Pergamon

International Journal of Machine Tools & Manufacture 42 (2002) 167–178

INTERNATIONAL JOURNAL OF
MACHINE TOOLS & MANUFACTURE
DESIGN, RESEARCH AND APPLICATION

A new segmentation method for point cloud data

H. Woo, E. Kang, Semyung Wang, Kwan H. Lee *

Department of Mechatronics, Kwangju Institute of Science and Technology (K-JIST), 1 Oryong-dong, Puk-gu, Kwangju 500-712, South Korea

Received 15 March 2001; received in revised form 31 July 2001; accepted 2 August 2001

Abstract

In the process of generating a surface model from point cloud data, a segmentation that extracts the edges and partitions the three-dimensional (3D) point data is necessary and plays an important role in fitting surface patches and applying the scan data to the manufacturing process. Many researchers have tried to develop segmentation methods by fitting curves or surfaces in order to extract geometric information, such as edges and smooth regions, from the scan data. However, the surface- or curve-fitting tasks take a long time and it is also difficult to extract the exact edge points because the scan data consist of discrete points and the edge points are not always included in these data. In this research, a new method for segmenting the point cloud data is proposed. The proposed algorithm uses the octree-based 3D-grid method to handle a large amount of unordered sets of point data. The final 3D-grids are constructed through a refinement process and iterative subdivisioning of cells using the normal values of points. This 3D-grid method enables us to extract edge-neighborhood points while considering the geometric shape of a part. The proposed method is applied to two quadric models and the results are discussed. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Reverse engineering; CAD/CAM; Segmentation; Laser scanner; Point cloud data

1. Introduction

Reverse engineering is a technology that enables us to generate a computerized representation of an existing part based on point data acquired from the part surface. It can be useful for many situations: (1) when the design model of a product is created by an artist and computer-aided design (CAD) data of the model need to be captured; (2) when a product encounters frequent design changes in the development cycle and the initial design data become obsolete; and (3) when spare parts for a product are needed but its engineering drawing is lost or unavailable [1,2].

The typical process of reverse engineering begins with collecting point data from the surfaces of a physical object. For obtaining the part's surface data, either contact or non-contact type measuring devices can be used. Contact type devices are generally more accurate but slow in data acquisition, and vice versa for non-contact type devices. But, in recent years, the accuracy of the non-contact type of devices has improved significantly, and they are starting to be used in the production mode [3]. The point data from any type of device can be used in this research.

The initial point data acquired by a measuring device generally require pre-processing operations such as noise filtering, smoothing, merging and data ordering in order to be useable in subsequent operations. Using the preprocessed point data, a surface model can be generated either by a curve-net-based method or by a polygon-based modeling method, which are shown in Fig. 1.

The curve-net-based method is more commonly used when generating a surface model from the scan data. The polygon-based modeling method is fast and efficient in fitting surfaces; however, it generates less accurate surface models than those of the curve-net-based method and it is also difficult to modify the final model [4]. Each method has its advantages and disadvantages and either method can be used based on the geometric shape of a part. In either case, a segmentation process is always needed to create a surface model since it plays an important role in fitting surface patches.

After a part is scanned, the acquired point cloud should be divided into several smooth regions for surface-fitting purposes, which is called the segmentation

* Corresponding author. Tel.: +82-62-970-2386; fax: +82-62-970-2384.
E-mail address: lee@kyebek.kjist.ac.kr (K.H. Lee).

PII: S0890-6955(01)00120-1

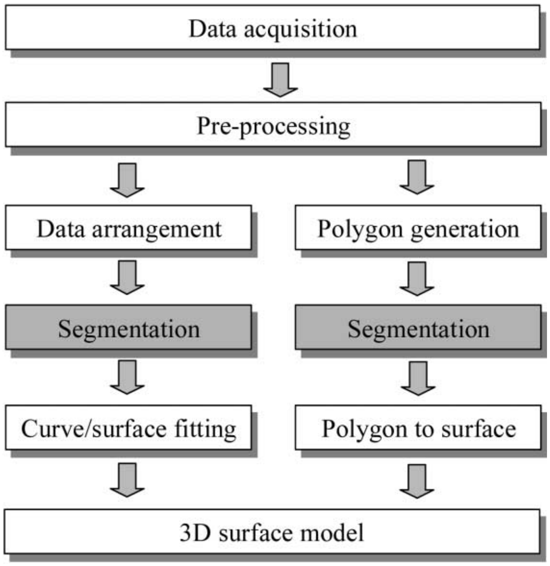

Fig. 1. Different flows for reverse engineering.

process. To facilitate surface modeling tasks, the point cloud should be arranged well and segmented. However, it is not easy for a computer to perform segmentation even for a simple object with quadric surfaces. The segmentation of an object with free-form surfaces, of course, is even more difficult and usually leads to ambiguous solutions. In reverse engineering, this segmentation has the greatest effects on product development time and the quality of the final surface model. The segmentation task is not only difficult but also time-consuming, and, in practice, has usually been performed by a human operator.

In this paper, a novel segmentation method using three-dimensional (3D) grids is proposed. Existing segmentation methods are reviewed in Section 2 and the proposed segmentation method based on 3D-grids is described in Section 3. Finally, the proposed method is applied to two sample parts that have quadric surfaces and the results are discussed in Section 4.

2. Review of existing segmentation methods

Segmentation has been an essential part in the process of surface modeling from scanned point data [1]. It is the process of partitioning a point cloud into meaningful regions or extracting important features from the point data. The majority of point data segmentation methods can be classified into three categories: edge-detection methods, region-growing methods and hybrid methods. The edge-detection methods attempt to detect discontinuities in the surfaces that form the closed boundaries of components in the point data. Fan et al. [5] used local surface curvature properties to identify significant boundaries in the data range. In order to avoid the loss of localization, scale-space tracking, which convolves the entire data with Gaussian masks having different values of the spread parameter, was employed. Chen and Liu [6] segmented the CMM (coordinate measuring machine) data by slicing and fitting them by two-dimensional NURBS spline curves. The boundary points were detected by calculating the maximum curvature of the NURBS spline. Milroy et al. [7] used a semi-automatic edge-based approach for orthogonal cross-section (OCS) models. Surface differential properties were estimated at each point in the model, and the curvature extremes were identified as possible edge points. Then an energy-minimizing active contour was used to link the edge points. Yang and Lee [8] identified edge points as the curvature extremes by estimating the surface curvature. After detecting edge points, an edge-neighborhood chain-coding algorithm was used for the formation of boundary curves. The original point set was then divided into subsets by a scan line algorithm.

