BAYESIAN SEGMENTATION OF RANGE IMAGES OF POLYHEDRA OBJECTS USING ENTROPY CONTROLLED QUADRATIC MARKOV MEASURE FIELD MODELS

Carlos Angulo. José L. Marroquin and Mariano Rivera

Comunicación Técnica No I-07-09/11-06-2007
(CC/CIMAT)

Bayesian Segmentation of Range Images of Polyhedral Objects using Entropy Controlled Quadratic Markov Measure Field Models.

Carlos Angulo, José L. Marroquín and Mariano Rivera .
Centro de Investigaciones en Matemáticas A.C. Guanajuato, Gto., México
angulo , jlm, mrivera @cimat.mx

Abstract

In this paper, a method based on Bayesian estimation with prior MRF models for segmentation of range images of polyhedral objects is presented. This method includes new ways to determine the confidence associated with the information given for every pixel in the image as well an improved method for the localization of the boundaries between regions. The performance of the method compares favorably with other state of the art procedures when evaluated using standard benchmark.

1 INTRODUCTION

A range image is a representation of the distance from each point in a scene to a given reference point: the values of the pixels represent distances, which for representation purposes are showed as gray levels. Therefore, darker pixels represent points closer to the camera, while lighter pixels are more distant. Fig, 1 shows the difference between a range image and an intensity image taken with an ordinary CCD.

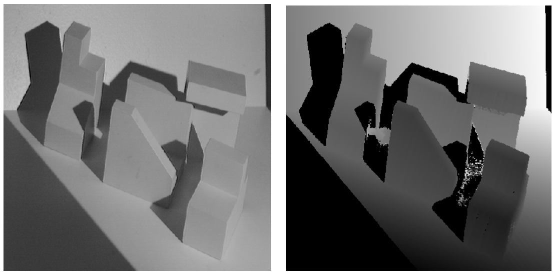


Fig. 1. Intensity Image and range image of the same scene

Although range cameras are nowadays available, they still are in development process and therefore expensive and limited in resolution. There are two main ways to obtain range images: Using a Laser Range Finder (LSR) or using a Structured Light Scanner (SLS). The Structured Light Scanner uses two optical paths, one for a CCD camera and other for a light projector. The distance is then computed by triangulation. The light projector uses fringe patterns of different widths. From one pattern to the next one, the frequency of the projected fringe pattern is doubled. For instance, if in the first pattern, half of the scene is not illuminated while the other half is illuminated, then the second pattern will have two bright fringes and two dark ones, and so on; usually, 8 patterns are used from which one obtains depth quantized in 256 levels [1].

Our work is focused in images of scenes with polyhedrons obtained by an ABW Structured Light Scanner [16].

For each Pixel in an ABW range image $I\left(i,j\right)$, $0 \le i,j \le 512$ one can compute the corresponding 3D Cartesian coordinates (in millimeters) using the following formulas [2]:

$$x\left(i,j\right)=\left(j-255\right)*\frac{\left(\frac{I\left(i,j\right)}{scal}+offset\right)}{\left|f_k\right|},y\left(i,j\right)=\frac{\left(255-i\right)}{c}*\frac{\left(\frac{I\left(i,j\right)}{scal}+offset\right)}{\left|f_k\right|},z\left(i,j\right)=\frac{\left(255-I\left(i,j\right)\right)}{scal} \quad (1.1)$$

Where $scal$, $offset$, c and f_k are values obtained through the camera calibration.

For the examples we are using, the origin of the coordinated system is in the center of the image, at 255 millimeters from the camera, the z axis head in the direction of the camera, and the x and y axis point to the top and right of the camera, parallel to the image.

In industrial applications, the objects to be digitalized are often polyhedrons; in this case, the 3D models may be constructed by finding the equations of the planar faces of the polyhedron, and the lines that separate such regions. This particular instance of the problem has been studied by a number of researchers [7][10][11][12][13][14].

The most usual approaches in this area are region growing and edge detection based algorithms. In the region growing based algorithms the main idea is to select a seed region (which can be a pixel) and keep adding nearby pixels, which fulfill some constraints, to the region until there are no more pixels to add. The different ways to choose a seed region and the properties that the pixels to be added to it should have result in different particular methods. The algorithms in [3][10] [18] [14] are instances of this approach.

The edge detector methods focus in the search for the boundaries between regions applying a large diversity of procedures. Most of them use a pre-process to find the possible edge pixels and also a post-process to fill gaps in the detected edges. The algorithms in [3][11] are instances of this method.

A different segmentation method using planes is presented by Li [8]. This method is based in the reconstruction of 3 piecewise smooth fields a, b and c, which correspond to the coefficients of the different planes. These fields are found by minimizing an energy function given by:

$$U(f)=\sum_i (a_i x_i + b_i y_i + c_i - d_i)^2 + \lambda \sum_i \sum_{i' \in N_i} g(\sqrt{w_a(a_i - a_{i'})^2 + w_b(b_i - b_{i'})^2 + w_c(c_i - c_{i'})^2}) \quad (1.2)$$

Where d is the original image, w_a, w_b, and w_c are smoothness parameters and g is a potential function. For range images, the author proposes $g(\eta)=\ln(1+\eta^2)$ and a gradient descent method for the energy minimization.

