ELSEVIER

Computer-Aided Design 33 (2001) 183–193

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Cloud data modelling employing a unified, non-redundant triangular mesh

W. Sun[a], C. Bradley[b,*], Y.F. Zhang[a], H.T. Loh[a]

[a]*Department of Mechanical and Production Engineering, National University of Singapore, Singapore, Singapore*
[b]*Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada V8W 3P6*

Received 20 May 1999; received in revised form 27 March 2000; accepted 25 April 2000

Abstract

This paper describes an application of error-based triangulation to very large sets of three-dimensional (3D) data. The algorithm is suitable for processing data collected by machine vision systems, co-ordinate measuring machines or laser-based range sensors. The algorithm models the large data sets, termed cloud data, using a unified, non-redundant triangular mesh. This is accomplished from the 3D data points in two steps. Firstly, an initial data thinning is performed, to reduce the copious data set size, employing 3D spatial filtering. Secondly, the triangulation commences utilising a set of heuristic rules, from a user defined seed point. The triangulation algorithm interrogates the local geometric and topological information inherent in the cloud data points. The spatial filtering parameters are extracted from the cloud data set, by a series of local surface patches, and the required spatial error between the final triangulation and the cloud data. Two procedures are subsequently employed to enhance the mesh: (i) the edges of mesh triangles are adjusted to produce a mesh containing approximately equilateral triangles; and (ii) mesh edges are aligned with the boundaries present on the object to minimise smoothing of naturally occurring features. Case studies are presented that illustrate the efficacy of the technique for rapidly constructing a geometric model from 3D digitised data.

Keywords: Triangulation; Polyhedral mesh; Reverse engineering

1. Introduction

Rapid product development (RPD) refers to recently developed technologies that assist manufacturers and designers in meeting the demands of reduced product development time. For example, injection-moulding companies must now be equipped to compete in the so-called niche markets where product volumes are small and lead times are short. The time available to transform a stylist's product model into the injection mould and die tooling has diminished. Reverse engineering utilising 3D-vision technology and associated data modelling software is a RPD technology currently under investigation. The flowchart depicted in Fig. 1 shows the major steps occurring in the reverse engineering process. The stylist's design model is digitised (using the vision system) and the resulting cloud data is input to an appropriate modelling package. The software creates a suitable geometric model necessary for the manufacture of the tooling. Typically, the geometric model is compatible with the requirements for a rapid prototyping system or CNC machine tool.

A stylist's model can be digitised by manual methods, such as a touch probe mounted on a CMM. This is a time consuming process both for full-size models and free-form shapes that may require thousands of data points to adequately define the surface. Machine vision, as shown in Fig. 2, is an ideal alternative for rapidly digitising the model's form. The design model is comprehensively digitised by the vision system to produce a large cloud data set. The cloud data set shown has been "thinned" from the 150,000 original (x,y,z) points to 5000 in order to improve visualisation. Advanced machine vision sensors allow high resolution "range images" of complex shapes to be rapidly acquired (compared to a CMM) and find applications in manufacturing, virtual prototyping, animation and biomedical engineering.

2. Literature review

Philosophies for modelling machine vision generated data have followed several fundamental directions [1]: (a) reconstruction of surfaces utilising an implicit function such as a parametric [2] function; or (b) surface modelling employing a polyhedral mesh [3]. The former approach has tended to apply the segment-and-fit method discussed by Hoffman and Jain [4]. The cloud data is segmented into

* Corresponding author. Tel.: +1-250-721-6031; fax: +1-250-721-6051.
E-mail address: cbr@me.uvic.ca (C. Bradley).

0010-4485/01/$ - see front matter © 2001 Published by Elsevier Science Ltd.
PII: S0010-4485(00)00088-9

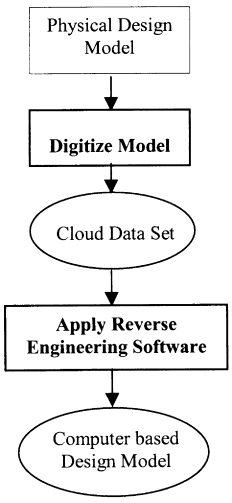
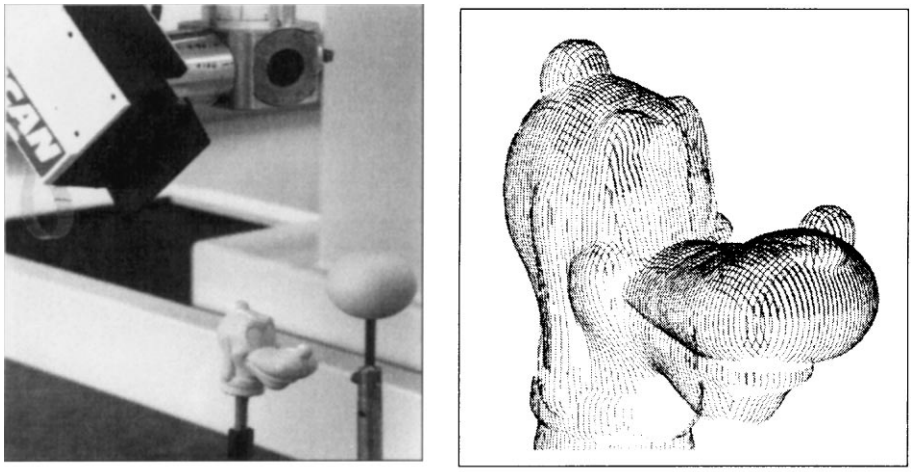


Fig. 1.

bounded patches each representing a discrete surface region found on the physical object. Data segmentation, accomplished either manually or through software, defines the patch boundary curves and produces a patchwork of surface regions [5]. Data modelling methods, such as those employing parametric [6] or quadric [7,8] functions are applied to fit appropriate surfaces to the data patches. Non-uniform rational B-spline (NURB) curves and surfaces are a current research topic due to their ability to accurately approximate most types of surface entity encountered in design and manufacturing applications [9].

