**Range image segmentation based on function approximation**

G. Maître, H. Hügli, F. Tièche, J. P. Amann

University of Neuchâtel, Institute of Microtechnology
Rue A.-L. Breguet 2, CH-2000 Neuchâtel
Switzerland

## ABSTRACT

This paper considers the segmentation of range image measurements into surface patches which are either plane or curved and which are described formally by a function. After a formal description of the segmentation, we present and compare three methods suited for plane and curved patch segmentation and show the results of experiments conducted for testing their practical behaviour. The two first methods use the classical approach of region growing whereas the third method is based on a relaxation process. This original and last method exhibits simplicity and low computational complexity. Thanks to its parallel nature, it can be considered as a good candidate for range image segmentation in real-time applications.

## 1. INTRODUCTION

In machine vision, there is a growing interest in range imaging. This is mainly because, in contrast to the traditional intensity images, range images give the true geometric shape of the object surface, which is a very intrinsic feature of the object. Also, as a consequence of this interest, many range image sensors are already available, and more performant sensors under development. Range imaging is successful where a simple geometric description of the scene where a straightforward data interpretation is needed, as for example in dimension control. However, in the case of more complex tasks requiring higher level interpretation of the scene, several processing steps are required: significant among those steps is the segmentation of range data into surface patches.

Image segmentation is a well known problem in computer vision[1]. In the case of range images, segmentation is considered as the basic step by which range data of a scene is divided into several regions. Each region stands for a surface patch which is uniform with respect to a given property and which distinguishes it from the neighboring patches. Segmentation based on function approximation creates patches whose points are approximated by one function per patch. An interesting feature of this method is that the segmented image is a complete range image description from which the image data can be reconstructed.

In this paper, we discuss and compare three methods used for the segmentation of range image data and present results from a series of experiments. The following features are common to all the methods: they are global and the function approximation is realized by least square polynomial fitting.

The principle of the first method is very general and has often been used[2,6,7,9]. It uses region growing by merging regions. In our application, it is used to segment a range image into planar patches.

The second method has been proposed by Besl and Jain[3]. It applies a region growing process around an initial uniform seed. A surface patch is approximated by one of an ordered family of functions which has the feature that a higher order function can describe a lower order one but not *vice versa*. In their implementation of the method, polynomials are used.

The third method is an original one which proceeds iteratively by diffusion. It is adequate for planar surfaces. It has a slightly poorer segmentation performance compared to the two other methods but, thanks to its simplicity, lower algorithmic complexity and parallel nature, it can be considered as a good candidate for range image segmentation in real-time applications.

We structure the paper as follows. First, the range image segmentation problem will be formalized and we will show how it is connected to the general segmentation problem (section 2). In section 3, we present the first method, performing region merging, followed by the second method which proceeds by region expansion around an initial seed (section 4). Section 5 is devoted to the original method based on a diffusion process. Finally, we presents results of experiments performed on sensed images and form a conclusion.

## 2. RANGE IMAGE SEGMENTATION

### 2.1 Image segmentation

We recall briefly the commonly used formal definition of image segmentation[1,3,7].

An image, in its general mathematical form, is a two-dimensional array X of n-tuples. Missing values are coded as NIL.

$$x_{ij} \in (\mathbb{R}^n \cup \{NIL\}) \qquad (1)$$

An image element or pixel $x_k$ is therefore defined by its position and its value

$$x_k = (i,j,x_{ij}) \qquad (2)$$

An image *region* $R_k$ is a set of pixels and an image *partition* P a set of image regions

$$P = \{R_1, R_2, \ldots, R_N\} \qquad (3)$$

such that

1) the union of the regions covers exactly the set I of all pixels

$$\bigcup_{k=1}^{N} R_k = I \qquad (4)$$

2) regions are pairwise disjoint

$$R_i \cap R_j = \emptyset, \quad \forall i,j \in [1,2,\ldots,N], i \neq j \qquad (5)$$

A region $R_k$ is said to be *connected*, if for every two pixels $x_i$, $x_j$ of $R_k$ there exists a path from $x_i$ to $x_j$ which lies entirely inside $R_k$. Two disjoint regions $R_i$ and $R_j$ are *adjacent* or *neighboring* if their union $R_i \cup R_j$ is connected.