Although this method produces good results with synthetic images corrupted with white Gaussian noise, with real range images the results are not as good, mostly because quantization of the range data is not taken into account. Because of the way the range images are acquired they have a severe problem for any algorithm that tries to segment them in terms of planar regions. The images are quantized to 256 different values; therefore they do not have piecewise constant derivatives in a given planar region; the quantization of a plane in the image results in a stair-like pattern; Fig. 3 shows a detail of a planar region in one image where one can see that the derivatives corresponding to a single plane are not constant, and hence, they cannot be used to detect the borders between the regions. To make a good segmentation it is necessary to detect the three kinds of borders illustrated in fig. 2. Because of the quantization effect the detection of the roof borders is especially difficult. There are noisy points too. They are produced by specular reflections at the moment of taking the images.

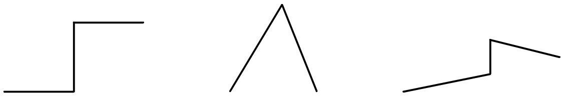


Fig. 2. The three kinds of borders: Step, roof and mixture.

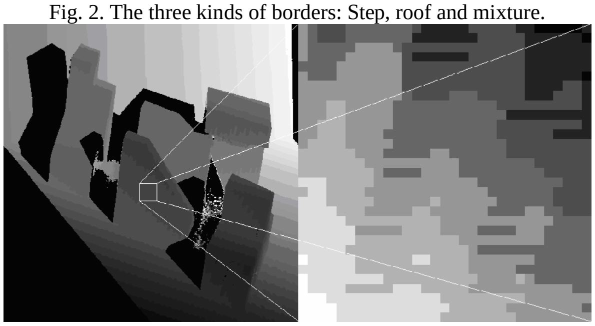


Fig. 3. Zoom of a Range Image

Because the structured light scanner uses two optical path ways, one for the projector and the other for the CCD, there are shadows in the images. These points do not give information about the bodies and must be eliminated as well. Another difficulty stems from the fact that, as can be noticed in (1.1), in the Cartesian coordinate system the grid of measured points is not regular.

The goal of this work is to present a segmentation method for range images of polyhedral objects, which overcomes these difficulties and improves significantly the performance of published methods. It is based in 2 ideas:

1) A preprocessing stage in which the spurious information originated by the quantization is eliminated.

2) A probabilistic approach to segment the images in terms of planar regions, using Bayesian estimation, which takes into account the polyhedral nature of the scene.

The results produced by this method feature a better localization of the borders between regions, making the reconstruction qualitatively more similar to the ground truth in a number of experiments. These results are also quantitatively better than other state-of-the-art segmentation algorithms, according to standard metrics.

In section 2 we present the algorithms used and the theory in which they are based. In section 3 we present the results obtained and make qualitative and quantitative comparisons with other published methods, and finally, give our conclusions.

2 PROPOSED METHOD

2.1 Bayesian estimation and hidden Markov random measure fields

The objective is to find a label field f which indicates which plane is supported for every pixel in the image. In particular, the label $f(r)$ for pixel r will be equal to k if and only if $r \in R_k$, where R_k is the region in the image that corresponds to plane k, which has parameters θ_k. In our method we do not try to find this field directly. Using the Hidden Markov Random Measure field technique [4] we search for a field of probability measures p from which, according to this model, the labels are generated in an independent way for each pixel. In this way, $p_k(r)$ represents the probability that $r \in R_k$ and the space dependency (granularity constraints) is encoded in the potentials for a prior Gibbs distribution for this hidden measure field, which in addition has to satisfy the constraints:

$$\sum_{k=1}^{K} p_k(r) = 1 \quad y \quad p_k(r) \geq 0 \quad \forall k, r \quad (2.1)$$

Where $p(r)$ is the discrete probability measure that corresponds to pixel r and K is the number of regions in which the image is being segmented.

According to the model derived in [4], the optimal estimator for the probability measure field for the one which minimizes the potential function:

$$U(p,\theta) = -\sum_{r \in L} \log\left(v(r,\theta) \cdot p(r)\right) + \lambda \sum_{\langle r,s \rangle} \left|p(r) - p(s)\right|^2 \quad (2.2)$$

Subject to the constraints (2.1), where $v(r,\theta)$ is the likelihood vector, in which:

$$v_k(r,\theta) = P\left(I(r) \mid r \in R_k, \theta_k\right) \quad (2.3)$$

And the second sum is over all the pairs of neighbors pixels r, s of the image and λ is a non-negative parameter. The likelihood can be any distribution which adapts for the problem, for example if a Gaussian distribution with variance σ^2 is assumed: $v_k(r,\theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(I(r) - m_k(r))^2}{2\sigma^2}}$. Where $m_k(r)$ is the value of the model k evaluated at the coordinates of pixel $r = (i, j)$; $m_k(i, j) = a_k x(i, j) + b_k y(i, j) + c_k$

Where $\theta_k = (a_k, b_k, c_k)$ is the parameter vector associated to model k.

The minimization of (2.2) is difficult, because of the high non linearity introduced by the log function.