Data modelling employing polyhedral meshes can be classified depending on the inherent data structure produced by the vision system. The structure can vary between highly organised arrays of points to the minimal structure found in the cloud data. Machine vision systems tend to imbue the data with some degree of structure. For example, vision systems utilising a photosensitive array (e.g. a 2D-charge coupled device) and structured illumination source produce a range image composed of a regular grid of data points. Therefore, a polygonal model can be created in a straightforward manner by linking data points in a neighbourhood to form the mesh. If an object has been digitised through the acquisition of multiple range images, then an appropriate registration and alignment technique must be implemented to merge the adjoining polygonal domains. Soucy and Laurendeau [10] describe a method using a Venn diagram to identify the overlapping data regions, which is followed by a re-parameterisation and merging of regions. Turk and Levoy [3] devised an incremental algorithm that updates a data reconstruction by eroding redundant geometry and zippering along the remaining boundaries. Finally, a consensus step that re-introduces the original geometry to establish final vertex positions is used. Algorithms dependent on a pre-existing data structure typically perform with greater efficiency than unconstrained algorithms employing unstructured data. However, a major disadvantage of these algorithms is the inherent dependence on specific sensor type or manufacturer.

Algorithms that have been developed for modelling less structured 3D data sets assume that no a priori information regarding the connectivity of points in the data set is available. Furthermore, other algorithms assume no restriction on either the convexity or differentiability of the underlying surface. The only assumption is that there exists a sufficiently high data sampling resolution to permit unambiguous reconstruction of the model. The two-dimensional Delaunay triangulation has been extended to three dimensions by Fang and Piegl [11], for example, who also provided implementation details. Another Delaunay triangulation technique, based on a divide-and-conquer paradigm, was proposed by Cignoni et al. [12]. Lawson [13] used geometric reasoning to construct a triangular facet



(a) (b)

Fig. 2.

mesh and subsequently, Choi et al. [14] extended the same method using a vector angle order instead of Euclidean distance to determine the linkage of the data points. Recently, Hoppe et al. [15] developed a signed distance function by estimating the local tangential plane and using a marching cube method to extract a triangular polyhedral mesh.

The method, described in this manuscript, addresses the following issues:

- the construction of a triangular polyhedral mesh from a copious and unstructured set of 3D cloud data;
- the removal of all assumptions concerning the underlying surface form and the structure of the data produced by the sensor;
- the control of the accuracy, between the triangulation and the original cloud data set, to within a pre-defined tolerance.

The algorithm is developed based on a set of heuristic rules. The mesh grows by interrogating the geometric and topological information in the neighbourhood of the candidate points through application of the heuristic rules in order to connect a new point to the boundary of the expanding mesh. The mesh growth commences from a "seed" point that is selected from a patch centre in the original cloud data file. This is accomplished using the user interface of the software package developed in the course of the work. A spatial filtering method is used to reduce the original cloud data file size. The cut-off of the spatial filter is adjustable and permits enough cloud data to be retained so that the reconstruction meets the error criterion. This is controlled through the determination of the maximum permissible triangular facet edge length based on a local model of the underlying surface.

3. Cloud data acquisition

A non-contact laser-based range sensor was used to collect the cloud data from the object surface. The range sensor (Hymarc, Hyscan 45C) operates on the optical triangulation principle and utilises a unique synchronised scanning mechanism that provides an effectively wide baseline separation (between source and detector) but encloses all the hardware components in a physically compact head. The accuracy specification of the cloud data is ± 0.025 mm, over a 100-mm depth of field, and a stand-off distance of 100 mm. The sensor scans a laser spot over an object's surface, in a continuous high-speed manner, and gathers data at a maximum rate of 10,000 points per second. The sensor is integrated with a computer controlled CMM as shown in Fig. 2a. The head is positioned and translated (during the scanning process) by the CMM control system and the surface points are captured during the movement of the sensor head. The sensor's small field of view and the object's form can often necessitate multiple scanning passes to fully digitise the object. However, all the data sets, from each separate scanning pass, are combined into one global data file referenced to a single point. Typical working specifications for the sensors are: 180 data points acquired over the 80-mm span of the scan line. The scanner is traversed at a uniform speed of 1.5 mm per second, yielding scan lines that are spaced 0.5 mm apart. A sequence of points, acquired along each scan line does constitute a modicum of data structure. However, a typical digitisation process requires the range sensor to be re-positioned six-to-eight times, at various positions and viewing angles, to completely sample the surface patches. The resulting cloud data file is a mosaic of range data patches as shown in the images of the cloud data collected from the model in Fig. 2b.

4. Data modelling based on an error criterion

The cloud data set, shown in Fig. 3, is a cloud data file of a digitised face mask and contains 104,216 points. A triangulation-based modelling scheme does not necessarily require such a high density of surface points in order to construct an accurate mesh. Extremely large size cloud data sets could produce long computation times, exceed memory storage limits and generate computational stability problems. However, arbitrary removal of cloud data points, based on simple data thinning methods [5,8], may not ensure the necessary accuracy between the triangulation and the original data is achieved. Therefore, the cloud data reduction methods are required that do satisfy any pre-defined error specification. Thus, the method must retain sufficient cloud data to satisfy the error specification between the triangulation and the original data. Several algorithms, such as the PM algorithm developed by Hoppe [16] and Eck et al. [17], reduce the polyhedral mesh size by means of subdivision connectivity. However, this technique is not viable for unstructured cloud data.