The region-growing methods, on the other hand, proceed with segmentation by detecting continuous surfaces that have homogeneity or similar geometrical properties. Hoffman and Jain [9] segmented the range image into many surface patches and classified these patches as planar, convex or concave shapes based on a non-parametric statistical test for trend, curvature values and eigenvalue analysis. Besl and Jain [10] developed a segmentation method based on variable-order surface fitting. They made use of low-order bivariate polynomials to approximate a range of data. Through the estimation of Gaussian curvature and mean curvature, some core regions were identified. A region-growing method was then applied to find all boundary edges.

Most segmentation methods based on discontinuity can run short when a portion of an edge shows a small difference or when regions are homogeneous. The region-based segementation methods allow for reliable homogeneity in a region but these methods often fail to localize regional outlines accurately. To resolve the limitations involved in both approaches, hybrid segmentation approaches have been developed, where the edge- and region-based methods are combined. The method proposed by Yokoya and Levine [11] divided a three-dimensional measurement data set into surface primitives using biquadric surface fitting. The segmentated data were homogeneous in differential geometric properties and did not contain discontinuites. The Gaussian and mean curvatures were computed and used for performing the initial region-based segmentation. Then, after employing two additional edge-based segmentations from the partial derivatives and depth values, the final segmentation result was applied to the initial segmented data. Checchin et al. [12] used a hybrid approach that combined edge-detection based on the surface normals and region growing to merge over-segmented regions. Zhao and Zhang [13] employed a method based on triangulation and region grouping that uses edges, critical points and surface normals. In their algorithm, the initial edges and critical points were detected using morphological residues, and the linked pairs of edge points were triangulated. For the final segmentation, smaller surface triangular facets were expanded into large ones and segmented according to the surface normal information.

Most researchers have tried to develop segmentation methods by exactly fitting curves or surfaces to find edge points or curves. These surface- or curve-fitting tasks take a long time and, furthermore, it is difficult to extract the exact edge points because the scan data are made up of discrete points and edge points are not always included in the scan data. In this research, a new method is proposed that can extract edge-neighborhood points using 3D grids and segment the point cloud of an object with quadric surfaces without fitting any curve or surface. In addition, for extracting the geometric information from the scan data, a fast normal estimation method using the data structure of the scan data is proposed. This method can overcome data merging and overlapping problems caused by the additional scanning and registration process and it is applicable to automatic CAD modeling systems.

3. A proposed segmentation method

3.1. Overview of the process

The scan data obtained by non-contact measuring devices consist of a number of points that only include three-dimensional coordinates on the surface of an object. It is, therefore, difficult to obtain the geometric information of a part directly from the raw data. In order to extract geometric information, such as normals and curvatures, from the scan data, additional operations such as surface fitting, curve fitting or polygonizing are required. Once the geometric information of a part is obtained, it can be used for data reduction, segmentation and other applications.

To facilitate the acquisition of geometric information from the point data, a normal estimation method that is applicable to the point data acquired by a stripe-type laser scanner was developed. It must be performed after removing noisy points or outliers from the raw data. The proposed segmentation uses an octree-based 3D-grid splitting process that uses the iterative subdivisioning of cells based on the normal values of points, and the region-growing process to merge the divided cells into several groups. In the grid-splitting process, the evaluation of homogeneity is performed with point data in each cell. Therefore, the evaluation is less sensitive to noise than the other methods, where the amount of data is very small at the start of the segmentation process. As well, it can handle a huge amount of unordered point data. The overall procedure of the proposed segmentation method is described in Fig. 2.

3.2. Point normal estimation

When segmenting the scanned point data, the geometric shape of a scanned part should be taken into consideration. To get the geometric information of a part surface, point normal values or various curvature values, such as Gaussian, mean and principal curvatures, have been used. But it is difficult to calculate these values directly from the point data, and so many researchers have tried to estimate the normal and curvature values of point data by generating polygon models or by fitting parametric surfaces. Surface fitting, however, takes a long time and it is also difficult to acquire exact models from the scan data.

If the characteristics of the point data generated by various scanning methods are considered, the geometric information of a part can be estimated much faster. The point data can be classified as either structured or unstructured according to the scanning device or scanning method. For example, the point data obtained by laser or Moirè scanners are structured, while those obtained by portable CMMs or hand-held scanners are unstructured.

In estimating the point normals, Delaunay triangulation is usually used regardless of the point data type. For unstructured data, point normals are calculated after

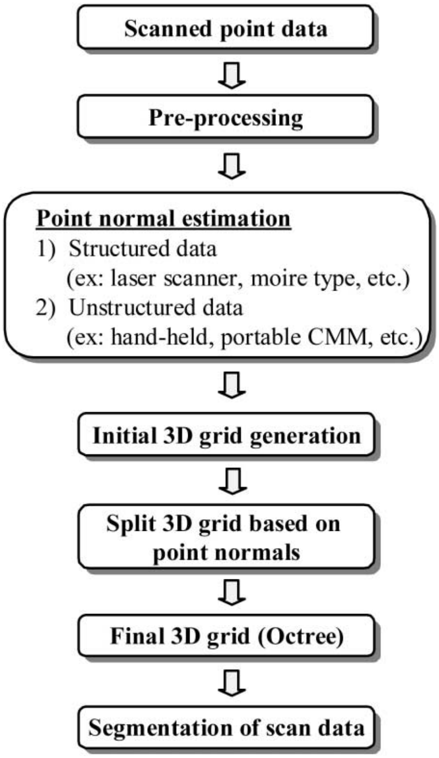
Scanned point data
Pre-processing
Point normal estimation
1) Structured data
(ex: laser scanner, moire type, etc.)
2) Unstructured data
(ex: hand-held, portable CMM, etc.)
Initial 3D grid generation
Split 3D grid based on point normals
Final 3D grid (Octree)
Segmentation of scan data

Fig. 2. Overall procedure of the proposed method.

Delaunay triangulation [14], but for structured data, different normal estimation methods can be used considering their patterns.

The point data acquired by a stripe-type laser scanner have an inherent order. Each scan path consists of a series of line segments, as shown in Fig. 3(a), which shows well-ordered scan lines. Fig. 3(b) shows the scan data generated as the laser probe proceeds along the *x*-axis. Each point is represented by $P_{i,j}$, where i denotes the scan line number (i=1, …, m) and j is the order of the point in a scan line (j=1, …, n).