In [6] it is shown that when each probability vector $p(r)$ has low entropy, the following approximation may be used:

$$\log\left(v(r)\cdot p(r)\right)\approx\sum_{k}\log\left(v_k(r)\right)p_k(r)^2 \quad (2.4)$$

Using this approximation, the optimal estimator may be found by minimizing the following energy function which is quadratic with respect to p :

$$U(p,\theta)=\sum_{k=1}^{K}\sum_{r\in L}-\log\left(v_k(r)\right)p_k(r)^2-\mu p_k(r)^2+\lambda\sum_{k=1}^{K}\sum_{r\in L}\sum_{s\in N_r}\left(p_k(r)-p_k(s)\right)^2 \quad (2.5)$$

Where N_r are the neighboring pixels for r; the up, down, left and right neighbor: $N_r=\left\{s\in L \mid \left|r-s\right|=1\right\}$. The term $\mu p_k(r)^2$ is added to control the entropy of p so that the approximation (2.4) may be used, i.e. promotes p to be as informative as possible.

Assuming a Gaussian distribution for the error, the potential to minimize is:

$$U(p,\theta)=\sum_{k=1}^{K}\sum_{r\in L}\left(\left(z(r)-m_k(r)\right)^2\delta_r-\mu\right)p_k(r)^2+\lambda\sum_{k=1}^{K}\sum_{r\in L}\sum_{s\in N_r}B(r,s)\left(p_k(r)-p_k(s)\right)^2 \quad (2.6)$$

In (2.6) we include new variables δ_r and $B(r,s)$. Both terms are weights added to obtain better quality in the segmentation. The term δ_r weights the confidence of the information given by the value of the image in the pixel r: a pixel with relevant information will have a value of $\delta_r=1$ while a pixel with spurious or irrelevant information will have a value close to 0. The term $B(r,s)$ weights the interaction between the pixels r y s; because a smooth probability field is desired, a weight of 1 is given for the majority of the neighboring pixels; to improve the localization of the borders between regions, these weights are reduced in places were a border is likely to be found.

During the energy minimization the constrains (2.1) must be satisfied, to enforce this, the Lagrange multipliers method is used. The Lagragian is given by:

$$L(p,\theta)=\sum_{k=1}^{K}\sum_{r\in L}\left[\left(\left(I(r)-m_k(r)\right)^2\delta_r-\mu\right)p_k(r)^2+\lambda\sum_{s\in N_r}B(r,s)\left(p_k(r)-p_k(s)\right)^2\right]-\sum_{x\varepsilon L}\gamma_x\left(1-\sum_{k}p_k(x)\right) \quad (2.7)$$

Once the optimal estimator for the p field is found, the label field is estimated by taking the mode of every estimated $p(r)$: $f(r)=k\Leftrightarrow p_k(r)>p_i(r)\ \forall i\neq k$.

To start the minimization of (2.7) the computation of some terms is needed. A group of starting models for the segmentation must be calculated, at the same time, the initial value for K must be obtained. The confidence of the information for every pixel has to be calculated to compute the δ field. Also the field B has to be calculated to define the interaction between pixels. In the followings subsections we describe the procedure for obtaining these terms.

2.2 Computation of the δ_r field

The main purpose of this stage is to eliminate the spurious information introduced by the quantization of the range images. The problem with an image of a quantized plane (see fig. 4) is that the gradient is zero in most points, and hence, it can neither be used to compute the plane parameters locally, nor as a criterion for determining the segmentation. In fact, even in the absence of noise, the true plane will pass only through a few points –labeled "corner points" in fig. 5.

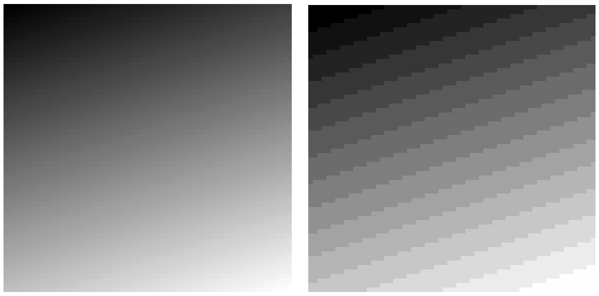


Fig. 4. A synthetic plane and its quantization

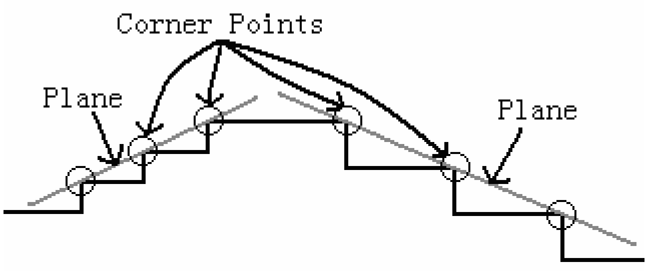


Fig. 5. Corner Points in one dimension

In one dimension, if the difference operators $D_{x^-}(I)$ and $D_{x^+}(I)$ are defined as:

$$D_{x^-}(I(x)) = I(x) - I(x-1)\,;\, D_{x^+}(I(x)) = I(x) - I(x+1) \quad (2.8)$$

The corners points fulfill:

$$D_{x^+}(I(x)) \cdot D_{x^-}(I(x)) = 0 \quad D_{x^+}(I(x)) + D_{x^-}(I(x)) > 0 \; (2.9)$$

Using the same idea, the same operators may be defined for two dimensions:

$$\begin{array}{l} D_{x^+}(I(x,y)) = I(x,y) - I(x+1,y)\,;\, D_{x^-}(I(x,y)) = I(x,y) - I(x-1,y) \\ D_{y^+}(I(x,y)) = I(x,y) - I(x,y+1)\,;\, D_{y^-}(I(x,y)) = I(x,y) - I(x,y-1) \end{array} \quad (2.10)$$

To find the corner points in two dimensions the points have to satisfy the conditions (2.9) in two directions, i.e. they have to be corner points in *x* and *y*. The corresponding conditions are:

$$\begin{array}{l} D_{x^+}(I(x,y)) \cdot D_{x^-}(I(x,y)) = D_{y^+}(I(x,y)) \cdot D_{y^-}(I(x,y)) = 0 \\ D_{x^+}(I(x,y)) + D_{x^-}(I(x,y)) > 0\,,\, D_{y^+}(I(x,y)) + D_{y^-}(I(x,y)) > 0 \end{array} \quad (2.11)$$

Using only the corner points, a local plane may be fitted to quantized data with an error that depends only on the noise and not in the quantization, as can be seen in figure 6.

