes to the solution of the problem. In our genetic algorithm, the fitness function is based on three criteria: number of edge pixels correctly located, number of small regions, and number of end points in fragmented edges.

In order to decide if an edge pixel is correctly located, a 5 x 5 window is placed around the pixel. If within the windows, the edge pixel separates two regions with different normals, the pixel is a correctly located edge. The procedure locates some points in the extreme of the window with its respective point opposite. If the angular difference between the normal's point is greater than a threshold, then the edge pixel is located between two different regions.
On the other hand, a region defined by the found edges is considered small if its size in pixels is smaller than a threshold.

The fitness function counts the occurrence of the three mentioned criteria and assigns the fitness of each individual *i* in the population by:

$$f(i) = WC\left(\frac{Ncorrect\_edge\_pixel}{Total\_edge\_pixel}\right) + WPR\left(\frac{1}{Nsmall\_region}\right) + WF\left(\frac{1}{Nend\_points}\right) \qquad (6)$$

where,

- *Ncorrect_edge_pixel,* is the number of pixels correctly located.
- *Total_edge_pixel,* is the number of edge pixels located by the genetic algorithm (all edge pixels except the edge pixels defined by the depth gradients).
- *NSmall_region,* is the number of small regions formed by the located edges.
- *Nend_points,* is the number of end points in the fragmented edges.
- *WC, WPR and WF* are weight assigned to each one of the mentioned criteria.

Our optimization problem is a multi-objective problem. We look for an edge map with closed edges, well located edges, and few small regions that could be noise. In order to solve the problem, we use the weighted sum approach for multi-objective optimization problems. This approach consists of adding all the objective functions together using different weighting coefficients for each. The multi-objective problem is transformed into a scalar optimization problem [14]. Other multi-objective optimization methods are more efficient than the weighted sum approach [15], however we used this approach because the computational time required by our segmentation method is very high and other multi-objective methods also require high computational time.

D. Labelling

The result of the explained genetic algorithm is a thin edge map without fragmentation. The obtained edges are labelled with the value zero. From the edge map we extract a segmentation of the range image by setting the same labels to connected pixels. First the labelling procedure locates a pixel not labelled in the image and assigns it a label (an integer value). Then the neighbours of the located pixel are also labelled with the same integer value and the neighbours of their neighbours until an edge pixel is found. Finally, the procedure locates another pixel without the label and assigns it a different label, and then labels the connected pixels. This continues until there are no pixels without a label in the image.

IV. Experimental Results

The proposed segmentation method was implemented on an Athlon 1.5 with 512 megabyte of memory. We used the genetic algorithm Library GALib for the implementation of the genetic algorithm. First a training procedure was carried out to assign values to the parameter of the segmentation algorithm (thresholds, parameter of the genetic algorithm and weight of the objective function), on 10 range images obtained from ABW database of the University of South Florida.

A.. Determination of algorithm parameters

Different values of thresholds to select depth gradients and orientation gradients were tried until the found values allowed obtaining satisfactory initial edge map (many true gradients, but little noise). The values finally established to the thresholds were: 15 units of range and 1 radian for the depth gradient threshold and orientation gradient threshold respectively.

Similarly, the genetic algorithm parameters were established manually, different values for each parameter were tried, and the following values were chosen.

Population Size	30 individuals
Number of Generations	50
Crossover rate	0.1
Mutation rate	0.4

Table 1. Values for the genetic algorithm parameters

The values for the weights of the objective function were assigned by the following procedure:

First, a value is assigned to WC (weight associated with the number of edge points correctly located in the edge map). For this parameter of the objective function, values between (0.4, 0.8) had more probability of being selected. Then, a value is uniformly selected between (0, 1-WC) to assign it to WF parameter (weight associated with the number of endpoint edges in the edge map). Finally, the result of (1 – (WC+WF)) is assigned to WPR parameter (number of small regions conformed by the located edge in the edge map). The value assigned to WC has more probability to be a value greater than the other parameters, because, if the edges are well located then few small regions must appear. Furthermore, a closed edge can also be a false edge.

This procedure was executed 20 times. A fitness average of the genetic algorithm was calculated from the fitness of each best edge map obtained for each training image (10). This fitness average was used to determine if a set of values is better than others. The set of values that generated a higher fitness average was selected for the weights as follow:

WC	0.5
WF	0.3
WFPR	0.2

Table 2. Values of WC, WF and WFPR

B. Segmentation of ABW images to test

After the segmentation algorithm parameters were established by the training procedure. We used the algorithm to segment 30 range images to test. The 30 images were also obtained from the ABW database of University of South Florida. Table 3 shows the fitness of each edge map obtained for each image and the fitness average.

