Journal of Universal Computer Science, vol. 11, no. 6 (2005), 1102-1114
submitted: 23/3/04 accepted: 11/3/05, appeared: 28/6/05 © J.UCS

Collect the Fitted Surfaces Into Complex Based On C^0 Continuity

E.A. Zanaty
(Mathematics Department, Faculty of Science, South Valley University, Sohag, Egypt
zanaty22@yahoo.com)

M.R. Girgis
(Computer Science Department, Faculty of Science, Minia University, El-Minia, Egypt
moheb_r_g@yahoo.com)

Abstract: Surface reconstruction addresses the problem of creating a surface model from a point set digitized from a physical object. After performing the surface fitting on the bases of individual patches (adjacent surfaces), it is necessary to improve the obtained results by connecting the adjacent surfaces according to the desired smoothness. This paper presents a fast method for collecting the adjacent surfaces into complex based on C^0 continuity. The method works with the surfaces and their segments. Firstly, an arbitrary surface is selected, and the points with closest distances to that surface are extracted. Then, the points are sorted according to the Euclidean distance to the surface. Finally, the surface and the sorted points are joined together and presented to a *refitting* technique. This technique includes a procedure to decide if the data is similar to the current surface and for updating the surface parameters for each new point.

Keywords: surface reconstruction, reverse engineering, surface fitting
Categories: I, I.3, I.3.5

1 Introduction

The problem of building a 3D model from unstructured set of points measured from the surface of a given physical object appears in many areas including computer vision and reverse engineering [Faugeras, et al. 83], [Hoover et al. 96]. Building such a model is a problem of growing importance since efficient 3D scanning technologies such as laser range scanning became widely available. In order to sample 3D objects adequately, multiple scans have to be taken. Merging the measurements of each scan results in a large set of unstructured data points.

A number of techniques have been proposed for grouping unstructured set of points into appropriate subsets (segments) [Bajcsy et al. 90], [Besl and Jain 97], [Bolle and Vemuri 91], [Hebert and Ponce 82], [Hoover et al. 96], [Vanco et al. 2000], [Varady, et al. 97], [Varady et al. 98]. This process of subdividing the point set is referred to as segmentation. The best surface of an appropriate type can be fitted to each set of data points and this process is referred to as the fitting [Fitzgibbon et al. 97], [Bajcsy et al. 90], [Besl and Jain 97]. The simplest form of surface is the plane,

described by a first order equation. It is a relatively easy task to fit a plane to a set of points using least squares methods with low errors in the resultant approximation, and indeed, planes can be fitted more reliably than other surfaces.

The use of higher order surfaces than planes is necessary if real world objects are to be processed by a computer vision and graphics. There are many ways of representing higher order surfaces. Some representations model surfaces by means of surface patches. The simplest surfaces to use are planar [Faugeras et al. 83] but many methods exist [Beach 91], [Hoffmann 89], [Rogers and Adams 90], [Farin 90] to model non-planer surface patches. They are usually based on a parametric representation for the surface.

One common approach used in such methods is to specify a set of control points, which define a polygon mesh or net. The particular functions used for point coordinates are then created using a suitable combination of these control points with a set of basis functions. The overall shape of the surface patch can be modified by altering the positions of the control points. Each control point can be thought of as contributing an attractive potential that locally attracts the surface towards the point.

A variety of methods of this type have been formulated using different basis functions [Beach 91], [Hoffmann 89], [Rogers and Adams 90], [Farin 90]. Two of the most popular ones are the Bernstein basis (giving Bezier patches) and the B-spline basis. However, this type of surface representation is not easy to use for computer vision and graphics strategies. Among the problems are difficulties in uniquely representing a surface, especially with control points. It is impossible to describe the same surface with totally different control points. This will lead to difficulties with any subsequent recognition or other strategies that need to match a segmented surface with a stored one. Another way of representing surfaces is to describe them as implicit surfaces, but it may take some more complicated form. When it is a polynomial function, the surface generated is called an algebraic surface. Further discussion of algebraic surfaces can be found in the book by [Hoffmann 89].