The method developed in this research applies voxel binning [5], to the cloud data, as an initial step. The bin size is first calculated based on a required error tolerance between the original cloud data and the final triangulation. The overall steps for computing the triangulation are presented below:

- Compute bounding box for the cloud data points:
  $\{x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max}\}$
- Offset the bounding box limits outward by a point coincidence tolerance (TOL):

$$x_{\min} = x_{\min} - \mathrm{TOL}$$

$$x_{\max} = x_{\max} + \mathrm{TOL}$$

$$y_{\min} = y_{\min} - \mathrm{TOL}$$

$$y_{\max} = y_{\max} + \mathrm{TOL}$$

$$z_{\min} = z_{\min} - \mathrm{TOL}$$

$$z_{\max} = z_{\max} + \mathrm{TOL}$$

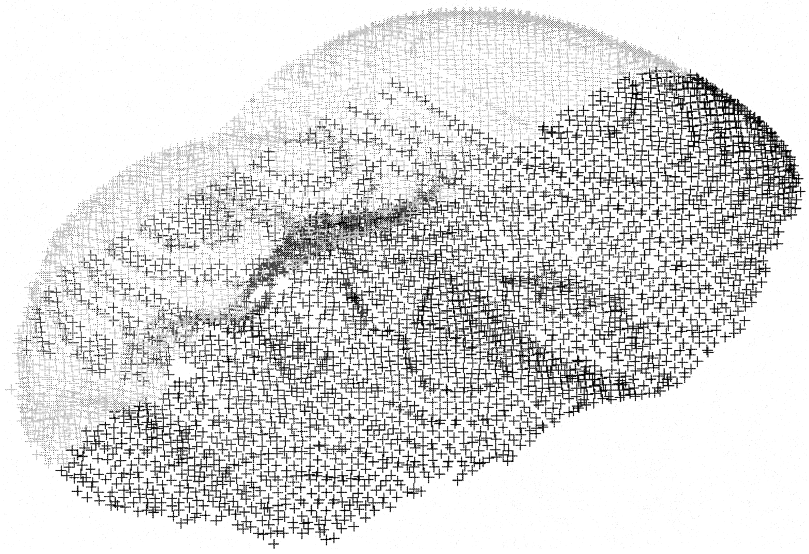


Fig. 3.

This step resolves conflicts when points lie on cell planes.

- Sample N points from the cloud data file and interpolate a local parabolic quadric surface patch to the N points by means of least squares fitting.
- Calculate the maximum second partial derivatives from the quadric function coefficients. The derivative values are used to calculate the maximum triangle edge length (Ω) for a desired error tolerance ϵ between the cloud data and triangulation.
- Determine maximum triangle edge length Ω, from the derivative values and ϵ.
- Compute the bin cell size (B), corresponding to ϵ, from the following expression:

$$B \leq 2\sqrt{\frac{\epsilon}{(M_1 + 2M_2 + M_3)}} \tag{1}$$

where, M_1, M_2 and M_3 are the maximum partial derivative values of the locally interpolated surface.

- Apply voxel-binning data thinning to the cloud data using bins of dimension B.
- Construct the final triangular polyhedral mesh to the thinned data set.

The details of the key steps in the above algorithm are presented in the following sections.

4.1. Local surface patch interpolation

The maximum values of the second partial derivatives M_1, M_2 and M_3 are estimated from a second order quadric surface that is interpolated to a set of points $\mathbf{P}$ randomly sampled in the locality of the central point P_j. Experiment revealed that a small set of 24–32 points was an effective trade-off between computation time and sufficient data to form a reasonable interpolation to the underlying surface form. The quadric approximation is given by

$$\mathbf{S}(u, v) = au^2 + buv + cv^2 \tag{2}$$

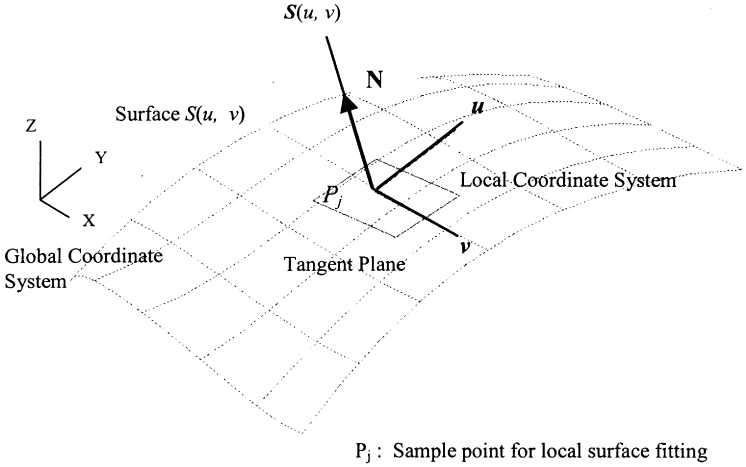


Fig. 4.

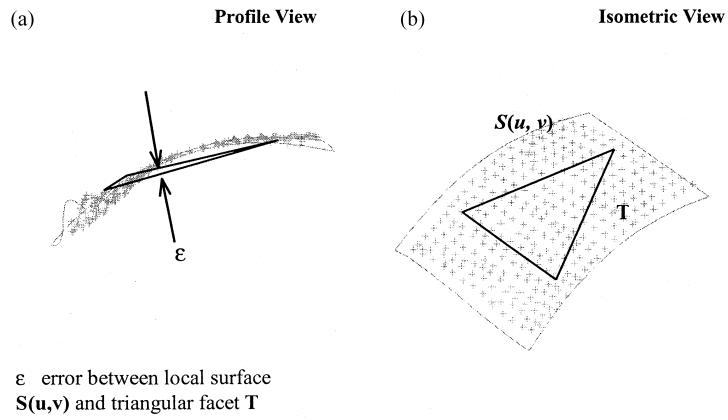


Fig. 5. (a) Profile view. (b) Isometric view.