Given the predicate $\mathcal{U}$, which indicates if a region is *uniform* (homogeneous), the image *segmentation* is realized by a partition $P_\infty$ such that

1) $\mathcal{U}(R_k)$ is true for every region $R_k$ in $P_\infty$ and
2) $\mathcal{U}(R_i \cup R_j)$ is false for every two regions $R_i$ and $R_j$

The methods we are presenting hereafter make the commonly used assumption that the uniformity involves the connectivity of the regions. As a consequence, for non-adjacent regions, $\mathcal{U}(R_{i\cup j})$ is false in any partition P.
(Note: Henceforth we write $R_{i\cup j}$ instead of $R_i \cup R_j$).

## 2.2 Range image

*Intuitive description*

"Range image" gives a good intuitive description of what it is: a range image can be considered as an digital video image where the intensity value is replaced by the distance to the object. Since each pixel has a corresponding direction of view, each range image element determines a point in the 3D space.

*Mathematical representation*

Hence, a range image may be represented by a two-dimensional array P of triplets standing for the cartesian coordinates of points in the 3D space.

$$p_{ij} \in (\mathbb{R}^3 \cup \{NIL\}) \qquad (6)$$

The value NIL stands for missing measures. The absence of measures may have different causes, depending on the ranging system and the transformations which have been applied to the range image[12].

*Intrinsic features of a range image*

Intuitively one recognizes that the mathematical description of a range image hides some intrinsic features, we try to highlight hereafter because they found the function based segmentation approach.

A range image results from a sampling of a two-dimensional surface in the 3D space. The surface can be decomposed into smooth patches seperated by discontinuities (fig.1). To each smooth surface patch $S_k$ corresponds a region $R_k$ in the image. A spatial coherence in the image region arises from the spatial coherence of the sensed surface patch.

If the image has been constructed by an approximatively parallel projection of the scene, surfaces may be considered as being entirely composed of $N_s$ smooth *graph* surface patches[3] $S_k(s_k,D_k)$ defined by

$$z = s_k(x,y), \text{ iff } (x,y) \in D_k, \ \forall k \in [1,2,\ldots,N_s] \qquad (7)$$

where $s_k(x,y)$ is a twice-differentiable function and $D_k$ is a domain or subset of $\mathbb{R}^2$. The domains do not overlap each other so that the overall sensed image can be considered as a piecewise-smooth graph surface. The coordinates x,y and the image coordinates i,j are linked together: the spatial coherence in the xy-plane corresponds to the spatial coherence in the range image and *vice versa*.

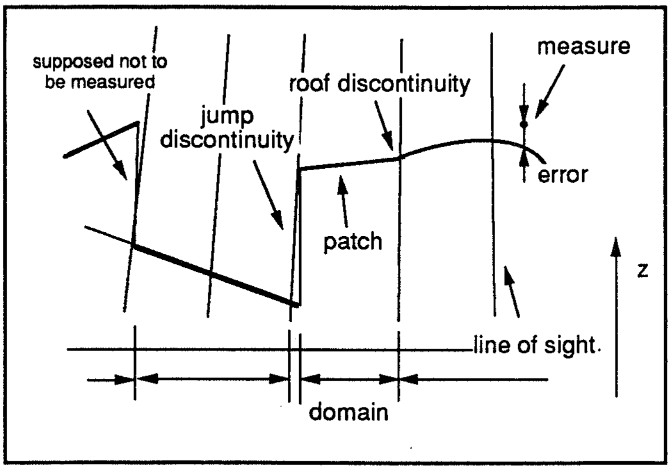


Fig.1. Decompostion of a sensed surface into smooth patches

We note that the functions are in general arbitrarly shaped. However, a restriction of range image analysis to certain classes of scenes provides constraints over the type of encountered functions $s_k(x,y)$. In the case of polyhedral objects, for example, the functions $s_k(x,y)$ are first-order polynomials.

If the range image points $p_{ij}$

$$p_{ij} = [x_{ij} \; y_{ij} \; z_{ij}]^T \tag{8}$$

are considered as being sampled on the sensed surface at position $(x_{ij},y_{ij})$, the measurement error is defined as

$$e_{ij} = z_{ij} - s_k(x_{ij},y_{ij}) \quad \text{for some } D_k \tag{9}$$

*Cartesian range image (CRI)*

In its most common form, a range image is a 2D array K of integers. This is true when the points are sampled on a regular grid and when the z coordinate is quantized. With the sampling interval $\Delta d$ in x and y, and the quantization interval $\Delta h$ in z, the cartesian coordinates of the 3D points are linked to the CRI data by the following equations:

$$x_{ij} = i\Delta d \quad y_{ij} = j\Delta d \quad z_{ij} = k_{ij}\Delta h \tag{10}$$

For our experiments, we use CRIs.