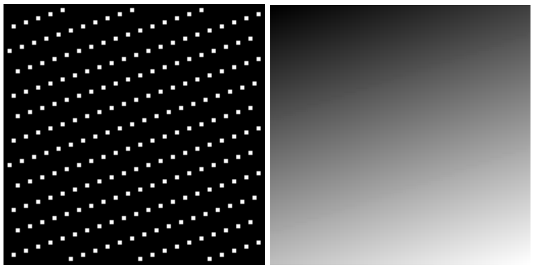


Fig. 6. Corner points of the plane in fig. 4 and the fitted plane. Fits with little error.

The process for finding the corner points is crucial to the method presented in this paper, because this helps to obtain the parameters of the initial models in which the image will be segmented and gives the values for δ_r. $\delta_r = 1$ is assigned if r is a corner point and a small value of δ_r is used if it is not (we use $\delta_r = 10^{-6}$).

It is also necessary to eliminate the pixels in shadowed areas from the computation. A pixel is considered to be in a shadowed area if $I(r) < 3$.

2.3 Initial estimation of plane's parameters

This procedure consists in successively sweeping the entire image with a window, looking for corner points which are in a plane. The process needs three parameters: an initial window size (initsiz), a minimum number of points in the window (minp) and the maximum distance between the points and the plane in the fitting process (maxdist). The process is presented in algorithm 1:

```
for $i$ =initsiz/2 to #columns-initsiz/2
  for $j$ =initsiz/2 to #rows-initsiz/2
    {
    Center the window in $(i, j)$;
    if inside the window there are minp or more corner points
      {
      Fill the matrix (2.12) with the information of the corner points inside the window;
      Solve the system to obtain the parameters of the plane;
      while all the corner points inside  the window are at a distance less than maxdist
        {
        if the calculated plane has not be added to the list of planes
          add the current calculated plane to the list;
        else
          change the plane calculated before for the actual plane in the list of planes;
        Fill matrix (2.12) with the information of the corner points inside the window;
        Recalculate the plane parameters solving the system;
        }
      if the plane was added to list of planes
        Delete all the corner points used to calculate the plane;
    }
  }
```

Algorithm 1

With this method one can compute an approximation for the planes in the scene and a value for K.

To obtain the correct plane parameters, it is very important to use the mapping to Cartesian coordinates given by Eq. (1.1), so that the least squares estimator for the parameters is obtained by solving the system:

$$\begin{pmatrix} \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} x(i,j)^2 & \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} x(i,j)\,y(i,j) & \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} x(i,j) \\ \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} x(i,j)\,y(i,j) & \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} y(i,j)^2 & \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} y(i,j) \\ \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} x(i,j) & \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} y(i,j) & \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} 1 \end{pmatrix} \begin{pmatrix} a_k \\ b_k \\ c_k \end{pmatrix} = \begin{pmatrix} \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} x(i,j)\,z(i,j) \\ \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} y(i,j)\,z(i,j) \\ \sum\limits_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} z(i,j) \end{pmatrix} \qquad (2.12)$$

Once the initial models and the value for K are obtained, the segmentation process can be initialized.

To initialize the search of the probability field, we use a field of uniform distributions, i.e., we set $p_k(r) = \frac{1}{K} \quad \forall r, k \le K$ where K is the number of planes.

2.4 Estimation of the B field

This field should be equal to 1 where the neighboring pixels should have an interaction and 0 where they do not, because a region border is there.

Three kinds of borders should be detected: Step, roof and the combination of both (see Fig. 2. The step and combination borders can be detected using a threshold in the difference operator used before, i.e., if r is the pixel at the left of the pixel s; we set $B(r,s)=0$ if $D_{x^+}(r) > \alpha$, where α is a parameter to be adjusted. The same is done in the vertical direction. At the same time the shadow pixels have to be disconnected from the pixels used for the segmentation, so we set $B(r,s)=0$ if r or s are shadow pixels.

The roof border detection is more complicated. These borders are in the intersection of the planes, but not all the planes intersections make a roof border in the image. Only the pixels in the intersection of two planes that are close to the interface between regions corresponding to the same planes should be disconnected.

The interface between the regions R_k and R_l is defined as the set of pixels that belong to either one of the regions, and that have at least one neighbor that belongs to the other one,

To detect roof borders -and hence, to set the corresponding $B(r,s)=1$- the following procedure is used:

1. – Select a pair of planes m_k and m_l with their respective regions R_k and R_l

2. – Search for the interface between regions is R_k and R_l. If it is empty, go to 5.

3. – Compute $xmin$, $xmax$, $ymin$ and $ymax$ such that for all pixels (x,y) in the interface $xmin \le x \le xmax$, $ymin \le y \le ymax$, with the values $xmin$ and $ymin$ as large as possible and $xmax$ and $ymax$ as small as possible. These values define a rectangle like the one in fig. 7.