To estimate the normal vectors of the points, the scan data shown in Fig. 4(a) first need to be triangulated. For this type of structured data, triangulation can be performed by using only two neighboring scan lines. It is therefore possible to obtain the point normals automatically during the scanning process. Fig. 4(b) shows an example of triangulated scan data.

As shown in Fig. 4(b), three vertices of a triangle cannot be in one scan line; that is, when two vertices are in the same scan line, the last one must be in another. In addition, two edges are sufficient to obtain the normal vector of a triangle, since it can be calculated using the cross product.

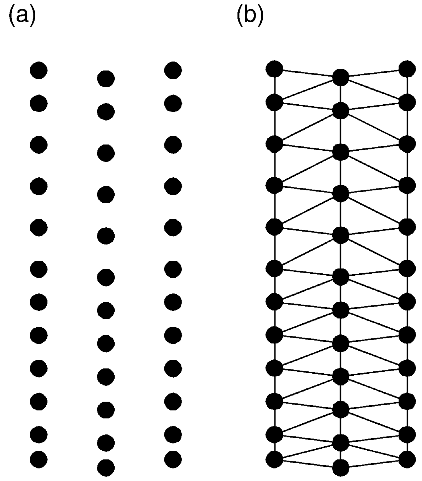

Fig. 4. Triangulation of scan data. (a) A stripe type laser scan data; (b) Triangulated point data.

To determine the orientation of a point normal, it is assumed that the origin of each *edge-pair* stays on the second line. Since this method is based on two-dimensional (2D) triangulation and the *x*-coordinate values of two scan lines are the same, the distance along the *y*-axis is the criterion for deciding edges. In the proposed algorithm, the scan data are triangulated before registration.

With the given scan lines labeled as (i)th and (i+1)th, the first edge for the first *edge-pair* is constructed by connecting the first point, $P_{i+1,1}$, in the (i+1)th line and

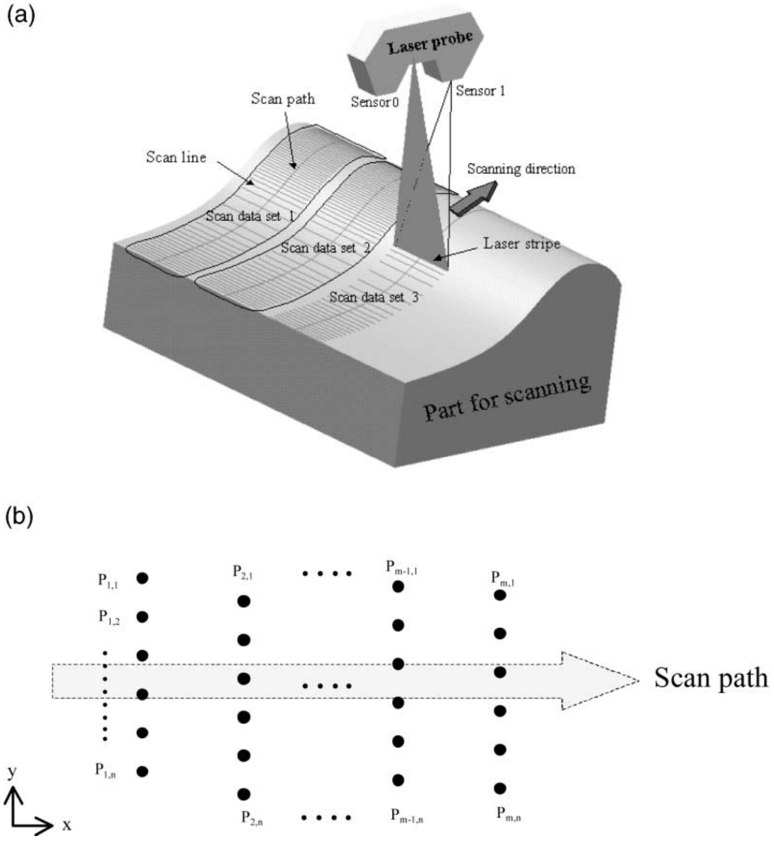

Fig. 3. A laser-scanning system. (a) Laser scanning; (b) Point data in each scan path.

the first point, $P_{i,1}$, in the (i)th line as shown in Fig. 5(a) and (b). Once the first edge is decided, the destination point for the second edge must be the next closest point to the origin, $P_{i+1,1}$. Therefore, two cases are possible when determining the second edge as shown in Fig. 5(a) and (b).

When the destination point of the second edge, $P_{i,2}$, is selected from the (i)th line as in Fig. 5(a), the second edge becomes the first edge of the next *edge-pair*. When the destination point, $P_{i+1,2}$, is selected from the (i+1)th line as shown in Fig. 5(b), a new edge that starts from the destination point of the second edge, $P_{i+1,2}$, to the destination point of the first edge, $P_{i,1}$, becomes the first edge of the next *edge-pair* so that the origin moves to the destination point of the second edge. By determining the edges with this strategy, the normal orientations constructed by *edge-pairs* can be easily arranged. After estimating the normal values of all *edge-pairs*, the point normals are calculated.

The normal value of point P is estimated from a group of *edge-pairs* that uses the point P as one of their vertices. Fig. 6 shows point P and its neighboring *edge-pairs*. The normal value of point $P_{i,j}$ and the normalized point

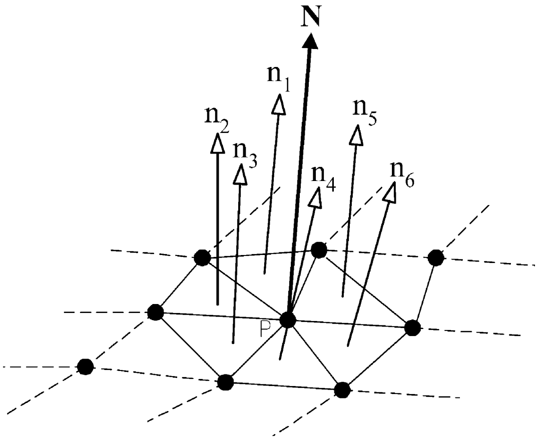

Fig. 6. Calculation of the normal value of a point. (a) Octree subdivision; (b) Tree representation for 3D-grids.