### 2.3 Segmentation based on function approximation

As seen in last section, the range image can be considered as sampled piecewise-smooth graph function. This motivates a segmentation approach based on function approximation: the segmented image should be constituted of *connected* regions over which the image data is well approximated by a smooth function.

*Approximation of image data over a single region*

We call image surface patch $A_k(a_k,R_k)$ the mathematical structure composed of an image region $R_k$ and a function $a_k(x,y)$. The patch $A_k(a_k,R_k)$ is a good approximation of the image data over $R_k$, if the approximation error

$$\varepsilon_k = \|\varepsilon_{ij}\|_{R_k} \tag{11}$$

with

$$\varepsilon_{ij} = z_{ij} - a_k(x_{ij},y_{ij}) \tag{12}$$

is small. The function norm $\|\|$ may be the max norm ($L_\infty$), the (Euclidian) root-mean-square error norm ($L_2$), or the mean absolute error norm ($L_1$).

*The unconstrained segmentation problem*

When no *a priori* knowledge about the scene is available, there exists no constraint over the smooth graph functions $s_k(x,y)$. In this case, the function based image segmentation problem consists in partitioning the image into surface patches so that both the number of patches and the overall approximation error are as small as possible.

Formally, we can state the problem as follows[3]:

Given the image **P**, find $N_a$ approximating functions $a_k(x,y)$ and $N_a$ connected image regions $R_k$, over which those functions are evaluated, such that the overall approxi-mation error

$$\varepsilon_I = \|\varepsilon_{ij}\|_I \tag{13}$$

and the total number of functions and regions $N_a$ are minimal.

The problem, as it has been defined above, is not specified enough to have a unique solution.

i) The functions must be smooth and described by a few number of parameters. In addition, each sensed surface patch $S_k$ of a simple object (e.g. polyhedron) should give rise to a single image patch $A_k$. However, there are no objective constraints on the functions.
ii) The minimization of $\varepsilon$ and the minimization of $N_a$ are contradictory. In order to apply optimization techniques, a cost function should defined where the influence of the two parameters are settled by mean of weights.

*The constrained segmentation problem*

Giving an objective constraint for the functions $a_k(x,y)$ and limiting the approximation error $\varepsilon_k$ to a threshold value $\varepsilon$,

$$\varepsilon_k < \varepsilon \tag{14}$$

give rise to a well defined problem:
under condition (14), the number $N_a$ of regions $R_k$ has to be minimized.
The segmentation based on function approximation is now linked to the general definition of image segmentation (§2.3). The predicate $\mathcal{U}$ is defined by

$$\mathcal{U}(R_k) = (\varepsilon_k < \varepsilon) \tag{15}$$

The two methods using region growing try to give an optimal solution to the so constrained problem. The third method is based on other constraints.

## 3. SEGMENTATION BY STEPWISE OPTIMAL REGION MERGING

### 3.1 Principles

The principle of region merging used to segment an image is very general and simple: starting with a partition in uniform regions, adjacent regions are merged two-by-two, resulting in bigger uniform regions. The process is stopped when no further uniform region can be created.

This idea has already been used by Horowitz and Pavlidis[1,7] in their split-and-merge algorithm. More recently, it has been used in several segmentation algorithms[2,6,9], also for the particular case of range image segmentation by function approximation.

While Horowitz and Pavlidis merged adjacent regions fulfilling the uniformity condition in a non-adaptive way, leading to locally optimal results, the three algorithms proposed use a stepwise (sequential) merging approach: at each step, only the two most similar regions are merged, thus delivering globally optimized results.

This stepwise optimal region merging (SORM) algorithm can be described as follows:

1. *find an initial image partition consisting of (small) uniform regions*
2. *merge iteratively the two most similar adjacent regions*
3. *stop when the dissimilarity is too high, that is when the merging would result in a non-uniform region*

The algorithm uses a measure of dissimilarity $d_{ij}$ between two regions $R_i$ and $R_j$. The dissimilarity is a function of quantities $v_i$ and $v_j$ attributed, respectively, to the regions $R_i$ and $R_j$. $v_i$ represents a "mean value" over the region $R_i$. When merging two regions $R_i$ and $R_j$, the value of the attribute $v_{i\cup j}$ and also the dissimilarity $d_{(i\cup j)k}$ between the new created region $v_{i\cup j}$ and each of its neighbors $R_k$ are updated.