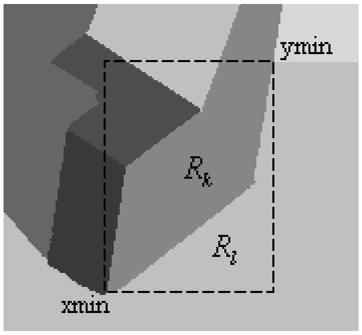


Fig. 7. Interfase entre regiones

4. – Search for the roof border inside the rectangle. Because of the mapping to Cartesian coordinates, the intersection between the planes cannot be found directly. To do this all the pixels (i,j) inside the rectangle have to be checked one by one. The equation of the intersection line is obtained equating the equations of the planes and solving for one of the variables: $y=\frac{(a_k-a_l)x+(c_k-c_l)}{(b_k-b_l)}$. If a point $x(i,j)$ corresponding to a pixel (i,j) is evaluated in the line, the value of the coordinate y en the intersection line is obtained. If the value of $y(i,j)$ is subtracted to this y value, the sign of the result shows if the pixel is over or under the intersection line. Doing the same to a neighboring pixel, one can determine the intersection of the planes passes between the two pixels. This is done for all pixels inside the rectangle:

$$B((i,j),(i+1,j))=1 \Leftarrow \left(\frac{(a_k-a_l)x(i,j)+c_k-c_l}{b_k-b_l}-y(i,j)\right)\left(\frac{(a_k-a_l)x(i+1,j)+c_k-c_l}{b_k-b_l}-y(i+1,j)\right)\le 0 \quad (2.13)$$

$$B((i,j),(i,j+1))=1 \Leftarrow \left(\frac{(a_k-a_l)x(i,j)+c_k-c_l}{b_k-b_l}-y(i,j)\right)\left(\frac{(a_k-a_l)x(i,j+1)+c_k-c_l}{b_k-b_l}-y(i,j+1)\right)\le 0$$

5. – If all the pairs of planes have been checked, stop, if not, go to step 1.

2.5 Minimization of the energy function

Once all the terms of function (2.7) have been calculated, one may use Gauss-Seidel method for the minimization with the following update rule:

$$p_k\left(x\right)=\frac{(1-\sum_{i=1}^{k}\frac{M_{i_x}}{D_{i_x}})}{D_{k_x}\sum_{i=1}^{k}\frac{1}{D_{i_x}}}+\frac{M_{k_x}}{D_{k_x}} \quad (2.14)$$

Where:

$$M_{k_x}=\lambda\sum_{s\in N_x}B\left(x,s\right)p_k\left(s\right)-\frac{\mu}{K},\Delta_{k_x}=(I(x)-m_k(x))^2\delta_x,D_{k_x}=\Delta_{k_x}+\lambda\sum_{s\in N_x}B\left(x,s\right)-\mu \quad (2.15)$$

To satisfy the non-negativity constraint of the probabilities, after every update, all negative probabilities are set to zero and normalization is done if needed.

2.6 Implementation Details

Because the values for the parameters for the initial models may be inaccurate, it is necessary to adjust them as the segmentation goes on. Thus, in our proposal, the plane parameters are re-computed using least squares every 10 Gauss-Seidel iterations, considering, for fitting plane m_k only the corner points x with $p_k\left(x\right)>0.5$. There, for example, instead of the upper left term $\sum_{\substack{(i,j)\in W\\ \delta_{i,j}=1}} x\left(i,j\right)^2$ in (2.12) one should use $\sum_{\substack{p_k(i,j)>0.5\\ \delta_{i,j}=1}} x\left(i,j\right)^2$ and the same for the remaining terms.

Note that it is possible that in the course of the iterations, the support of a plane k, i.e., the set $\left\{\left(i,j\right):p_k\left(i,j\right)>0.5\right\}$, is empty for some k. This may happen, for example, when the initialization process generates a spurious plane, or more than one instance of a given plane. In these cases, the planes with empty support are eliminated to avoid numerical errors and to accelerate the process.

It is also possible, however, that 2 instances of the same plane, generated in the initialization phase, continue to survive during the iterations with non empty support. To fuse these 2 instances into a single plane, the following criterion is used:

a) The planes should be almost parallel. This happens if the angle θ_{ij} between planes i and j satisfies: $\cos\left(\theta_{ij}\right)=n_i\cdot n_j>0.97$ where n_i and n_j are the unit normal vectors to the planes.

b) The planes should have compatible supports. To detect this, we define $close_i$ as the number of corner points at a distance less than 1 from plane i and $close_{ij}$ as the number of corner points at a distance less than 1 from planes i and j. We the say that

2 planes i and j have compatible support if $\frac{2\times close_{ij}}{close_i + close_j} > 0.9$.

If both conditions are satisfied, the plane with higher average probability for the points in the support of the planes is allowed to continue the process and the other plane is eliminated.

After the plane parameters have been re-computed and the number of models has been updated, the B field is also re-computed, using the procedure of section 2.4.

3 EXPERIMENTS

To make a comparison of our algorithm we used the image database from the web page of the Image Analysis Research Laboratory of the University of South Florida [15].

This database consists of 30 images, obtained with a ABW structured light scanner, together with the ground truths and the best segmentation obtained by this group.