normal $\hat{\mathbf{N}}_{P_{i,j}}$ at the same point are calculated by Eqs. (1) and (2), respectively:

$$\mathbf{N}_{P_{i,j}} = \frac{\sum_{i=1}^{m} \mathbf{n}_i}{m} \tag{1}$$

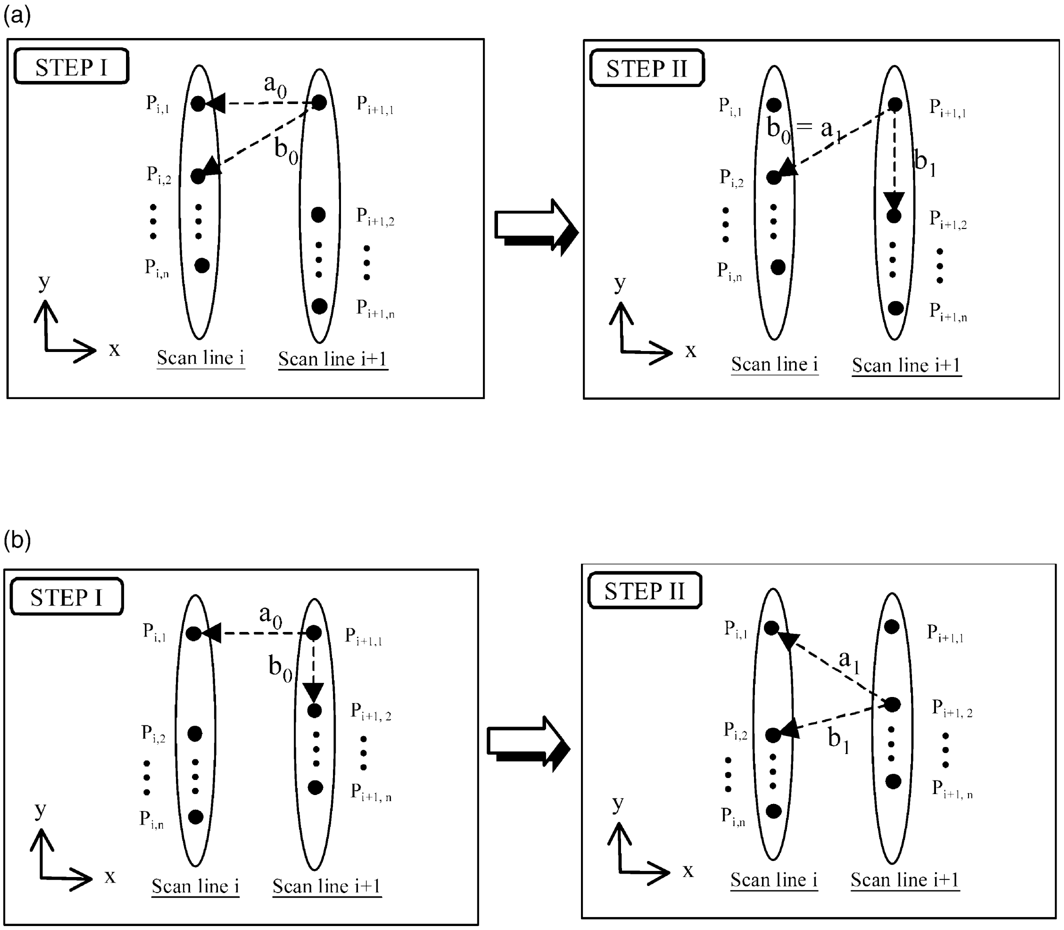

Fig. 5. Two cases of edge determination: (a) Case I; (b) Case II.

where *m* is the number of *edge-pairs*, and

$$\hat{\mathbf{N}}_{P_{i,j}}=\frac{\mathbf{N}_{P_{i,j}}}{|\mathbf{N}_{P_{i,j}}|} \tag{2}$$

After calculating the point normals for each scan data set, the normal values are stored into a point data structure, which has *x*-, *y*- and *z*-coordinates and *x*, *y* and *z* normal components. Afterwards, registration is performed to arrange all point clouds into a single coordinate system.

3.3. 3D-grid generation and subdivision

The scan data after registration keep the normal values for each point, but the points are neither ordered nor merged. This requires additional operations such as ordering, merging and cross-sectioning. These operations can be avoided by applying the octree method to the scan data. The method can be represented as a tree structure, as shown in Fig. 7. It has been widely used in the area of computer graphics and image processing [15,16]. In the octree method, each cubical octant in the pyramidal structure has eight daughters created by halving the parent cell along the *x*, *y* and *z* directions based on the criteria defined by the application. In approximating a geometric object, the octants completely inside or outside the object are not subdivided further while those octants which contain a portion of the object's boundary continue to be subdivided.

In this research, the octree method is used for segmenting the scan data. After registration is completed, a bounding box that can encase the point cloud is created and the initial grids are generated based on the bounding box. The shortest axis among the bounding box's *x*-, *y*- and *z*-axes is selected. The size of the cells is determined by dividing the shortest axis with a user-defined number. Since an initial cell is divided into eight cells during the subdivision process, initial cells in the shape of a cube are desirable.

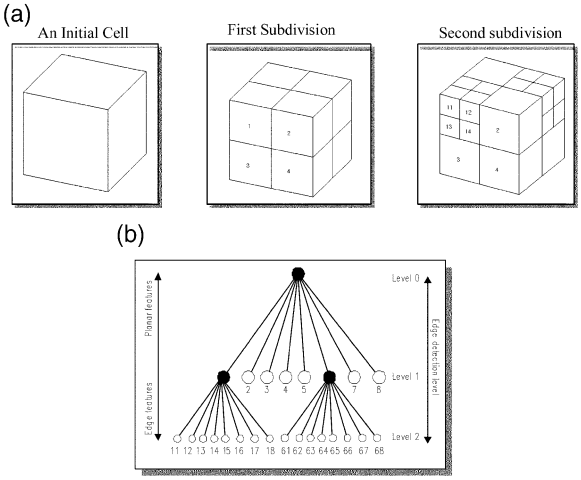

Fig. 7. The octree structure.

Fig. 8(a) shows the scan data of a part having a planar slope surface encased by a bounding box and Fig. 8(b) shows its initial cells. Among the initial cells generated, unnecessary cells, which do not contain any points, are eliminated. During the refinement process, the point model is approximated with uniform cells. For generating non-uniform cells that use different edge sizes, the normal values of the points within each cell are used. If the direction of a point normal shows a significant difference from those of adjacent points, that point is regarded as a point on an edge. These edge points in the scan data may not exactly represent an edge since they are sampled from the part surface in a discrete basis. In addition, it is difficult to select the edge points from unordered scan data. The non-uniform 3D-grid method is useful in handling a large amount of unordered sets of point data since it is applicable without arranging the data.