The local co-ordinate system employed for the quadric surface is defined in Fig. 4. The origin of the co-ordinate system frame (u,v,**S**) is placed at the point P_j and the **S** axis aligned with the local surface normal **N** as shown. The surface normal is obtained from an initial planar fit to the same local set of cloud data points. By definition, the gradient at the local origin, with respect to the parameters u and v is zero. The direction of the u and v axes can then assume an arbitrary direction in the plane perpendicular to **N**. The coefficients (a,b,c) of function S are determined from a least squares fit to the local data [18]

$$\mathbf{X} = [\mathbf{A}^T\mathbf{A}]^{-1}\mathbf{A}^T\mathbf{B} \tag{3}$$

where the function parameters are collected in the vector **X**

$$\mathbf{X} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \dfrac{\partial^2 S}{\partial u^2} \\ \dfrac{\partial^2 S}{\partial u \partial v} \\ \dfrac{\partial^2 S}{\partial v^2} \end{bmatrix} \tag{4}$$

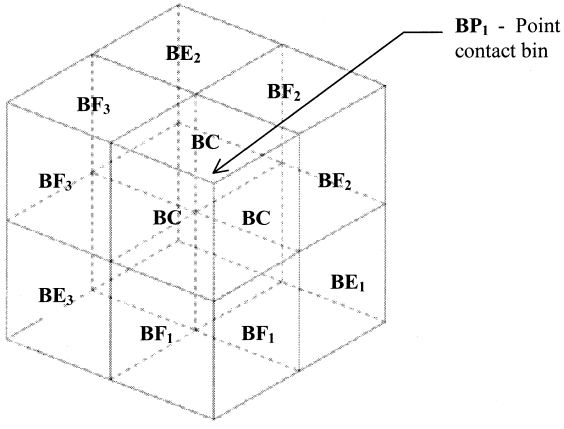


Fig. 6.

and the vectors **A** and **B** are given by

$$\mathbf{A} = \begin{bmatrix} u_1^2 & u_1v_1 & v_1^2 \\ u_2^2 & u_2v_2 & v_2^2 \\ \vdots & \vdots & \vdots \\ u_n^2 & u_nv_n & v_n^2 \end{bmatrix} \tag{5}$$

$$\mathbf{B} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}. \tag{6}$$

4.2. Calculation of the maximum edge length of the triangulation

The maximum partial derivatives of $\mathbf{S}(u,v)$ are used in conjunction with the error tolerance, ϵ, to determine the maximum permissible triangular facet edge length, Ω. For the surface $\mathbf{S}(u,v)$ in C^2 interpolated by point set $\mathbf{V}_{ij}$ and an arbitrary linear triangle **T** with vertices $(A, B, C) \in \mathbf{V}_{ij}$, the deviation between the triangular facet and the surface patch satisfies [19,20]

$$\sup_{(u,v)\in T} \|S(u,v) - T(u,v)\| \leq \tfrac{2}{9}\Omega^2(M_1 + 2M_2 + M_3) \tag{7}$$

where

$$M_1 = \sup_{(u,v)\in T} \left\| \frac{\partial^2 S(u,v)}{\partial u^2} \right\| \tag{8}$$

$$M_2 = \sup_{(u,v)\in T} \left\| \frac{\partial^2 S(u,v)}{\partial u \partial v} \right\| \tag{9}$$

$$M_3 = \sup_{(u,v)\in T} \left\| \frac{\partial^2 S(u,v)}{\partial v^2} \right\| \tag{10}$$

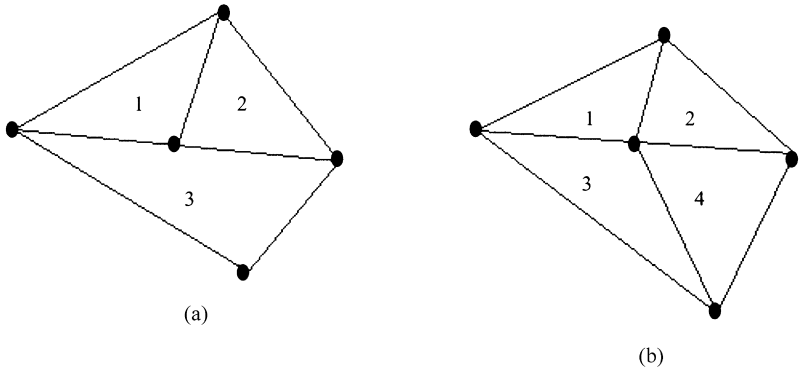


Fig. 7. (a) Incorrect triangulation — vertex lies on the edge of another triangle. (b) Correct triangulation.

Combining Eqs. (7)–(10) with the specified error tolerance ϵ (Fig. 5 presents a graphical interpretation) gives the maximum permissible triangle edge

$$\Omega = 3\sqrt{\frac{\epsilon}{2(M_1 + 2M_2 + M_3)}} \tag{11}$$

Eqs. (7)–(11) have been applied to the problem of tessellating surfaces defined by parametric B-spline or NURB surfaces. However, Eq. (7) can be improved with a sharper bound to give

$$\sup_{(u,v)\in T} \|S(u,v) - T(u,v)\| \leq \tfrac{1}{8}\Omega^2(M_1 + 2M_2 + M_3) \tag{12}$$