3.1 Quantitative Evaluation

To evaluate the images we use the method in [3], which we transcribe here to make this paper self-contained.

This method classifies the regions obtained in the machine segmentation images in 5 groups:

- **Correctly detected regions:** A region in the machine segmentation which corresponds exactly to a region in the GT.
- **Under-segmented regions:** A region of the machine segmentation which corresponds to a group of regions in the ground truth.
- **Over-segmented regions:** A group of regions in the machine segmentation which correspond to a single region in the ground truth image.
- **Missed regions:** Regions in the ground truth image which does not correspond to any region in the machine segmentation.
- **Noise regions:** Regions in the machine segmentation which do not correspond to any region in the ground truth image.

Because making two regions to correspond exactly is very difficult, a tolerance value T is defined. This tolerance value controls the strictness of the evaluation and takes values between 0 and 1, which have the meaning of the overlapping between regions. The evaluation was performed with a tolerance of 95% for all test images. Table I shows the number of region correctly detected (labeled Correctly), over segmented regions (Over), under segmented regions (Under), missed regions (Missed) and noise regions (Noise) for every image, for both, our method (labeled CIMAT) and the method in [3] (labeled UE), which is the algorithm which has the best evaluation in [3][17][10] and is considered one of the state of the art methods in this area.

	Total	Correctly		Over		Under		Missed		Noise	
	Regions	UE	CIMAT	UE	CIMAT	UE	CIMAT	UE	CIMAT	UE	CIMAT
ABW Test 00	30	12	**13**	0	0	0	0	18	17	18	22
ABW Test 01	26	13	**16**	1	1	0	0	12	9	9	9
ABW Test 02	24	12	**13**	0	0	0	0	12	11	10	11

ABW Test 03	16	11	**12**	0	0	0	0	5	4	2	3
ABW Test 04	22	12	**13**	0	2	0	0	10	7	9	7
ABW Test 05	21	10	**11**	0	0	0	0	11	10	10	11
ABW Test 06	19	11	**12**	0	1	0	1	8	4	5	4
ABW Test 07	20	8	**11**	0	0	0	0	12	9	11	9
ABW Test 08	24	**15**	11	0	2	0	0	9	11	9	11
ABW Test 09	20	8	**10**	0	0	0	0	12	10	11	12
ABW Test 10	13	7	7	0	1	0	0	6	5	3	5
ABW Test 11	12	6	6	0	1	0	0	6	5	4	5
ABW Test 12	25	**9**	6	0	1	0	0	16	18	15	19
ABW Test 13	20	10	**14**	0	0	0	0	10	6	10	6
ABW Test 14	20	10	**13**	1	3	0	0	9	4	6	5
ABW Test 15	20	8	**10**	0	1	0	0	12	9	10	9
ABW Test 16	12	7	**9**	1	0	0	0	4	3	3	4
ABW Test 17	12	5	**8**	0	0	0	0	7	4	6	4
ABW Test 18	20	10	**12**	0	1	0	0	10	7	10	8
ABW Test 19	26	**11**	9	0	0	0	0	15	17	13	18
ABW Test 20	18	10	**13**	0	0	0	0	8	5	7	6
ABW Test 21	12	**8**	7	0	0	0	0	4	5	4	12
ABW Test 22	14	**5**	4	0	1	0	0	9	9	8	8
ABW Test 23	18	8	**11**	0	0	0	0	10	7	10	11
ABW Test 24	19	8	**11**	0	1	0	0	11	7	11	8
ABW Test 25	14	8	**9**	0	0	0	0	6	5	5	5
ABW Test 26	13	7	7	0	0	0	0	6	6	6	7
ABW Test 27	12	5	**7**	0	0	0	0	7	5	6	4
ABW Test 28	9	6	**9**	0	0	0	0	3	0	4	0
ABW Test 29	12	5	**7**	0	1	0	0	7	4	7	4
TOTAL	543	265	**301**	3	17	0	1	275	223	242	247

As one can see, the performance of our algorithm is better with respect to the number of correctly detected and missed regions. The number of over and under-segmentated regions is slightly higher in our procedure; this is due to very small or thin regions that appear mainly in larger regions, such as the wall and floor. This problem is produced in places where the noise is high and there is are no corner points nearby. In fig. 8 we show a typical example of two regions evaluated as over-segmented due to this problem.