Because of the problems with parametric representations, in vision and graphics application, two alternative representations of higher order surfaces are usually used instead. Either: the surface is represented by a general higher order algebraic surface, (a detailed analysis and explicit formulae for the non-linear least-squares optimization methods can be found in [Marshall and Martin 92], [Lukacs et al. 98]), or the surface is described as a particularly simple special case of higher order surface, representing a commonly used primitive geometric shape, such as a plane, cone, sphere or cylinder.

To obtain good representations for object reconstruction, it is generally sufficient to obtain approximate models, consisting of a collection of simple surfaces with proper continuity and smoothness in a mathematical sense.

Due to the diversity of surface representations and the algorithms for segmentation and surface fitting, the adjacent surfaces may not be continuous, e.g. two almost adjacent parallel planes may be segmented with a gap between them, or we may not have the desired degree of continuity.

However, region growing algorithms provide subsets of points to which well defined surfaces are fitted, but there may be gaps (i.e. unassigned data points) between them or they may partly overlap each other [Varady et al. 97], [Varady et al. 98]. Furthermore if one surface is segmented into more than one segment (region) due to the local irregularities in the surface and the appropriate surface is fitted to those

segments (regions) individually, we can get different surfaces and generally the continuity between them cannot exist. Now, the challenge is to collect the fitted surfaces with a desired continuity.

This paper presents a fast method for collecting the fitted surfaces to obtain the continuous model of the given object. In this method, a small threshold is taken for extracting the points with closest distance to a surface. The points are sorted according to the Euclidean distance from a surface and collected with the surface in a set. We apply the refitting technique to each set, where the surface is refitted to its segment by adding the points point by point and repeating the fitting step.

The paper is organized as follows: Section 2 describes the proposed method. Section 3 presents the experimental results of applying the proposed method to two different objects.

A regular object in which we are interested is bounded by relatively large primary surfaces. Primary surfaces include plane, sphere, cone and cylinder, where every surface in the object is surrounded by some adjacent surfaces.

In this section, we present the proposed method for finding a suitable connection between adjacent surfaces. The method works with surfaces and segments. We assume that each surface is surrounded by some surfaces. The points that have smaller distances to the surface than a given threshold are extracted. The extracted points are collected together in an array and sorted according to the Euclidean distance between the points and the surface. The array of extracted points is inserted with the surface into a set, whereby the measured object will be represented by some sets instead of surfaces. The refitting technique is applied to each set. Each surface is refitted by adding neighboring points and the fitting step is repeated. The refitting process for a surface continues until all the extracted points that belong to that surface and don't change its similarity constraints are added.

The proposed method can be divided into the following three steps:

- Extracting data points.
- Joining extracted points with surfaces.
- Applying the refitting technique.

The next subsections describe these steps in detail.

2.1 Data points extraction

In this subsection, we explain how to extract the closest points to a surface S from the adjacent surfaces. All possibilities of adjacent surfaces have been given to find the suitable and fast way for extracting data points. We don't distinguish between a surface and a segment and we refer to both by S. A point can be extracted from an adjacent segment/surface, if it is close enough to the surface S. In order to find a closest point to the surface, it is necessary to select a suitable threshold. Due to variations in the dimension and shape features of the object and the density, noise and distribution of the measured data, it is not easy to find a universal value for the threshold, which is appropriate in all cases. Therefore we search for a suitable threshold that depends on the surface location, i.e. whether the surface has a connection with other adjacent surfaces or not. After many tests on real point data, we found out that the suitable threshold is:

$$\delta = \frac{1}{t}\sum_{i=1}^{t} | d(S, p_i) |$$

where *t* is the number of points and $d(S, p_i)$ is the Euclidean distance between the surface S and the points $p_i, i = 1,2,...,t$. The threshold here has the advantage that it varies from surface to another based on the shape part and location of the point on the surface, i.e., δ will be very small if the surface S is very close to the points of the segment, and in this case the fitting of the surface to the segment is optimum and we don't need to update the surface. But a large value of δ indicates that the points of the segment don't lie on the surface and the surface must be refitted again by adding closest points from adjacent segments.