3.2 Stepwise optimal region merging applied to segmentation based on plane approximation

The SORM algorithm is used to solve the function based segmentation problem defined with following constraints:

i) Planar patches $A_k(a_k,R_k)$ are only allowed, constraining the function $a_k(x,y)$ to first order polynomials, even more: to the least square error fitting first-order polynomial
ii) The uniformity predicate $\mathcal{U}(R_k)$ is defined by equation (15), the $L_2$-norm being used for $\varepsilon_k$.

We call this more specific SORM algorithm optimal planar patch merging (OPPM) algorithm.

*Dissimilarity measure*

As dissimilarity measure $d_{ij}$ we use the increase in least square approximation error if the two regions $R_i$ and $R_j$ are merged,

$$d_{ij} = E^2_{i\cup j} - E^2_i - E^2_j \qquad (16)$$

the approximation error being defined as

$$E^2_k = \sum_{R_k} \varepsilon^2_{ij} = L_k\,\varepsilon^2_k \qquad (17)$$

where $L_k$ is the number of pixels in $R_k$.

*Stopping condition*

The algorithm stops when the merging of the two most similar adjacent regions, corresponding to $\min(d_{ij})$ would result in a non uniform region.

$$\frac{E^2_{i\cup j}}{L_{i\cup j}} \geq \varepsilon^2 \qquad (18)$$

*Region attributes*

The vector of the polynomial coefficients $\mathbf{u}_k$ and all the quantities $\mathbf{S}_k$, $\mathbf{h}_k$, $E^2_k$, $L_k$ which are necessary to update the dissimilarity measure and to test the stopping condition are attributed to the region $R_k$. $\mathbf{S}_k$ is a 3x3 matrix and $\mathbf{h}_k$ a $\mathbb{R}^3$-vector, used to determine $\mathbf{u}_k$ by solving following equation[5,9]

$$\mathbf{S}_k\mathbf{u}_k = \mathbf{h}_k \qquad (19)$$

*Initialization*

The initial partition consisting of small uniform regions is realized by performing a plane approximation for every quadruple of points $\mathbf{p}_{ij}$, $\mathbf{p}_{(i+1)j}$, $\mathbf{p}_{i(j+1)}$, and $\mathbf{p}_{(i+1)(j+1)}$ of the range image. The values $\mathbf{S}_{ij}$, $\mathbf{h}_{ij}$, $\mathbf{u}_{ij}$ and $E^2_{ij}$ are computed, resulting in a corresponding image, partitioned into one-pixel regions.

*Attribute update*

When merging the two most similar adjacent regions $R_i$ and $R_j$, the attributes of the newly created region $R_{i\cup j}$ are updated

$$\begin{cases} \mathbf{S}_{i\cup j} = \mathbf{S}_i + \mathbf{S}_j \\ \mathbf{h}_{i\cup j} = \mathbf{h}_i + \mathbf{h}_j \\ E^2_{i\cup j} = d_{ij} + E^2_i + E^2_j \\ L_{i\cup j} = L_i + L_j \end{cases} \quad \mathbf{u}_{i\cup j} = \mathbf{S}^{-1}_{i\cup j}\mathbf{h}_{i\cup j} \qquad (20)$$

In the same time, the dissimilarities between $R_{i\cup j}$ and all its adjacent regions $R_k$ are computed also, merging them hypothetically:

$$d_{(i\cup j)k} = \mathbf{u}^T_{i\cup j}\mathbf{h}_{i\cup j} + \mathbf{u}^T_k\mathbf{h}_k - \mathbf{u}^T_{(i\cup j)\cup k}\mathbf{h}_{(i\cup j)\cup k} \qquad (21)$$

3.3 Discussion

Before testing the performances of the OPPM algorithm on data, its following features must be highlighted:

1) The algorithm has the advantage to be entirely global, however the merging process is purely sequential and cannot be parallelized.
2) The same algorithm may be used for higher-order polynomial approximations. However, the initial

partitionning must be changed[9,10] since more than four points are needed to determine polynomial of order more than 2.