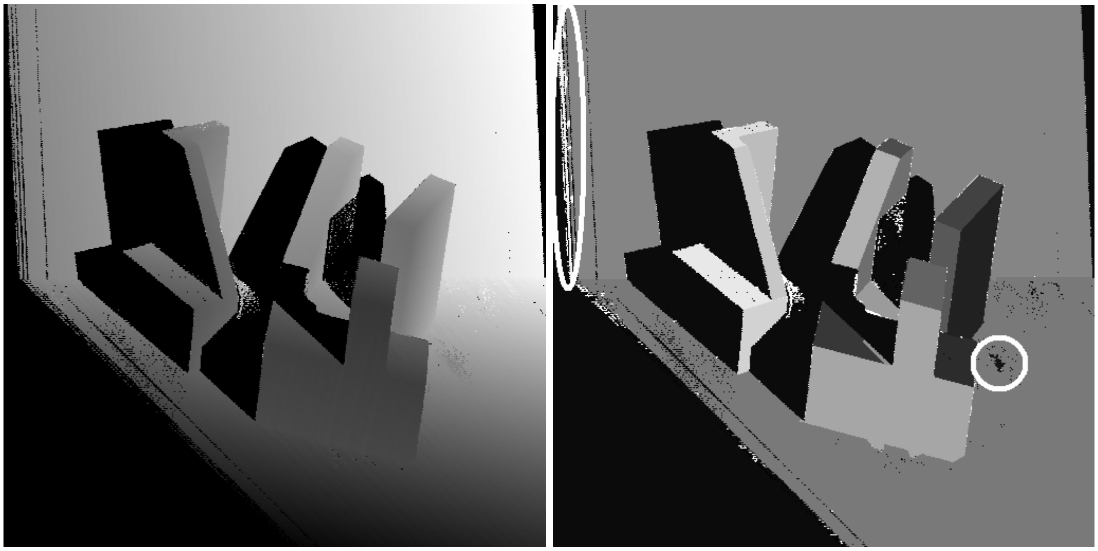
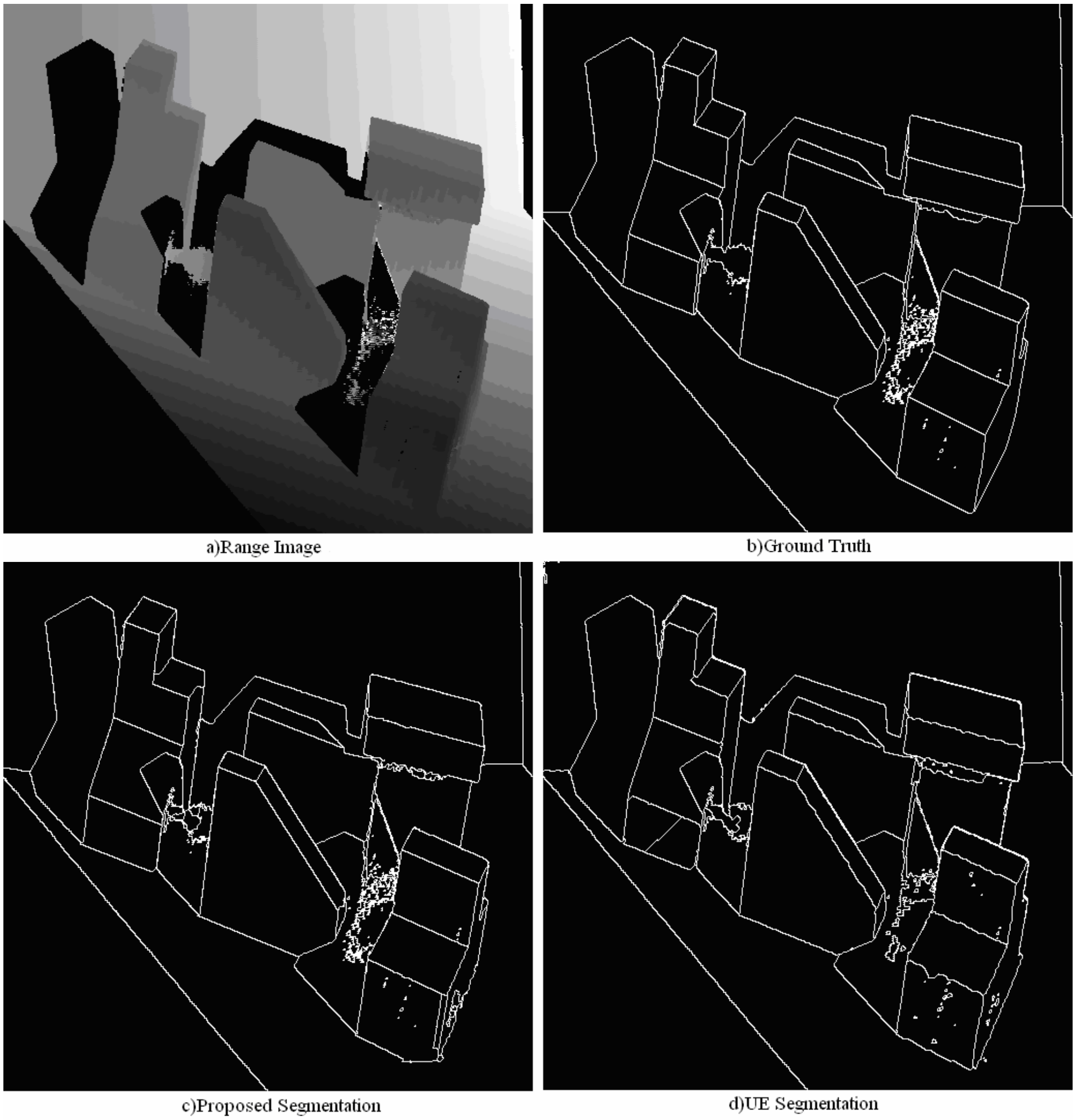


Fig.8 Example of the regions evaluated as Over-Segmentation, the regions produced by the noise are inside the white ellipses. Left: range image; right: segmentation produced by our method.