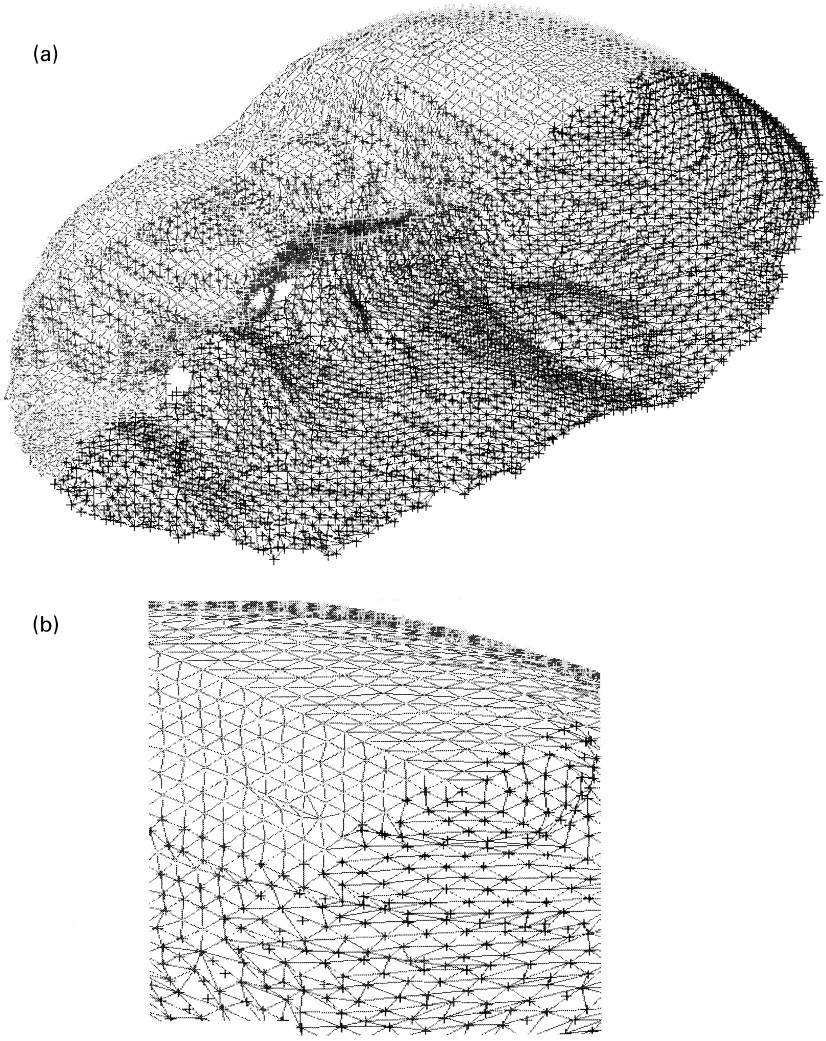


Fig. 8.

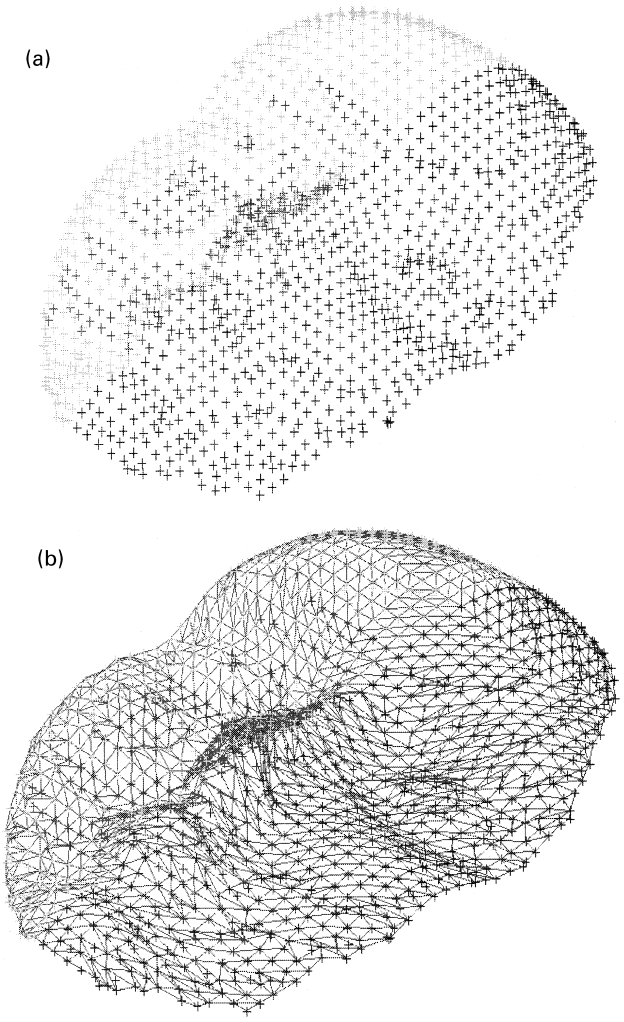


Fig. 9.

The proof of Eq. (12) is outlined in Appendix A. The refined expression for the maximum edge length of the triangulation using Eq. (12) is

$$\Omega = 2\sqrt{\frac{2\epsilon}{(M_1 + 2M_2 + M_3)}}. \tag{13}$$

4.3. Calculation of the bin size from the maximum triangulation edge length

The maximum triangle edge length determines the maximum bin size. The polyhedral triangulation is formed by meshing the points remaining at the centre of each bin. Therefore, the maximum point-to-point distance, Ω, influences the bin size B. Every bin has 26 adjoining bins that could possibly contain a point inside, as shown in Fig. 6. With reference to the diagram, there are eight bins indicated as **BP** in the figure that has one-point contact with the central bin **BC**. There are a further 12 bins, **BE**, that each have contact along one edge with **BC** and another six bins, **BF**, that contact across a full bin face with **BC**. The cloud data is very dense and it is assumed that the surface could pass through two neighbouring bins in the following ways: (a) through bins sharing a common face (**BF**); (b) obliquely pass through two neighbouring cells (i.e. at 45°) that share a common edge (**BE**); and (c) through bins that share a point (**BP**). Therefore, the maximum linking distance $l_{\max}$ between two points in neighbouring cells is given by

$$B = \frac{\sqrt{3}}{3} l_{\max} \tag{14}$$

where, B is the bin dimension. For the triangulation method described below, the maximum length of a triangular facet edge is equivalent to the maximum distance $l_{\max}$ linking neighbouring points

$$l_{\max} = \Omega \tag{15}$$

Combining Eqs. (14) and (15), the maximum bin size is derived as

$$B \leq \tfrac{2}{3}\sqrt{\frac{6\epsilon}{(M_1 + 2M_2 + M_3)}} \tag{16}$$

The data thinning process proceeds using the computed voxel bin size. Typically, multiple data points will fall within a single voxel and, therefore, each bin is examined to determine which point should be retained. The retained point is closest to the voxel's centre, thereby reducing the cloud data to a more uniformly spaced set. In effect, voxel binning reduces the influence of the digitiser, and digitising process, on the resulting surface model.

5. Generation of the polyhedral triangular mesh

The objective of mesh generation is to cover the reduced point set with a single surface mesh of triangular facets, M_i. The patch growth commences from an initial or "seed" cloud data point. The seed point is located at the approximate centre of a surface patch and the polyhedral triangular mesh expands outward until all the points in the patch are included. The meshing process is initiated with the selection of a seed point and all the remaining points form the set of valid meshing points. The set of points is sorted in increasing the Euclidean distance from the seed point, forming a sorted subset. The mesh is formed by joining the seed point to the nearest neighbours based on the selection rules described below. The process is repeated until all the points in the set are meshed. Data, which are successively included in the mesh, are removed from the initial set before the next element of the mesh is constructed. The work described herein has been restricted to the meshing of surfaces comprised of a single patch in order to aid the clarity of the description of the method. The method can easily be extended to mesh multiple patch surfaces [21].