3) The computational effort for a merging step is high, this essentially due to the evaluation of the dissimilarities between the newly created region $R_{i \cup j}$ and each of its neighbours $R_k$. In order to reduce this effort, the algorithm may be modified leting its optimality principle unchanged. A first idea is to reduce the number of iterations using an initial partition constituted of bigger regions[9,10].
   However, a second problem still remains. Scene discontinuities give rise to small regions which show high dissimilarity against their neighbours. In the beginning the iterative merging will let them unchanged while merging and so growing other regions. Hence, we often come to the situation where many small regions are adjacent to a big region. When merging the big region $R_i$ with one of its small neighbours $R_j$, the dissimilarity of the new region $R_{i \cup j}$ against all the neighbours has to be computed. This is useless since $E^2_{i \cup j} \approx E^2_i$. To avoid this situation, a conditional update can be introduced using a new region attribute $E'^2_k$. The purpose of $E'^2_k$ is to store what was the approximation error when dissimilarities have been updated. When merging the big region $R_i$ with the small region $R_j$, the following operation is performed:
   if $E^2_{i \cup j} \approx E'^2_i$, the dissimilarity is computed for neighbours of $R_j$ only and $E'^2_i$ is stored in $E'^2_{i \cup j}$, otherwise the dissimilarity is computed for all the neighbours and $E^2_{i \cup j}$ is stored in $E'^2_{i \cup j}$.

## 4. SEGMENTATION BY SEED EXPANSION

Besl and Jain described the method in details in their paper[3]. Hereafter, we show how it is linked to the other methods and where the differences are.

The function based segmentation problem is defined with the following constraints:

i) Function $a_k(x,y)$ is one of an ordered family of functions having the property that each higher order function can describe a lower order function but not *vice versa*. For each patch, the function with the lowest possible order should be selected. In practice, polynomials of order 4 or less are used.
ii) the uniformity predicate is defined by equation (15), the $L_2$-norm being used for $\varepsilon_k$.

We present just the core of Besl and Jain's solution:

Starting with an initial seed and a first-order polynomial, iteratively, a polynomial is fitted to the region data and the region is expanded as long as there are pixels compatible with the function. Whenever the approximation error is too big, a higher order polynomial will be used. The process stops when there are no more unused seeds.
More precisely, we write:

0. ***initialization***
   *choose a seed as the actual region $R_k$ and a first-order polynomial as the actual function $a_k(x,y)$*
1. ***function approximation***
   *approximate the points in the actual region $R_k$ by the function $a_k(x,y)$*
2. ***uniformity test*** $(\varepsilon_k^2 \geq \varepsilon^2)$
   *if $\|\varepsilon_{ij}\|_2^2 \geq \varepsilon^2$ then increase the order of the polynomial and go to step 1*
   *(stop if the maximal polynomial order is reached)*
3. **region growing**
   increase the actual region by all to $R_k$ connected pixels $x_{ij}$ which are compatible, i.e. for which $|\varepsilon_{ij}| \leq \alpha\varepsilon_k$
4. ***end-of-growing test***
   *stop if the size of $R_k$ has not increased in step 3, otherwise go to step 1*

The performance of the original algorithm[3] depends highly on the following two steps.

i) Preprocessing: the image surface curvatures are computed and initial seeds are created by eroding the regions of same curvature type.
ii) Postprocessing: neighboring patches with similar features are merged.

The algorithm has many parameters that have to be tuned.

## 5. SEGMENTATION BY DIFFUSION

This approach is quite different from the preceding ones. First, we note that an assumption over the scenes to treat is made: they are supposed to contain exclusively polyhedrons, and the images are supposed to be cartesian (11).

*Motivation, problem definition*

The motivation for the method is based on the following considerations. Since the scene contains polyhedral objects only, the functions which determine the smooth patches of the sensed surface are first-order polynomials:

$$s_k(x,y) = a_k x + b_k y + c_k \tag{22}$$

Hence, they satisfy the Laplace equation:

$$\Delta s_k(x,y) \equiv 0 \tag{23}$$

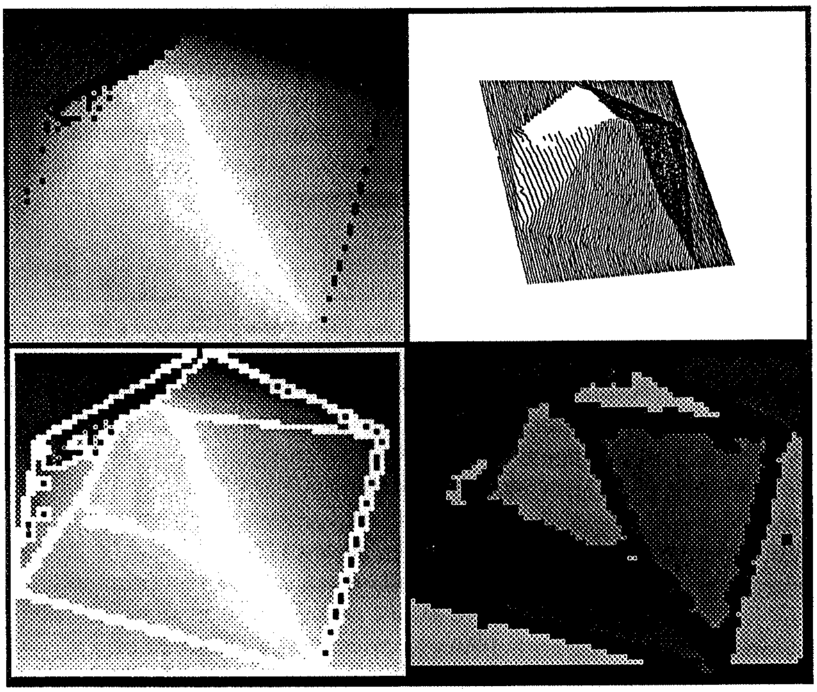

Fig.2
segmented range image "pyramid" (64x76)

top left: gray-scale coded range image
top right: perspective view of the range image
bottom left: segmentation by region merging
bottom right: segmentation by diffusion

If the range image would be ideal, i.e. exempt of measurement errors, the discrete form of the Laplace equation (24) would be satisfied for every position (i,j) except where discontinuities occur.

$$\Delta z_{ij} = z_{(i-1)j}+z_{i(j-1)}+z_{(i+1)j}+z_{i(j+1)}- 4\, z_{ij} = 0 \qquad (24)$$

Now, we restate the range image segmentation problem: given $\varepsilon$, find values $t_{ij}$ such that

$$\begin{cases}\Delta t_{ij} = 0 \\ |t_{ij}-z_{ij}| = \varepsilon\end{cases} \text{or} \qquad (25)$$

*Relaxation process*

The problem is solved by the use of a modified version of the successive overrelaxation process[4]. The solution $t_{ij}$ is obtained iteratively as

$$t'^{(n)}_{ij} = (1-\lambda)t^{(n-1)}_{ij} + \frac{\lambda}{4}\left(t^{(n-1)}_{(i-1)j} + t^{(n-1)}_{i(j-1)} + t^{(n-1)}_{(i+1)j} + t^{(n-1)}_{i(j+1)}\right) \qquad (26)$$

and

$$t^{(n)}_{ij} = \begin{cases} z_{ij}-\varepsilon & \text{, if } t'^{(n)}_{ij} \le z_{ij}-\varepsilon \\ z_{ij}+\varepsilon & \text{, if } t'^{(n)}_{ij} \ge z_{ij}+\varepsilon \\ t'^{(n)}_{ij} & \text{, otherwise}\end{cases} \qquad (27)$$

Equation (26) is in fact the discrete form of the heat or diffusion equation:

$$t'^{(n)}_{ij} - t^{(n-1)}_{ij} = \frac{\lambda}{4}\Delta t^{(n-1)}_{ij} \qquad (28)$$

The iterative process is initialized with

$$t^{(0)}_{ij} = z_{ij} \qquad (29)$$

and stops when

$$\max_{i,j} |\Delta t^{(n)}_{ij}| \le \alpha\varepsilon \qquad \text{with } \alpha \ll 1 \qquad (30)$$

A region of the segmented image consists of connected pixels such that

$$z_{ij}-\varepsilon < t^{(n)}_{ij} < z_{ij}+\varepsilon \qquad (31)$$

*Function approximation*

The first-order polynomial defining the corresponding planar patch is computed using only a few sparse points of the region.

The great advantage of this algorithm is that its iterative part can be entirely parallelized. It is also remarkable that equations (26) and (27) describe jointly the possible behaviour of a cellular neural network.

## 6. EXPERIMENTS AND RESULTS

The different methods have been tested on 15 sensed range images. While a part of the images come from the NRCC set of range images[12], some other images have been acquired using our own laser ranging system[8].