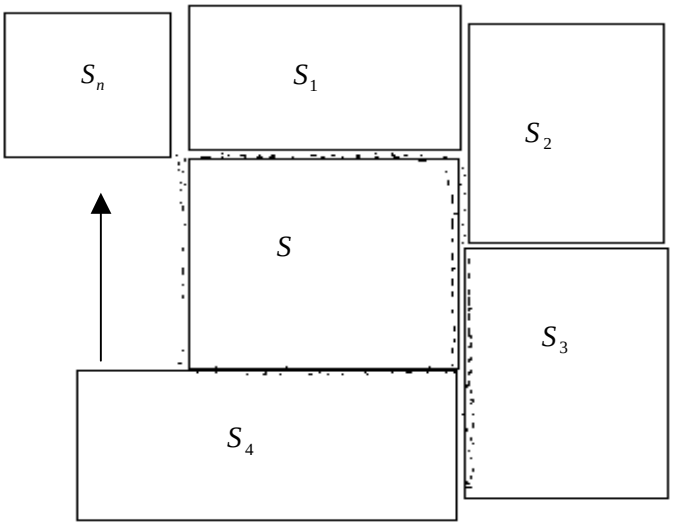


Figure 1: The adjacent surfaces, $S_i, i = 1,2,..,n$ *, which we want to connect with the surface* S .

Now, a point can be extracted if its distance is less than that threshold. This means that a point is considered close to the surface S , if and only if the following condition holds for this point:

$$| d(S, p_i) | \leq \delta \qquad (1)$$

Suppose we have *n* adjacent surfaces $S_i, i = 1,2,..,n$ to a surface S as shown in Figure 1. We start by selecting the surface and the segment of points S_1. Each point

in the segment S_1 which satisfies Eq.(1) is extracted. Then, the extracted points are inserted in an array P_1, and the corresponding distances between these points and the surface S are inserted in an array M_1. Thus, the arrays P_1 and M_1 can be written as follows:

$$P_1 = \{p_k\}_{k=1}^{r_1},$$
$$M_1 = \{d(S, p_k)\}_{k=1}^{r_1},$$

where r_1 is the number of points extracted from the segment S_1. Similarly, the closest points can be extracted from the surfaces $S_2, S_3, .., S_n$. The steps of the point extraction algorithm are given below.

Algorithm 1:

For $k = 1$ to n do

Select the segment S_k

Compute the Euclidean distance $d(S, p_i)$ between each point p_i in S_k and the surface S

Compute $\delta = \frac{1}{t}\sum_{i=1}^{t} | d(S, p_i) |$

Extract the points from the segment S_k that satisfy $| d(S, p_i) | \le \delta$

Insert the points into the array P_k

Insert the distances into the array M_k

End For

Stop

2.2 Joining extracted points with a surface

After extracting the closest points to a surface S from each adjacent surface S_i and inserting them in the arrays $P_i, i = 1,2,..., n$, we group these arrays into one array ψ as follows:

$$\psi = \bigcup_i P_i = \bigcup_i \{p_k\}_{k=1}^{r_i}, i = 1,2,.., n$$

where r_i is the number of extracted points from the surface S_i, $\{p_k\}_{k=1}^{r_i}$ are these extracted points, and n is the number of adjacent surfaces .

Also, the arrays, $M_i, i = 1,..., n$, of the corresponding distances between the extracted points and the surface S are grouped together into one array μ as follows:

$$\mu = \bigcup_i M_i = \bigcup_i \{d(S, p_k)\}_{k=1}^{r_i}, i = 1,2,..,n .$$

Finally, the array ψ of extracted points are sorted in ascending order according to their distances from S and inserted into an array Q. Then, the array Q and the surface S are grouped into one set γ:

$$\gamma = \{S, Q\}$$

where $Q = \{\bigcup_i \{p_k\}_{k=1}^{r_i}\}, i = 1,2,..,n$.

For example, if we have a plane α that has four boundaries as shown in Figure 2, then the array ψ of this plane is:

$$\psi = \bigcup_i P_i$$

where $P_i = \{p_k\}_{k=1}^{r_i}, i = 1,2,3,4$ are the four arrays that contain the points. These points are sorted and put into the array Q. The array Q and the surface α are grouped into a set γ:

$$\gamma = \{\alpha, Q\}$$

Note that, if we have two points p_k and p_j, the point p_k is inserted into Q before p_j if and only if $d(\alpha, p_k) < d(\alpha, p_j)$ and vice versa. This ordering is very important to maintain the same specification of the old plane, as we will see in the next section. In the same manner, the above steps can be applied to other shapes, such as spheres, cones and cylinder. The steps of the joining algorithm are given below.