Polyhedral mesh growth rules have been developed to prevent invalid triangular facets from becoming part of

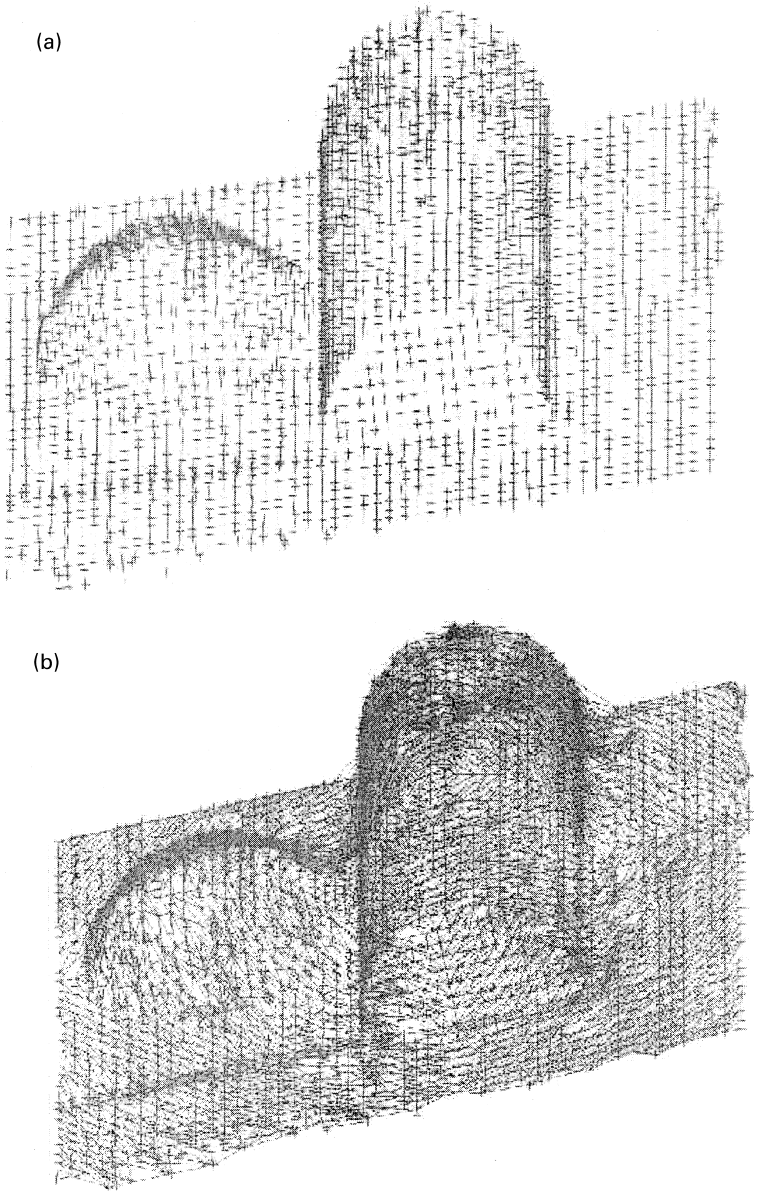


Fig. 10.

the mesh. The rule set is:

1. *Maximum point-to-point distance.* The triangular facet edge length cannot be longer than Ω. This criterion limits the searching distance for the next point.
2. *Minimum triangular facet angle.* The smallest angle within any triangular facet must be larger than the minimum angle parameter, θ; typically, $\theta = 12°$. This prevents nearly co-linear vertices from forming long, narrow triangular facets. Such facets are unwanted because their surface normal may not accurately match the object's local surface curvature.
3. *Incorrect vertex location.* One triangular facet edge can only be shared by a maximum of two other adjacent triangles. This eliminates situations, as illustrated in Fig. 7, where the vertex of one triangle lies on the edge of another triangle.
4. *Meshing to boundary points.* A candidate cloud data point can only be meshed to vertices on the boundary of the current "expanding" polyhedral mesh.
5. *Cloud data visibility.* A candidate cloud data point can only be linked to edges that are visible to it (the visibility of a vertex to an edge is defined below).

The boundary edges of the triangulated mesh are sorted based on the distance to a new vertex and those edges form a loop enclosing the meshed area. The edge list is shortened by the heuristic rule #1; i.e. only those boundary edges within Ω are retained. Two lines are projected and linked to the two vertices of the closest edge to the new vertex. Any

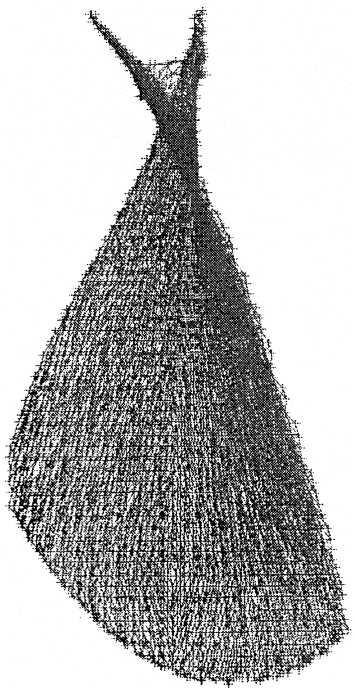


Fig. 11.

edges in the edge list that are crossed by the projecting lines will be considered as "shadowed" by the closest edge and deleted from the edge list. The process is repeated for the available second closest edge and the process recursively repeated so that the edge list only contains edges visible to the new vertex. Implementation of the data thinning and triangulation was performed in the Unigraphics environment employing its internal programming tools.