In subdivisioning of initial cells, the standard deviation of the point normal values within each cell is used. The average normal vector $\bar{\mathbf{N}}$ for each cell is calculated from the normalized point normals within each cell by:

$$\bar{\mathbf{N}}=\sum_{i=0}^{n}\mathbf{N}_i/n \tag{3}$$

where $\mathbf{N}_i$ is the unit normal vectors of each point in a cell and *n* is the number of points in the cell.

The standard deviation of point normal vectors is calculated by:

$$\sigma=\left|\sqrt{\frac{\sum_{i=0}^{n}(\mathbf{N}_i-\bar{\mathbf{N}})^2}{n}}\right| \tag{4}$$

$$=\sqrt{\frac{\sum_{i=0}^{n}(x_i-\bar{x})^2+\sum_{i=0}^{n}(y_i-\bar{y})^2+\sum_{i=0}^{n}(z_i-\bar{z})^2}{n}}.$$

The standard deviation of points in a cell indicates the level of changes in the shape of the part within each cell. When the standard deviation is larger than the user-defined tolerance, the cell is subdivided into eight daughter cells using an octree decomposition and empty cells are eliminated. The user, considering the part shape, predetermines the size of the tolerance. This subdivision process continues until a cell satisfies the termination conditions, such as when the standard deviation of point normals within a cell is smaller than the given tolerance or when a cell contains one point. Fig. 8(c) shows the final grids generated for the part given in Fig.

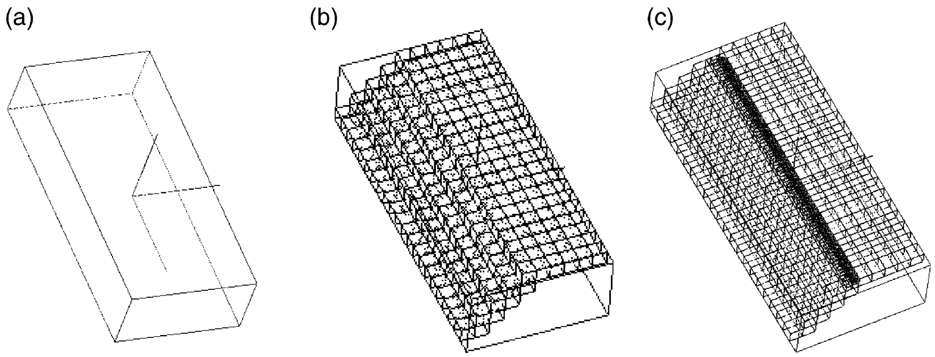

Fig. 8. 3D-grid generation by subdivisioning.

8(a). The process of 3D-grid generation is shown using the flow chart in Fig. 9.

3.4. Segmentation using the 3D-grids

Upon completion of 3D-grids by subdivisioning, the point data can be segmented using the procedure shown in Fig. 10. The sizes of the cells differ according to the shape of the object. In highly curved areas, many small cells are generated whereas fewer cells are needed in relatively planar areas. The number of iterations in the subdivisioning process increases when the detailed features of an object need to be represented. When the size of 3D cells becomes very small, these cells can correspond to the areas containing edges. The smaller the size of the selected cells, the nearer the points in the cells are to the edge of an object, as shown in Fig. 8(c).

When segmenting a point cloud, each region representing a different surface must be identified. The points in highly curved areas are extracted first by selecting smaller size cells. Removal of these cells automatically separates the point cloud into several regions by leaving gaps between different regions. The size of the 3D cells extracting the edge-neighborhood points is determined by the user-defined value. Different cells can be removed by using a different user-defined value, and this will change the shape of the edges accordingly.

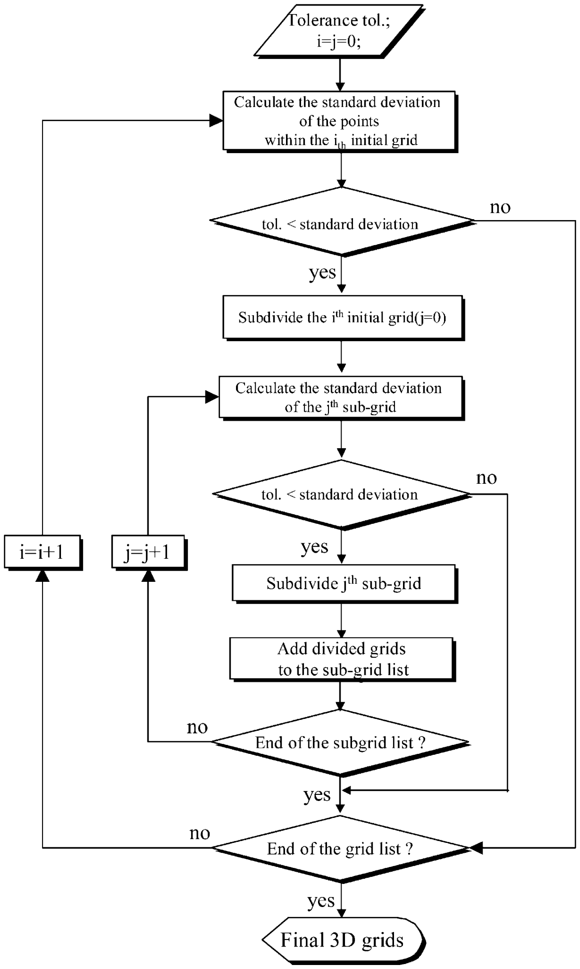
Tolerance tol.;
i=j=0;
Calculate the standard deviation of the points within the i_th initial grid
tol. < standard deviation
no
yes
Subdivide the ith initial grid(j=0)
Calculate the standard deviation of the jth sub-grid
tol. < standard deviation
no
yes
i=i+1
j=j+1
Subdivide jth sub-grid
Add divided grids to the sub-grid list
no
End of the subgrid list ?
yes
no
End of the grid list ?
yes
Final 3D grids

Fig. 9. The process of generating 3D cells.