Image	Fitness	Image	Fitness
0	0.521	15	0.950
1	0.724	16	0.851
2	0.880	17	0.930
3	0.967	18	0.744
4	0.804	19	0.908
5	0.630	20	0.869
6	0.906	21	0.788
7	0.937	22	0.909
8	0.796	23	0.877
9	0.859	24	0.905
10	0.909	25	0.901

11	0.956	26	0.955
12	0.924	27	0.909
13	0.937	28	0.820
14	0.867	29	0.814
Average 0,85823333			

Table 3. Fitness obtained for 30 range images to test

The maximum value for the fitness function in our algorithm is 1. Most of the fitness are near to the maximum value and the fitness average is high. This means that our algorithm finds edge maps with few false and fragmented edges. Figure 5 shows some edge maps obtained by our algorithm.

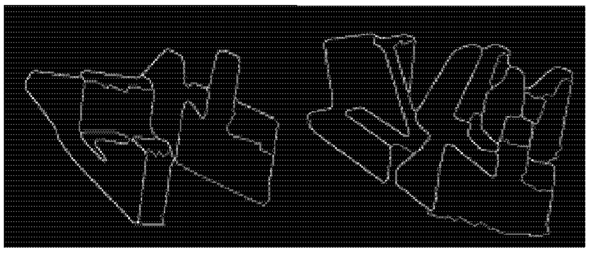


Figura 5. Obtained edge maps

The labelling procedure in Section 3.3 was made on the edge maps. We obtained segmented images of regions delimited by the edges. These segmentations were compared with other segmentation results generated with other segmentation methods. A visual comparison of results can be seen in figure 6 at the end of the article.

C. Evaluation of results

We evaluated the results of our segmentation algorithm using the methodology of evaluation proposed by Hoover [3]. The 30 range images to test are provided with their respective ground truth segmentation for this evaluation.

The compared segmenters have also been visually evaluated with the metrics proposed by Hoover. This information is available, so the evaluation comparison with other new methods is possible. In this section, we present the results of evaluation comparison of our segmentation method and the other four segmenters.

The perfect segmentation means the correct detection of all regions in the ground truth images with a tolerance under 100%. However all the segmentation methods until now perform poorly performance when the tolerance is greater than 80%. Table 4 at the end of this paper shows the evaluation results at a tolerance of 80%.

The resulting evaluation in the table indicates that our segmentation method is no better than other segmenters. However, the results are not very far from the others.

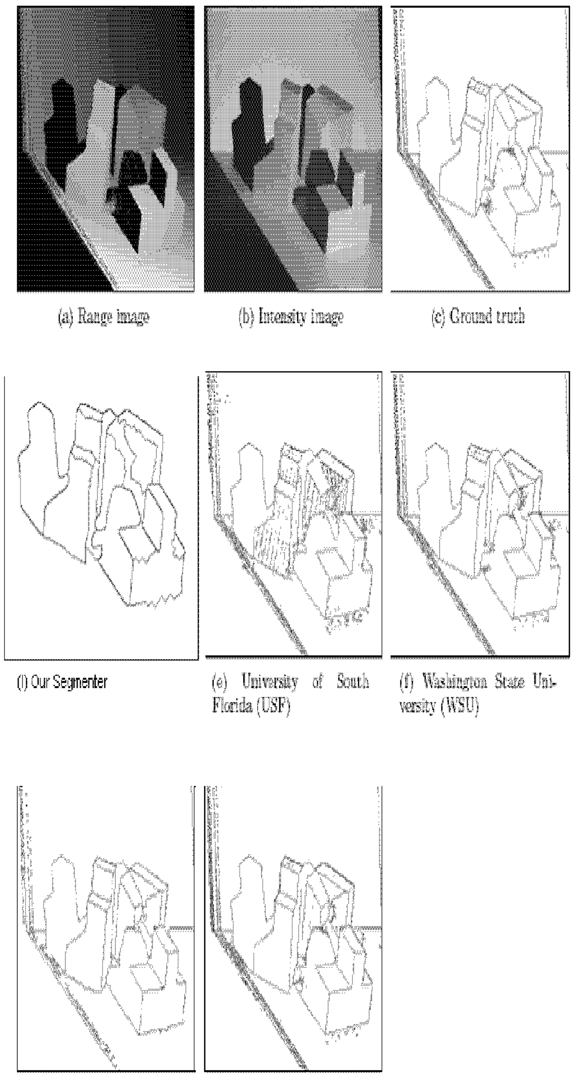
V. Conclusions And Future Work

In this article a segmentation method was presented. The method is based on three stages: first, the depth gradients and orientation gradients are calculated. In the second stage a genetic algorithm is used to find an edge map from the obtained gradients. Finally, a fast labelling is carried out.