6. Application examples

Milling, using CNC methods, can be performed from the surface definition generated by this technique. For example, many commercially available computer-aided manufacturing software packages can utilise the polyhedral surface representation. Free form surface tool path generation methods [22], employing non-isoparametric means, are ideally suited to the polyhedral triangular facet model. In particular, researchers investigating tool-part collision and gouge-free machining of free form surfaces employ this description. Three case studies are presented to show the utility of this technique for modelling complex free-form parts.

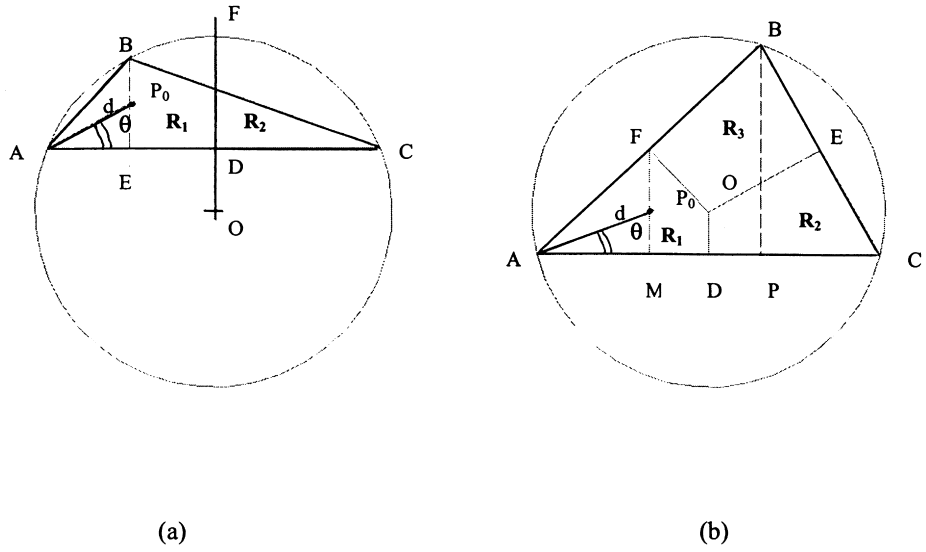
The first case study is of the mask, shown in Fig. 3, which is composed of four range data patches referenced to a single datum point on the bed of the CMM. The face mask is an intricate free form surface occupying a volume of 250 mm × 200 mm × 60 mm and the surface possesses smooth regions interrupted by sharp features with high surface curvature. The surface is problematic for B-spline surface fitting due to the difficulties inherent in knot vector creation, data point parameterisation, and control point location. The binning and triangulation algorithm was applied to the original data file containing 104,216 points. Employing an error tolerance of 0.4 mm a commensurate bin dimension of 3.0 mm was determined. Applying this spatial binning criterion to the data, and meshing the result, the polyhedral model shown in Fig. 8a was obtained. This triangulation contains 5410 vertices and a close up of the resulting mesh is shown in Fig. 8b. As shown in Fig. 9a, the application of an error criterion of 1.3 mm resulted in a bin dimension of 6.0 mm. Only 1441 points remained after data thinning and the result was meshed as shown in Fig. 9b.

The second case study is a polyhedral object composed of a cylindrical and spherical patch lying on a plane, as shown in Fig. 10a. The original object occupies a volume of 150 mm × 150 mm × 30 mm and was digitised to produce a cloud data file of size 89,197 points. The spatial binning algorithm was applied to the cloud data employing an error tolerance of 0.5 mm. This resulted in a commensurate bin dimension of 3.0 mm. The thinned data set, illustrated in Fig. 10a, contains 3163 points and the final triangular mesh after optimisation is shown in Fig. 10b. A second trial applied an error criterion of 0.75 mm to the spatial binning process and resulted in a bin dimension of 6 mm and a resulting data set of 799 points.

The third case study is of a turbine blade digitised to create five range data patches. The turbine blade is a relatively simple free form surface occupying a volume of 240 mm × 180 mm × 65 mm. The data thinning and triangulation algorithm were applied to the original data file containing 76,496 points. A bin dimension of 3.0 mm was obtained from the application of an error tolerance of 0.2 mm to the cloud data. The spatial binning and meshing phases of the process resulted in the polyhedral model shown in Fig. 11. This triangulation contains 4870 vertices.

Table 1
Summary of the modelling results for each example

	Original cloud data size	Error tolerance (mm)	Maximum triangle edge length (mm)	Bin size (mm)	Number of mesh vertices	Numbers of bad triangles
Face mask, Trial I	104,216	0.4	4.2	3.0	5410	9
Face mask, Trial II	104,216	1.3	8.5	6.0	1441	0
Polyhedral object, Trial I	89,197	0.5	4.2	3.0	3163	4
Polyhedral object, Trial II	89,197	1.0	8.5	6.0	799	11
Turbine blade	76,496	0.2	4.2	3.0	4870	17



(a) (b)

Fig. A1.

A complete summary of the algorithm's performance, for these three examples, is given in Table 1. An estimate is also provided in Table 1 for the number of incorrect triangular facets produced by the algorithm. An incorrect facet is the one that does not obey the set of heuristic rules created to govern the meshing process. Problematic cloud data sets can occur and it is not uncommon to discover incorrect tesselations.

7. Conclusion and future work

A practical method for achieving surface construction from 3D cloud data, with accuracy control, has been described. The method commences with an analysis of the deviation between the polyhedral mesh and the underlying surface produced for a given triangular facet edge length. This information is used in determining the voxel bin size employed in the data thinning process. A triangulation algorithm based on geometric and topological constraints is then applied to the reduced cloud data set to generate the polyhedral mesh. Three examples have been described to show the efficacy of the method.

Currently the triangulation algorithm is well behaved when applied to digitised data collected from surfaces such as the face mask and turbine blade. Surfaces possessing regions of very high curvature and complex topology will require the development of a more sophisticated algorithm. Modifications to accommodate objects with holes and sharp physical edges (as found in manufacturing and biomedical application) are under development. A technique to detect and amend incorrect triangular facets will also be considered.