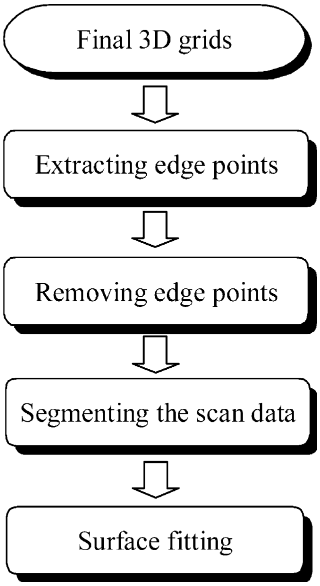

Fig. 10. The procedure of segmentation.

After removing edge-neighborhood points, the separation between regions is performed automatically by detecting any discontinuities using the gaps created by the removed cells. For computational efficiency, the octree data structure of the 3D cells is used again for the separation task. It starts with designating one cell in the leaf nodes as a seed cell. This seed cell grows up by merging adjacent cells by checking the homogeneity condition. The average normal values of points in each cell are used for evaluating homogeneity. If the difference between average normal values in two adjacent cells is within the tolerance, these two cells can be merged. The average normal values for each cell already computed during the previous subdivision were used.

In order to search the adjacent cells from the seed cell, the connectivity test is performed. If one of the vertices in a cell is shared with other cells, these are regarded as connected. The connectivity test is carried out at the parent node of cells if they do not belong to the initial grids. If two parent cells are distinct, the connection cannot be allowed among their daughter cells. However, if two parent cells are adjacent, two situations must be considered. First, if the two parent cells are adjacent without any dead daughter cells, all daughter cells under each parent node can be connected. In contrast, if two parent cells with dead daughter cells are adjacent, the connectivity test should be performed for all daughter cells. In order to merge the cells, both the connectivity and homogeneity conditions should be satisfied. The seed cells can grow through this process and they result in several different groups by merging connective and homogeneous cells.

Once the point data are segmented, appropriate surfaces can be fitted using the segmented points, and the entire surface model is constructed by extending these surfaces and finding the intersection of curves between them [17]. For industrial parts having quadric surfaces, such as planes, cylinders and cones, this method can be applied efficiently regardless of the type of point. This method directly uses the registered scan data and does not require the merging of point data from different views.

4. Application examples

The proposed segmentation method was applied to two sample parts, a test block and a cellular phone model. In acquiring the point data, a stripe-type laser scanner, Surveyor 1200 from Laser Design Inc., was used. The algorithms were written using Visual C++ 6.0 and OpenGL and run on a Pentium II 350. Before calculating the normal values of the points, outliers or spikes were removed from the initial scan data. For performing the preprocess tasks commercial software packages, Surfacer version 9.0 and DataSculpt 4.0, were used [18,19].

The size of the test block, which contains planes, cylinders and spheres, is 33 mm in height, 100 mm in width and 60 mm in depth. Fig. 11(a) shows the scan data prepared by registering the initial point clouds. Fig. 11(b) shows the normal values for each sampled point using a porcupine map. The point normals were calculated before registration.

To generate 3D non-uniform grids, initial cells with a uniform size were generated first by dividing the shortest axis of the bounding box with a user-defined number. The initial cells, after eliminating unnecessary cells that do not contain any point data, are depicted in Fig. 11(c). Once the initial cells are created, the non-uniform grids are generated iteratively using the standard deviation of the point normals in each cell.

In this example, the allowable maximum standard deviation value of point normals in each cell, 0.01, was chosen as the tolerance for subsequent subdivisions. The standard deviation of point normals in a cell physically indicates the level of shape changes of the part in the cell. Therefore, according to this user-defined value of tolerance, the number of iterations in the subdivisioning process is decided. It also determines the level of detail of features that can be recognized from the part. In this model, when the tolerance value of 1.0 is given, two iterations of subdivisioning occurs and when 0.5, five iterations of subdivisioning occurs. In order to recognize the detailed features, the tolerance value of 0.01 was chosen for the test block example and it generated nine iterations.

Fig. 11(d) shows the final results of the non-uniform grids generated by the subdivision process. The subdivided cells in this example consist of nine different levels in size. As the cell size decreases, the points in a cell are closer to the edges.

After subdivisioning of the point cloud, the points contained in the small-sized cells are extracted. Fig. 12(a) shows the points extracted by selecting the six smallest cell levels out of nine sizes. Fig. 12(b) shows the point data obtained by removing the extracted points from the initial scan data. Each patch is generated automatically performing the connectivity and homogeneity tests using the gaps generated by the removed points, as shown in Fig. 12(c). Once the segmentation process is completed, the surface fitting task is performed on these segmented point patches. Fig. 12(d) shows the surface model created from the segmented point data.

As the second example, the segmentation procedure was applied to the upper casing of a cellular phone model. The size of the model was 20 mm in height, 45 mm in width and 120 mm in depth. It contains both sharp edges and filleted edges. By adjusting the subdivided level of cells or the tolerance for the subdivisioning process, it is possible for the user to find not only sharp edges but also filleted edges.

Fig. 13(a) shows the initial scan data after the regis-

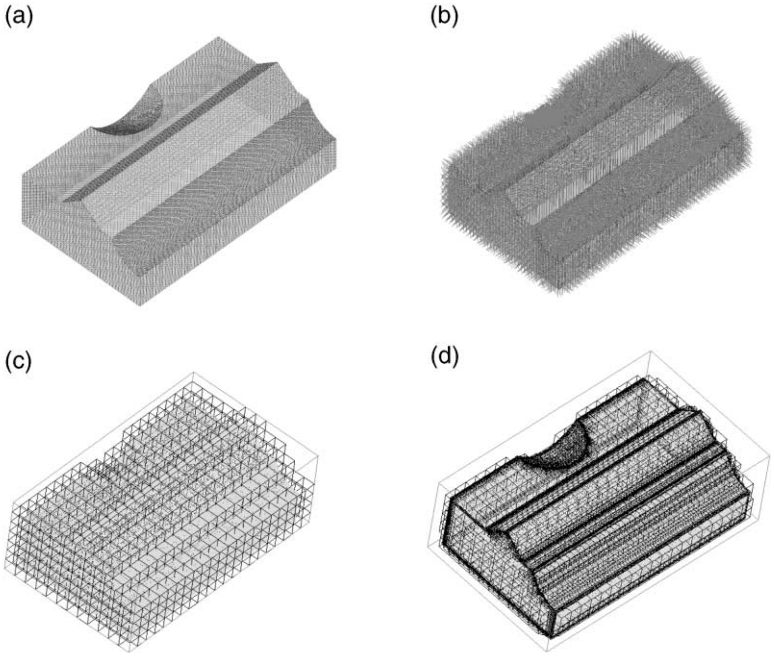

Fig. 11. Generation of the 3D non-uniform grids for the test block. (a) Initial scan data; (b) Point normals; (c) Initial grids; (d) Final 3D-grids.