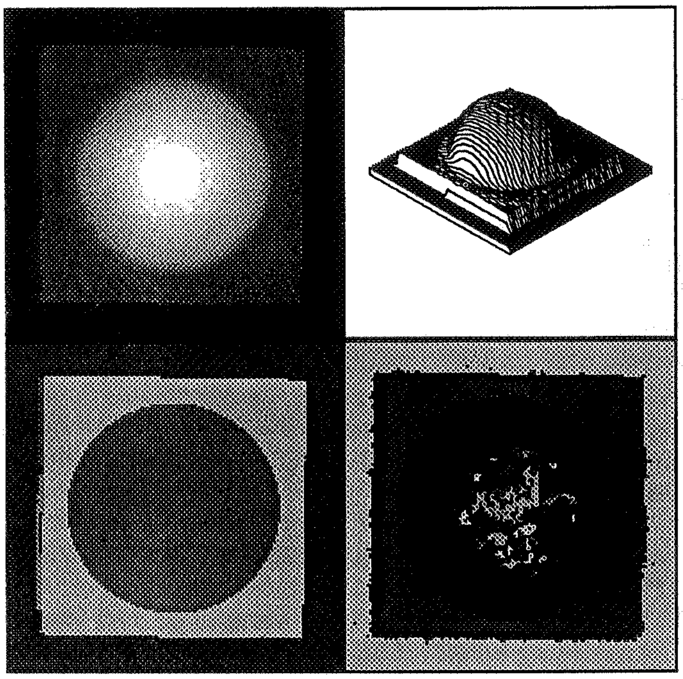

Fig.3. segmented range image "hemisphere" (128x128)

top left: gray-scale coded range image
top right: perspective view of the range image
bottom left: segmentation by seed expansion
bottom right: segmentation by diffusion

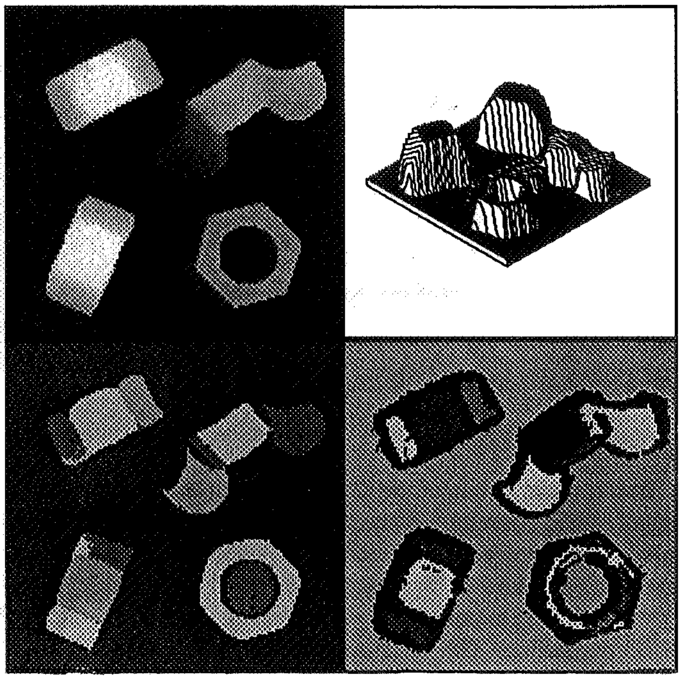

Fig.4. segmented range image "bolt" (128x128)

top left: gray-scale coded range image
top right: perspective view of the range image
bottom left: segmentation by seed expansion
bottom right: segmentation by diffusion

6.1 Stepwise optimal region merging

The stepwise optimal planar patch merging algorithm has been used in its straightforward version as described in §3.2. In order to keep the computation time within reasonable limits (about one hour on a μVAX 3400), the image size has been limited to approximatively 64x64 pixels.

For scenes containing only polyhedrons, the scene segmentation works as required. Figure 2 shows the results obtained for a range image of a pyramid: the regions are represented by their boundaries, superimposed to the range image.

6.2 Seed expansion

A simplified version of Besl and Jain's algorithm[3] has been implemented in LightSpeed™ Pascal on a Macintosh™ IIcx computer. The image size has been limited again, here to 128x128 pixels, so that the computation time does not overrun one hour.

The algorithm has been applied to different kind of scenes. The segmentation results are satisfying for scenes with manufactured objects as well as for natural scenes, the "hemisphere" image (fig.3) and the "bolts" image (fig.4), the results are as expected. For the "coffee-cup" image (fig.5) however, this is no more true: the concentric rings which are expected are partioned in finer regions.

Our experiments show that the algorithm works well as long as a sensed surface patch may be represented by a single polynomial function. For smooth graph surfaces which cannot be approximated by a single function on a single region, the algorithm becomes sensitive to the choice of the initial seeds and the additional merging step is necessary to obtain satisfying results.

6.3 Diffusion

The segmentation algorithm proceeding by diffusion has been applied to images representing polyhedrons and curved objects, using images of a maximum size of 256x256 pixels. Our experiments show an optimum of convergence for values of $\lambda$ slightly smaller than 1: we have chosen $\lambda = 0.96$. With $\alpha = 0.01$, the algorithm converges within 100 steps for all 256x256 images tested, corresponding in the worst case to 10 minutes of CPU on a μVAX 3400.