Algorithm 2:

Step1: Collect the arrays $P_i, i = 1,2,..., n$ into the array ψ

Step 2: Collect the corresponding arrays $M_i, i = 1,2,..., n$ into the array μ

Step 3: Sort the points in ψ into the array Q

Step 4: Join the segment of the surface S with Q and insert it into the set γ

Step 5: Stop

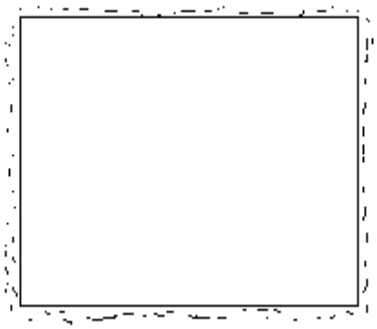


Figure 2: One set of a plane α.

2.3 The refitting technique

Now, the object in which we are interested consists of some sets instead of the surfaces. Every set γ is composed of a surface S surrounded by the points $p_k, k = 1,2,..,m$ in Q, where $m = \sum_{i=1}^{n} r_i$ is the number of points in the array Q and n is the number of the point subsets. Initially, we start by choosing an arbitrary set γ.

$$\gamma = \{S, Q\}$$

The surface is refitted to a segment by adding neighboring points $p_k, k = 1,2,..,m$, point by point and the refitting process stops when a termination condition is reached as we will see later.

Let the coefficients polynomial of the surface S be $\vec{q}$ and the coefficients polynomial $\vec{D}$ corresponds to the surface and a point p_k. Each data point is represented by its 3 coordinates, *x*, *y* and *z*, and the minimized equation of the surface with neighboring points (using the well known least squares method) becomes:

$$\text{Min } G(x,y,z) = \left| \sum_{i=1}^{t} S(x_i, y_i, z_i)^2 + \sum_{k=1}^{m} S(x_k, y_k, z_k)^2 \right|$$

where m is the number of points in the array *Q* and *t* is the number of points in the segment S.

Now we have the coefficients $\vec{q}$ of the first term of the above equation (when *m*=0) and we look for the solution after adding neighboring points one by one (when *m*>0).

Then, the above equation can be rewritten as:

$$\text{Min}\, G(x,y,z) = \left| \sum_{k=0}^{m} S(x_k, y_k, z_k)^2 \right| \rightarrow 0 \qquad (2)$$

where *k* is a parameter of the data points.

If *k*=0 then:

$$|\vec{D}| = |\vec{q}|$$

i.e. the surface S is obtained, and if *k*=1 then the surface S is updated by p_1.

By solving equation (2), the parameter vector $\vec{D}$ is derived. The termination criterion is:

$$\| \vec{D} - \vec{q} \| \geq \xi \qquad (3)$$

where ξ is a very small parameter.

The proposed technique works by selecting a surface S and the first point p_1 and continue until covering all the collection Q. The points are added to the surface starting with the nearest one to it. The vector of coefficients fits for every point and stops when the condition in equation (3) holds. When the growth of one surface stops we simply choose another set and start again. The whole process is repeated for all sets of the measured object.

There are some points in the set that are not added to any surface. These points are rejected. After many tests on the data (scanned data), we found that the proper setting of the parameters ξ of this algorithm is 0.001. The steps of the refitting algorithm, which is applied on each surface S_k of the considered object, are given bellow.

Algorithm 3:

Step 1: Select a parameter vector $\vec{q}$ of the surface S_k

Step 2: Select a point *p* from the array Q

Step 3: Estimate the parameter $\vec{D}$ by adding the point *p* to the surface S_k

Step 4: If $\| \vec{D} - \vec{q} \| \geq \xi$ stop

Step 5: Go to step 2 to select the next point

Step 6: Reject the remaining points in original segment

Step 7: End

3 Experimental Results

This section presents the results of applying the proposed method on a regular object to demonstrate its efficiency. The surface and point sets are fed to the proposed algorithm. The point sets are shown in Figures 3a and 4a, and the surfaces are shown

in Figures 3b and 4b. Before applying the algorithm, we show that the connection between surfaces does not exist due to diversity of fitted surfaces as shown in Figures 3b and 4b, whereby an extreme problem may happen especially during the object reconstruction, if the intersection curve between the surfaces is not available [Varady et al. 97]. The proposed algorithm doesn't fill the gap, if the sampling density isn't high enough, but it can achieve the continuity between the fitted surfaces.