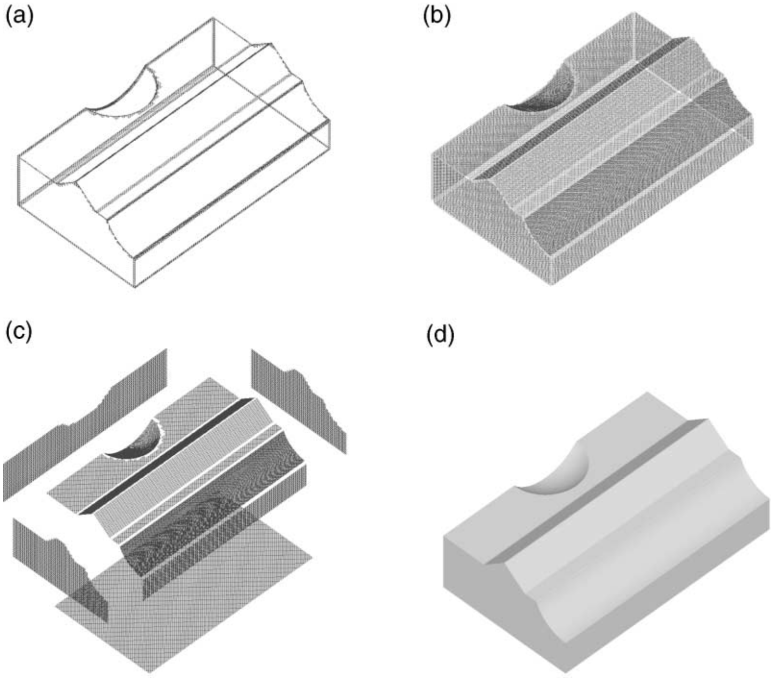

Fig. 12. Segmentation results for the test block example. (a) Extracted edge points; (b) Edge Points removed; (c) Segmented point data; (d) Surface model.

tration process. Fig. 13(b) displays the porcupine map of the normals for the sampled points in the scan data. First, the initial uniform cells are generated, as shown in Fig. 13(c), by dividing the *z*-axis into 12 cells. The subdivision of the initial grids using a standard deviation of the point normals of 0.01 results in the 3D-grid model shown in Fig. 13(d). The final 3D-grid model consists of cells with 11 different sizes. From this 3D-grid model, the sharp edges can be extracted by selecting the four smallest cell levels out of 11 sizes and the filleted edges can be extracted by selecting the fifth and sixth smallest level cells. The edge-neighborhood points including the sharp edges and the filleted edges are obtained as shown in Fig. 14(a).

The initial scan data excluding these edge-neighborhood points are shown in Fig. 14(b). Segmentation of

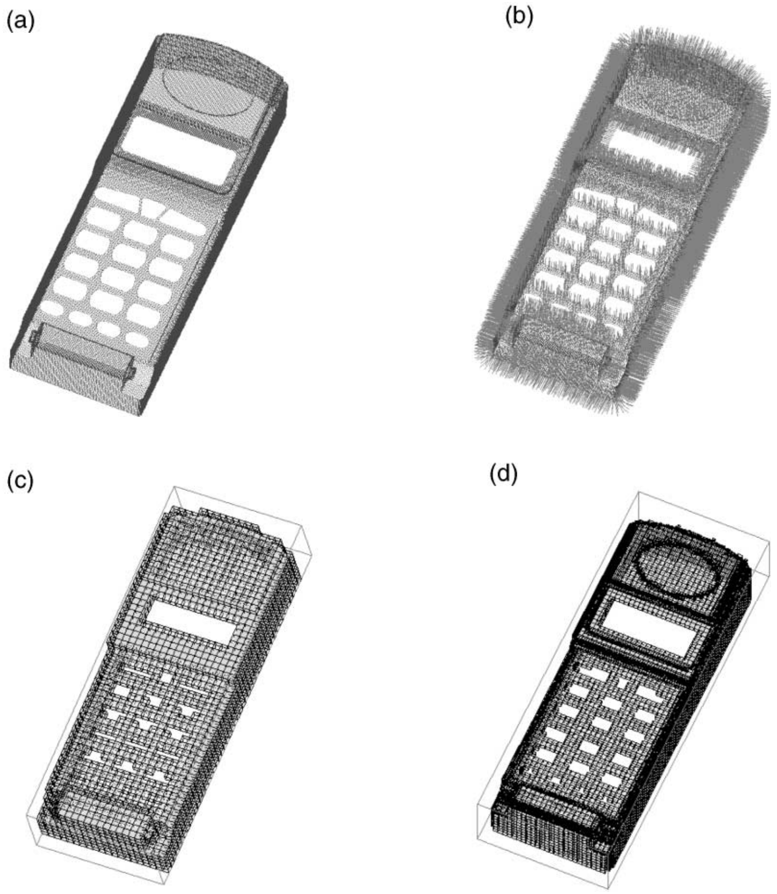

Fig. 13. Generation of the 3D non-uniform grids for the phone model. (a) Initial scan data; (b) Point normals; (c) Initial grids; (d) Final 3D-grids.

the point data was completed by the separation task using the gaps generated by removing the extracted points as depicted in Fig. 14(c). After the segmentation, each set of segmented points was fitted to a proper surface and the surface model was created by extending and intersecting these surface patches. The final surface model for the upper casing of the cellular phone model shown in Fig. 14(d) was constructed using a commercial CAD software, CATIA 4.1.9 by Dassault Systemes.

Table 1 summarizes the computing time for three examples tested in this research. The total computing time can be estimated by adding the final 3D-grid generation time to the time for segmenting the scan data into separate point clouds.

5. Conclusions

A new segmentation method using three-dimensional grids is proposed and implemented in this paper. The proposed segmentation algorithm is performed by extracting edge-neighborhood points and separating regions by the octree-based 3D-grid method. The subdivision of cells is performed iteratively using the standard deviation of point normals in each cell. As a result, fine cells are generated for the areas having drastic changes in shape.