The main difficulty in our method was the noise present in the processed range image. The noise in the images made the gradient calculation difficult and, consequently, the edge extraction.

The genetic algorithm implemented in the method found thin and closed edges, and considerably reduced the false edges detected by gradients. However the edge map does no describe very well the limit between the surfaces in the image. Perhaps this is the reason our segmenter's performance is no better than others.

Another approach like one based on scan line to extract edge map in range images could be tried to obtain an initial edge map with less noise than the procedure used in this work. We also thought about investigating other pre-processing procedures to reduce the high noise present in the range image of the ABW database.


(a) Range image
(b) Intensity image
(c) Ground truth
(l) Our Segmenter
(e) University of South Florida (USF)
(f) Washington State University (WSU)
(g) University of Bern (UB)
(h) University of Edinburgh (UE)
Figure 6. Visual comparison between different algorithm

VI. References

[2] R. González. Tratamiento Digital de Imágenes, Addison-Wesley, 1996.

[3] A. Hoover, Jean-Baptiste, Goldgof, Bowyer. A Methodology for Evaluating Range Image Segmentation Techniques, IEEE Workshop on Applications of Computer Vision, pp. 264–271, Sarasota, FL, December 1994.

[4] Jiang, Bunke. Fast segmentation of range images into planar regions by scan line grouping, Technical report IAM 92–006, Institute for Computer Science, University of Bern, Switzerland, April 1992.

[5] Jiang, Bunke. Edge Detection in Range Images Based on Scan Line Approximation, Computer Vision and Image Understanding, Vol. 73, No. 2, February, pp. 183–199, 1999.

[6] Jiagn, Bowyer, Morioka, Hiura, Sato, Inokuchi, Bock, Guerra, Loke, Dubuf. Some Further Results of Experimental Comparison of Range Image Segmentation Algorithms, 15th Int. Conference on Pattern Recognition, Spain, Sept. 2000.

[7] Heitkoetter, Beasley, The Hitch-Hiker's Guide, FAQ for comp.ai.genetic, 1993–2001, Available on internet: http://www.faqs.org/faqs/ai-faq/genetic/

[8] M. Powell, Comparing curved-surface range image segmenters, Master's thesis, Department of Computer Science and Engineering, University of Southern Florida, Apr. 1997.

[1] P. Besl, J. R. Jain. Segmentation through Variable-Order Surface Fitting, IEEE Transactions on PAMI, Vol. 10, Number 2, 1988.

[9] L. Silva, Estudo Sobre Deteção de Bordas em Imágenes de Profundidade, Departamento de Informática UFPR, 2000.

[10] L. Silva, O. Pereira, Segmentação de Imagens de Profundidade por Deteção de bordas, Proceedings of the 21st Brazilian Computer Society Congress, Thesis/Dissertations Contest—2nd Place Fortaleza/CE—Brazil, 2001.

[11] Range Image Database [online]. The Computer Vision / Image Analysis Research Laboratory at the University of South Florida. Available on internet: < http://marathon.csee.usf.edu/ >

[12] M. Wall, A. Galib. C++ Library of Genetic Algorithm Components [online]. Massachusetts Institute of Technology, 12 December 1999. Available on internet: <URL: http://lancet.mit.edu/ga/ >.

[13] A. Sappa, M. Devy, Fast Range Image Segmentation by an Edge Detection Strategy, IEEE, 2001 (complete).

[14] C. Coello, A. Christiansen, An Approach to Multiobjective Optimization Using Genetic Algorithms, In Dagli, C. H., Akay, M. Chen, C. L. P., Fernández, B. R., and Ghosh, J. (editors), Intelligent Engineering Systems Through Artificial Neural Networks, Volume 5, Fuzzy Logic and Evolutionary Programming, pp. 411—416, ASME Press, St. Louis, Missouri, USA, november 1995

[15] E. Zitzler, Evolutionary algorithms for multiobjective optimization: Methods and Applications", Ph.D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland.

Meth ods	GT Regions	Correctly detected	Over-Segmentation	Under-Segmentation	Missed	Noise
USF	15.2	12.7	0.2	0.1	2.1	1.2
WSU	15.2	9.7	0.5	0.2	4.5	2.2
UB	15.2	12.8	0.5	0.1	1.7	2.1
UE	15.2	13.4	0.4	0.2	1.1	0.8
OUR	15.2	10.8	0.8	2.4	5.2	2†.6

Table 4. Evaluation results of segmeters on the ABW database