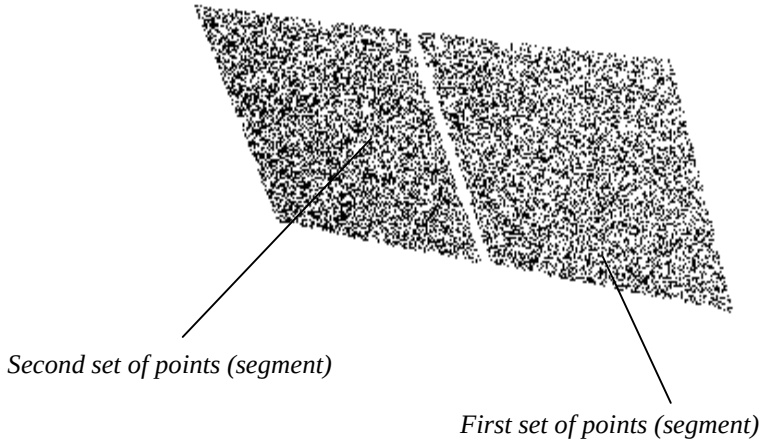


Figure 3a: A digitized data points on part of model object [4239 points].

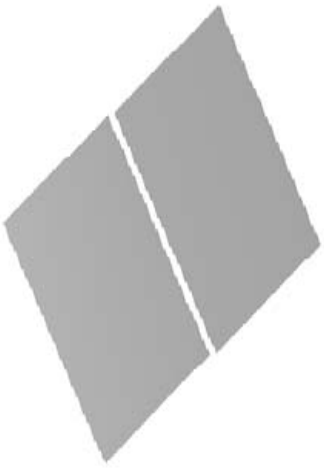


Figure 3b:Two parallel planes with a gap.

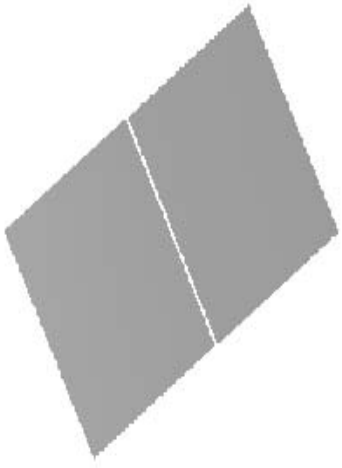


Figure 3c: The result after running the refitting algorithm with 30 points, when $\xi = 0.001$.

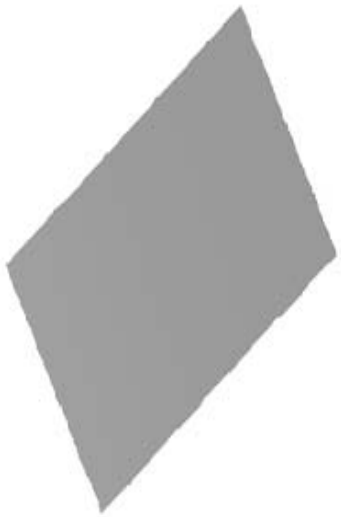


Figure 3d: The output result is one plane, when $\xi = 0.001$.

3.1 Example 1

In this example we present results of the application of the proposed algorithm for achieving the continuity. The algorithm is fed by surface information and data points, where the data points are acquired from a planar surface (part of model object [4239 points]) by 3D laser scanner, and they subdivide into two set of points (segments) as shown in Figure 3a. Thus, two planes are needed to fit the segments and the results are presented in Figure 3b. Figure 3b shows two parallel planes with a gap between them.