The proposed segmentation algorithm was applied to two example models, a test block and the upper casing of a cellular phone model. The point data for the models were segmented automatically using the proposed algorithm. The surface models developed for each model were satisfactory.

The 3D grids generated in this research can be used not only for segmentation but also for other applications. The resulting cells are similar to voxels in terms of their shapes; therefore, these cells can be used for volumetric representation. The grid data can also be used to create cross-sectional slice data, which are required for fabricating parts in rapid prototyping. If the slice thickness is adjusted to be the same as the size of a cell, fine features of a part can be fabricated.

Acknowledgements

This work was supported by grant No. 2000-1-30400-006-3 from the Basic Research Program of the Korea Science and Engineering Foundation.

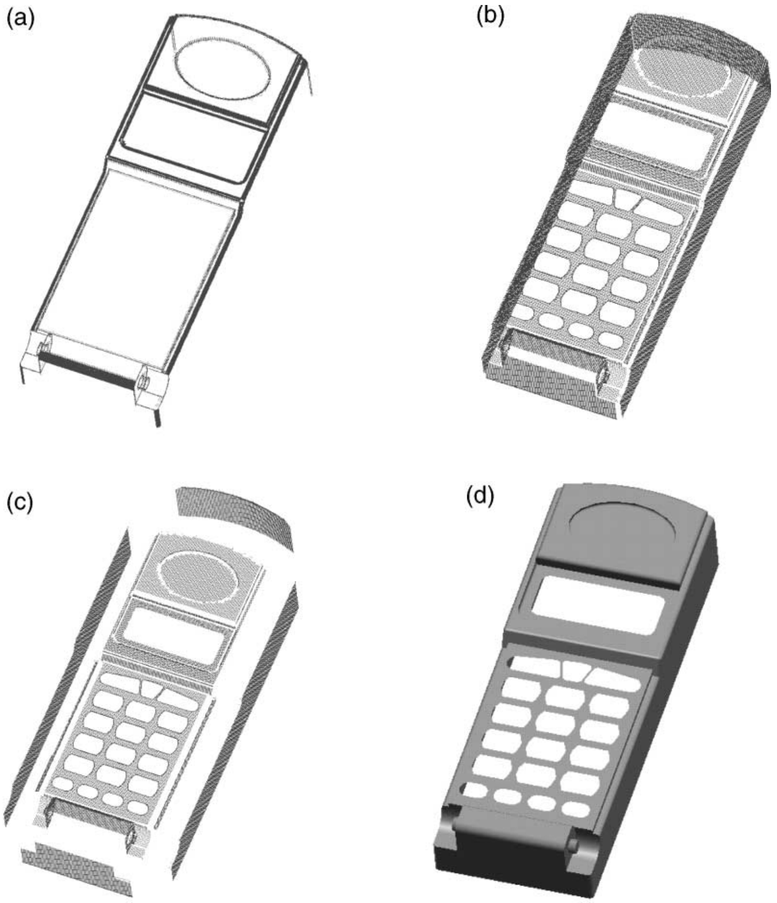

Fig. 14. Segmentation results for the phone model example. (a) Extracted edge points; (b) Edge points removal; (c) Segmented point data; (d) Surface model generation.

Table 1
The computing time of the proposed method

Model	Number of points	Number of patches	Grid generation time (s)	Segmentation time (s)
Two inclined planes (Fig. 8)	20,000	2	10	20
Test block (Fig. 11)	92,000	11	32	40
Cellular phone (Fig. 13)	215,288	32	90	120

References

[1] T. Varady, R.R. Martin, J. Cox, Reverse engineering of geometric models — an introduction, Computer-Aided Design 29 (4) (1997) 255–268.

[2] C. Tai, M. Huang, The processing of data points basing on design intent in reverse engineering, International Journal of Machine Tools and Manufacture 40 (2000) 1913–1927.

[3] H. Yau, C. Chen, R.G. Wilhelm, Registration and integration of multiple laser scanned data for reverse engineering of complex 3D models, International Journal of Production Research 38 (2) (2000) 269–285.

[4] T. Varady, P. Benko, Reverse engineering B-rep models from multiple point, in: Proceedings of Geometric Modeling and Processing, April 2000, IEEE, Hong Kong, pp. 3–12.

[5] T. Fan, G. Medioni, R. Nevatia, Segmented description of 3-D surfaces, IEEE Transactions on Robotics and Automation RA-3 (6) (1987) 527–538.

[6] Y.H. Chen, C.Y. Liu, Robust segmentation of CMM data based on NURBS, The International Journal of Advanced Manufacturing Technology 13 (1997) 530–534.

[7] M.J. Milroy, C. Bradley, G.W. Vickers, Segmentation of a wrap-around model using an active contour, Computer-Aided Design 29 (4) (1997) 299–320.

[8] M. Yang, E. Lee, Segmentation of measured point data using a parametric quadric surface approximation, Computer-Aided Design 31 (1999) 449–457.

[9] R.L. Hoffman, A.K. Jain, Segmentation and classification of range images, IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (5) (1987) 608–620.

[10] P.J. Besl, R.C. Jain, Segmentation through variable-order surface fitting, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (2) (1988) 167–192.

[11] N. Yokoya, M.D. Levine, Range image segmentation based on differential geometry: a hybrid approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (6) (1997) 643–649.

[12] P. Checchin, L. Trassoudaine, J. Alizon, Segmentation of range images into planar regions, in: Proceedings of IEEE 3D Digital Imaging and Modeling, IEEE, Ottawa, Canada, 1997, pp. 156–163.

[13] D. Zhao, X. Zhang, Range-data-based object surface segmentation via edges and critical points, IEEE Transactions in Image Processing 6 (6) (1997) 826–830.

[14] M. Laszlo, Computational Geometry and Computer Graphics in C++, Prentice Hall, New Jersey, 1996.

[15] M.D. Berg, M.V. Kreveld, M. Overmars, O. Schwarzkopf, Computational Geometry — Algorithms and Applications, Springer, New York, 1997.

[16] R. Jain, R. Kasturi, B.G. Shunck, Machine Vision, McGraw-Hill, Singapore, 1995.

[17] P.N. Chivate, N.V. Puntambekar, A.G. Jablokow, Extending surfaces for reverse engineering solid model generation, Computers in Industry 38 (1999) 285–294.

[18] Surfacer User's Guide, Imageware Inc, USA, 1997.

[19] DataSculpt User/s Manual Version 4.0, Laser Design Inc, USA, 1995.