Figures 2 to 4 show that the algorithm performs fairly for planar surfaces and that it is not suited for curved surfaces.

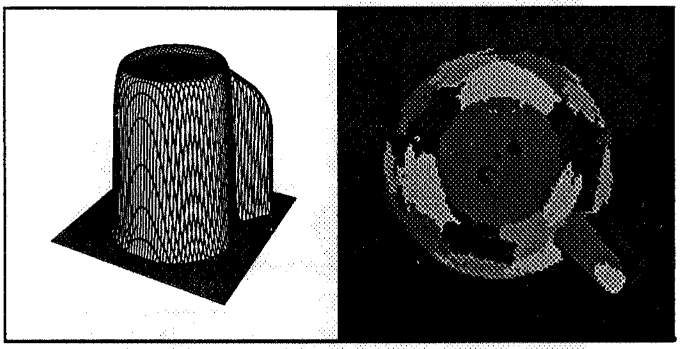

Fig.5
segmented range image "coffee-cup" (128x128)

left: perspective view of the range image
right: segmentation by seed expansion

## 7. CONCLUSIONS

In this paper, we have exposed the problem of range image segmentation based on function approximation and presented also three methods providing solutions. The first two methods are known from the litterature: both use "region growing", the first one by region merging, the second one by seed expansion. The third solution is new, proceeding by diffusion. The three methods have been tested on several images and compared, leading to results which can be summarized as follows:

The method proceeding by seed expansion gives the best segmentation results because of the use of polynomial functions of different orders. However, the method is not straightforward and several parameters must be tuned.

The second technique in turn, using region merging, is straightforward, but its realization showed a high computational cost which is not much reducible because of its purely sequential nature. The method can also be used to segment images of curved objects.

The newly introduced diffusion technique is suited for polyhedral objects only but, thanks to its simplicity and parallel nature, it is a good choice for real time applications. The analogy to both potential fields and neural networks mak make the diffusion method particularly interesting for further investigations.

## 8. ACKNOWLEDGMENTS

This work has been supported by the Swiss National Foundation for Scientific Research under project No. FN 20-25572.

## 9. REFERENCES

1. D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

2. J. M. Beaulieu and M. Goldberg, "Hierarchy in picture segmentation: A stepwise optimization approach", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-11, no. 2, pp. 150-163, February 1989.

3. P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-10, no. 2, pp. 167-192, March 1988.

4. S. Carlsson, "Sketch based coding of grey-level images", *Signal Processing*, vol. 15, no. 1, pp. 57-83, July 1988.

5. D. S. Chen, "A data-driven intermediate level feature extraction algorithm", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-11, no. 7, pp. 749-758, July 1989.

6. O. D. Faugeras and M. Hebert, "The representation, recognition, and positioning of 3-D shapes from range data", in *Three-Dimensional Machine Vision*, T. Kanade Ed., pp. 301-353, Kluwer Academic Publishers, Norwell, MA, 1987.

7. S. L. Horowitz and T. Pavlidis, "Picture segmentation by a directed split-and-merge procedure", in *Proc. 2nd Int. Joint Conf. Pattern Recognition*, pp. 424-433, August 1974.

8. H. Hügli and G. Maître, "3D by structured light: implementation and evaluation of a vision system for small parts", in *Optical 3-D Measurement Techniques*, A. Gruen and H. Kahmen Ed., pp. 153-163, Wichmann, Karlsruhe, Germany, 1989.

9. R. Leonardi, "Segmentation adaptive pour le codage d'images", Ph.D. thesis, no. 691, Dep. Elec. Eng., Swiss Federal Institute of Technology, Lausanne, Switzerland, 1987.

10. T. C. Pong, L. G. Shapiro, L. T. Watson and R. M. Haralick, "Experiments in segmentation using a facet model region grower", *Comput. Vision, Graphics, Image Processing*, vol. 25, pp. 1-23, 1984

11. T. Pun and P.-Y. Burgi, "Perceptual grouping of image primitives using a potential and field approach", in *Proc. 6th Scandinavian Conf. on Image Analysis*, vol. 1, pp. 317-324, Oulu, Finland,June 19-22, 1989.

12. M. Rioux and L. Cournoyer, "The NRCC three-dimensional image data files", Tech. rep. CNRC-29077, Division Elec. Eng., National Research Council Canada, Ottawa, Canada, June 1988.