3.2 Visual Comparison

Fig. 9 shows an example of the results obtained with our algorithm compared with those in [15] (Labeled UFPR). As one can see, our results looks more similar to the ground truth, particularly in the neighborhood of borders between regions. This may be better appreciated in the close-up presented in fig. 10. Because the focus of this work is the segmentation of polyhedral bodies, we believe that the correct detection of the borders is particulary important.



a)Range Image

b)Ground Truth

c)Proposed Segmentation

d)UE Segmentation

Fig. 9

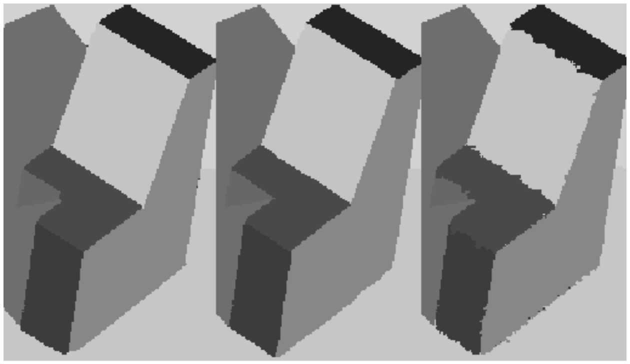


Fig. 10. Proposed segmentation, Ground Truth and UFPR segmentation

4 CONCLUSIONS

We have presented a method for the segmentation of range images of polyhedral objects, which is based in Bayesian estimation theory with a prior Markov random field model. The key elements for the success of this approach are the following:

1. The detection of corner points, and hence, the elimination of spurious information for fitting the planar models. This step is crucial for the correct initialization and updating of the models parameters.
2. The use of an appropriate geometrical model for the global Cartesian coordinates associated with each pixel in range image (Eq. (1.1)).
3. The use of a Quadratic Markov Measure Field model for the segmentation, which includes a confidence field (δ) and a spatially varying potential (based on the B field) that takes into account the analytic formula for the intersection of neighboring planes.

The performance of the resulting method compares favorably with other published approaches, both quantitatively and qualitatively.

5 REFERENCES

[1] A. Hoover, “The Space Envelope Representation for 3D Scenes”. Available in http://marathon.csee.usf.edu/range/seg-comp/SegComp.html

[2] A. HOOVER, “ABW Images: The Camera and Models.” (1995). Available in http://marathon.csee.usf.edu/range/seg-comp/SegComp.html

[3] Xiaoyi Jiang, Patrick J. Flynn, Horst Bunke, Dimitry Goldgof, Kevin Bowyer, David Eggert, Andrew Fitzgibbon, Adam Hoover, Gillian Jean-Baptiste, and Robert Fisher, “An experimental comparison of range image segmentation algorithms” IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 18, NO. 7, JULY 1996, pp. 673- 689

[4] Mariano Rivera, Omar Ocegueda and José Luis Marroquin, “Entropy controlled Gauss-Markov Random Measure Field for Early Vision”, VLSM 2005, LNCS 3752 pp. 137-148, 2005.

[5] Range image segmentation comparison project. http://marathon.csee.usf.edu/range/seg-comp/SegComp.html

[6] José L. Marroquin, Edgar Arce Santana y Salvador Botello, “Hidden Markov Measure Field Models for Image Segmentation”. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 25, NO. 11, NOVEMBER 2003

[7] Daniel Huber, Owen Carmichael and Martial Hebert, “3-D Map Reconstruction from Range Data”, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00), Vol. 1, April, 2000,

[8] Stan Z. Li, “Markov Random Field Modeling in Image Analysis"

[9] K. Sato and S. Inokuchi. “Three-dimensional surface measurement by space encoding range imaging”. J. Robotic Systems, 2:27–39, 1985.

[10] Jaesik Min, Mark Powell, Kevin W. Bowyer, “Automated Performance Evaluation of Range Image Segmentation Algorithms”, IEEE Transactions on Systems, Man, and cybernetics – Part B Vol. 34 No.1 February 2004 pp. 263-271.

[11] X. Jiang. “An adaptive contour closure algorithm and experimental evaluation” IEEE Trans. Pattern Anal. Machine Intelligence Vol.22, pp. 1252-1265, Nov. 2000

[12] R.W. Taylor, M. Sarini, A.P. Reeves, “Fast segmentation of Range Imagery into planar regions” Computer vision, Graphics, and image Processing, Vol. 45 No. 1 pp.42-60, 1989

[13] Chen Y. and G. Medioni, “Object Modeling by registration of multiple range Images” IEEE Conf. on Robotics and Automations, pp. 2724-2729

[14] X.Y. Jiang, H. Bunke “Fast segmentation of range Images into Planar Regions by Scan Line Grouping” Institute of informatics and applied Mathematics, University of Bern, Switzerland.

[15] http://marathon.csee.usf.edu/range/seg-comp/images.html

[16] T. G. Stahs and F. M: Wahl, “Fast and Robust Range Data Acquisition in a Low-Cost Environment”, in SPIE #1395: *Close-Range Photogrammetry Meets Match. Vis.*, Zurich,1990, 496-503..

[17] X. Jiang, K. Bowyer, Y. Morioka, S. Hiura, K. sato, S. Inokuchi, M. Bock, C. Guerra, R.E. Loke, J.M.H. du Buf. “Some further results of experimental comparison of range image segmentation algorithms,” Proc. Int. Conf. on Pattern Recognition, 2000.

[18] A.W. Fitzgibbon, D.W. Eggert, and R.B. Fisher, “High-level CAD Model Acquisition from Range Images,” technical report, Dept. of Artificial Intelligence, Univ. of Edinburgh, 1995.

[19] Xiao Wang and Han Wang, “Markov random field modeled range image segmentation”, Pattern Recognition Letters 25 (2004) 367–375