The first plane is fitted from 2439 points*:*

$$0.005812\,X+0.14547\,Y+0.9987\,Z+0.00017=0$$

while the second is fitted from 1800 points:

$$0.012954\,X+0.003164\,Y+0.9999\,Z+0.00000189=0$$

Then, algorithm 1 is applied on the first plane, where 30 points are extracted from the second segment. The surface and the neighboring points (30 points) are joined together in one cell, using algorithm 2. Then, algorithm 3 is applied, where the first surface is updated by these points, and the result becomes:

$$0.00812\,X+0.00943\,Y+0.9997\,Z+0.000043=0$$

Similarly, 30 points are extracted from the first segment, and the second surface is updated by these points, where the result is:

$$0.00923\,X+0.00864\,Y+0.99987\,Z+0.000005423=0$$

The result of the updated planes is shown in Figure 3c.

Again the two surfaces are updated by using algorithms 1 to 3, as we have seen above, where the final result of both planes, as shown in Figure 3d, is:

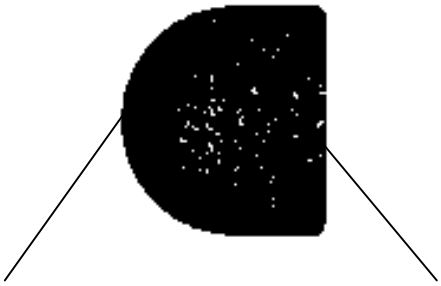
$$0.0001\,X+0.0087445\,Y+0.99996\,Z+0.0000026=0$$

3.2 Example 2

This example shows the applicability of the proposed method on the revolution surfaces. Figure 4a shows the data points that comes from mechanical part (point set of revolution surface [4280 points]). The revolution surface consists of cylinder and sphere point sets, where the cylinder segment has 1080 points and the sphere segment has 3200 points. The appropriate cylinder and sphere are fitted to each point set respectively, where the results are presented in figure 4b. Figure 4b shows adjacent cylinder and sphere, where there is no intersection between the sphere and the cylinder in the general case. However the intersection curve is not available between them, where it is demanded for creating the faces of the reconstruction object. The sphere center lies at the position (0.0001 , 0.000 , -0.00003) and has radius 2.000. The equation of the cylinder is:

$$(X+0.002249)^2+(Y-0.00244)^2+(Z-0.000015)^2-r^2-$$
$$(0.0004615 (X +0.0002249)+0.0002543$$
$$(Y-0.00244)+ 0.999999 (Z-0.000015))^2 =0$$

where c=(-0.002249, 0.00244, 0.000015) is the arbitrary point on the axis, N=(0.0004615, 0.0002543, -0.999999) is the unit vector along the axis and r=2.002 is the radius of the cylinder. Algorithm 1 is applied on the surfaces and their segments, where 150 neighboring points have been extracted from the cylinder segment. The points and the sphere are joined together and put into a cell, using algorithm 2. The sphere is selected and updated with 150 points using algorithm 3, where ξ=0.001 and δ=0.00074.



Spherical set of points (segment) *Cylindrical set of points (segment)*

Figure 4a: A digitized data points on part of revolution surface (point set "revolution")[4280 points].

Similarly, the cylinder is updated by 200 points (which have been extracted from the sphere segment). After applying our method, the centers of sphere and cylinder are changed slightly to become (-0.00009, 0.0000056, 0.0000) and (0.000001, 0.00007, 0.000004), respectively, where the radius of cylinder circle is 2.0006 and sphere is 2.003 and the result is shown in Figure 4c. Here the quality of the algorithm

can be determined from achieving the continuity between adjacent surfaces without changing the information of the fitting, whereby a continuos model can be obtained.

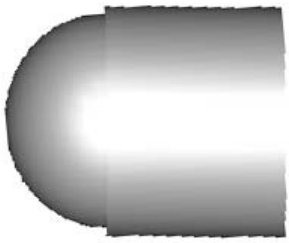


Figure 4b: The base sphere and cylinder surfaces after fitting, when $k = 0$.

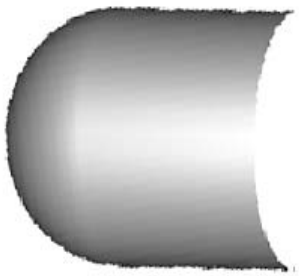


Figure 4c: The output result of the algorithm , when $\xi = 0.001$.

4 Conclusion

A method for collecting the fitted surfaces together into complex based on C^0 continuity was presented. In the proposed method, the points with closest distance to a surface are extracted. A surface and the points were joined together and inserted into a refitting technique, where the proposed technique includes a procedure to decide if a data is similar to the current surface and for updating the surface parameters for a given new point.

The continuity was tested between each two surfaces before and after applying the method using the procedure that was presented in [Zanaty 2002]. We noted that this method is faithful for obtaining a C^0 continuity. Our method has the particular advantage of being robust for finding the smoothness between adjacent surfaces even when the intersection between them is not available.

Acknowledgments

E.A.Zanaty is grateful to Prof. Guido Brunnett for his fruitful discussions and continuous help during the study at TU-Chemnitz, Germany.

References

[Bajcsy et al. 90] Bajcsy, R., Solina, F., Gupta, A.: "Segmentation versus object representation are they separable?", In Analysis and Interpretation of Range Images, Eds. R. Jain and A. K. Jain, Spring-Verlage, New York (1990).

[Beach 91] Beach R.C.: "An introduction to the curves and surfaces of computer aided design", Van Nostrand Reinhold, New York(1991).

[Besl and Jain 97] Besl, P.J., Jain, R.K. : "Segmentation through variable-order surface fitting", IEEE Trans. on Pattern Analysis and Machine Intelligence, 2, 10(1997), 167-192.

[Bolle and Vemuri 91] Bolle, R.M., Vemuri, B.C.: "On three-dimensional surface reconstruction methods", IEEE PAMI (1991),13, 1-13.

[Farin 90] Farin, G.: "Curves and surfaces for computer aided geometric design: a practical guide", Academic Press (1990).

[Faugeras et al. 83] Faugeras, O.D., Hebert, M., Pauchon, E.: "Segmentation of range data into planar and quadric patches", Proceedings of Third Computer Vision and Recognition Conference, Arlington, VA (1983), 8-13.

[Fitzgibbon et al. 97] Fitzgibbon, A. W., Eggert, D., Fisher, R.B.: "High-level CAD model acquisition from range images", CAD, 29, 4 (1997), 321-330.

[Hebert and Ponce 82] Hebert, M., Ponce, J.: "A new method for segmenting 3-D scenes into primitives", In Proc. 6th International Conference on Pattern Recognition(Munich, Germany Oct. 19-22), IEEE New York (1982), 836-838.

[Hoffmann 89] Hoffmann C.M., "Geometric and solid modeling : An introduction", Morgan Kaufmann, San Mateo, California, USA (1989).

[Hoover et al. 96] Hoover, A., Jean-Baptiste, G., Jiang, X., Patrick, J., Horst, F.: "An experimental comparison of range segmentation algorithms ", IEEE PAMI, 18(1996), 673-689.

[Lukacs et al. 98] Lukacs , G., Marshall , A. D., Martin, R. R. : "Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation", Proc. of the Fifth European Conference on Computer Vision, Freiburg (1998).

[Marshall and Martin 92] Marshall, A.D., Martin, R.R.: "Computer vision, models and inspection," World scientific Series in Robotics and Automated Systems, 4 (1992).

[Rogers and Adams 90] Rogers, D.F., Adams, J.A.: "Mathematical elements for computer graphics (2nd edition)", McGraw-Hill, New York, USA (1990).

[Vanco et al. 2000] Vanco, M., Brunnett, G., Schreiber, Th.: "A direct approach towards automatic surface segmentation of unorganized 3D points", Proceeding SCCG 2000 (2000).

[Varady et al. 98] Varady, T., Benko, P., Kos, G.: "Reverse engineering regular objects: simple segmentation and surface fitting procedures", International Journal of Shape Modeling, 4, 3 (1998),127-141.

[Varady et al. 97] Varady, T., Martin, R.R., Cox, J.: "Reverse engineering of geometric models-an introduction", CAD, 29,4 (1997), 255-268.

[Zanaty 2002] Zanaty, E.A. : "Algorithms for reliable surface fitting in reverse engineering", Ph.D. thesis, South Valley University, Sohag, Egypt(2002).