Appendix A

Proof. The proof is similar to the method in Filip et al. [23]. Let $e(u,v) = S(u,v) - T(u,v)$, where $|e(u,v)|$ has its maximum value at a point P_0 that falls within a region of the triangle $T(u,v)$. Let $d = |A - P_0|$ and θ be the angle between AC and AP_0. Here, the triangle is treated for two conditions: an obtuse triangle or an acute triangle, as shown in Fig. A1a and b .

For an obtuse triangle, as shown in Fig. A1a, the triangle region is divided into two regions R_1 and R_2 created by the intersection of edge AC (of length Ω) and perpendicular line FD from the centre O of the inscribing circle. If P_0 is in the interior of R_1 it can be shown that $d\cos\theta \le AD = \Omega/2$; $d\sin\theta \le BE \le FD$ and $FD \le AC/2 = \Omega/2$. Therefore, $d\sin\theta \le \Omega/2$.

For an acute triangle, as shown in Fig. A1b, the triangle is divided into three regions R_1, R_2 and R_3 by lines FO, OD, OE. The following line-pairs are mutually perpendicular: FO and AB, OD and AC, OE and BC. Line AC is the longest edge of the triangle. If P_0 is in the interior of R_1, the other case being proved similarly, $d\cos\theta \le AD = \Omega/2$; $d\sin\theta \le FM = BP/2$ and $BP < BC < AC$. Therefore, $d\sin\theta \le AC/2 = \Omega/2$.

The following identities apply to both cases: $d\cos\theta \le \Omega/2$ and $d\sin\theta \le \Omega/2$. Following the proof in Filip et al. [23]

$$\|e(P_0)\| \le \tfrac{1}{2}(d^2\cos^2\theta M_1 + 2d^2\sin\theta c\cos\theta M_2 + d^2\sin^2\theta M_3) \quad \text{(A1)}$$

As $d\cos\theta \le \Omega/2$ and $d\sin\theta \le \Omega/2$, Eq. (A1) can be reformulated as

$$\|e(P_0)\| \le \tfrac{1}{2}\left(\tfrac{1}{4}\Omega^2 M_1 + 2\tfrac{1}{4}\Omega^2 M_2 + \tfrac{1}{4}\Omega^2 M_3\right) = \tfrac{1}{8}\Omega^2(M_1, 2M_2 + M_3). \quad \text{(A2)}$$

□

References

[1] Curless B, Levoy M. A volumetric method for building complex models from range images. Proceedings of SIGGRAPH'96, 1996, p. 303–12.

[2] Sakar B, Menq CH. Smooth surface approximation and reverse engineering. Computer Aided Design 1991;23(9):623–8.

[3] Turk G, Levoy M. Zippered polygon meshes from range images. Proceedings of SIGGRAPH'94, 1994, p. 351–58.

[4] Hoffman R, Jain K. Segmentation and classification of range images. IEEE Pattern Analysis and Machine Intelligence 1987;9(5):608–20.

[5] Weir DJ, Milroy M, Bradley C, Vickers GW. Reverse engineering physical models employing wrap-around B-spline surfaces and quadrics. Proceedings of the Institution of Mechanical Engineers-Part B, vol. 210, 1996, p. 147–57.

[6] Varady V, Martin RR, Cox J. Reverse engineering of geometric models — an introduction. Computer-Aided Design 1997;29(4): 255–68.

[7] Chivate PN, Jablokow AG. Solid-model generation from measured point data. Computer-Aided Design 1993;25(9):587–600.

[8] Weir DJ, Milroy M, Bradley C, Vickers GW. Reverse engineering physical models employing wrap-around B-spline surfaces and quadrics. Proceedings of the Institution of Mechanical Engineers — Part B, vol. 210, 1996, p. 147–57.

[9] Piegl L, Tiller W. The NURBS Book. Berlin: Springer, 1995.

[10] Soucy M, Laurendeau D. A general surface approach to the integration of a set of range views. IEEE Pattern Analysis and Machine Intelligence 1995;17(4):344–58.

[11] Fang TP, Piegl L. Delaunay triangulation in three dimensions. IEEE Computer Graphics and Applications 1995;15(5):62–69.

[12] Cignoni P, Montani C, Scopigno R. A fast divide and conquer Delaunay triangulation algorithm. Computer-Aided Design 1998;30(5): 333–41.

[13] Lawson CL. Software for C^1 surface interpolation Mathematical Software III. New York: Academic Press, 1977.

[14] Choi BK, Shin HY, Yoon YI, Lee JW. Triangulation of scattered data in 3D space. Computer-Aided Design 1988;20(5):239–48.

[15] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. Computer Graphics (Proceedings of SIGGRAPH), 1992, p. 71–8.

[16] Hoppe H. Progressive meshes. Computer graphics (Proceedings of SIGGRAPH), 1996, p. 99–108.

[17] Eck M, DeRose T, Duchamp T, Hoppe H, Lounsbery M, Stuetzle W. Multiresolution analysis of arbitrary meshes. Computer Graphics (Proceedings of SIGGRAPH), 1995, p. 82–90.

[18] Ferrie FP, Lagarde J, Whaite P. Darboux frames, snakes and superquadrics: geometry from the bottom up. IEEE Transactions on Pattern Analysis and Machine Intelligence 1993;15(8):771–83.

[19] Piegl LA, Richard AM. Tessellating trimmed NURBS surfaces. Computer-Aided Design 1995;27(1):16–26.

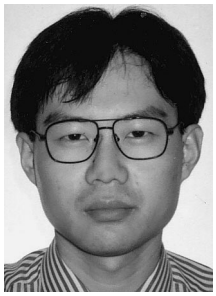
[20] Sheng X, Hirsch BE. Triangulation of trimmed surfaces in parametric space. Computer-Aided Design 1992;24(8):437–44.

[21] Bradley C. Error-based polyhedral meshing of 3D vision data. SPIE Photonics East, Conference on Sensors in Machining Boston, 1999, p. 26–33.

[22] Li S, Jerard RB. Five axis machining of sculptured surfaces with a flat-end cutter. Computer Aided Design 1994;26(3):165–78.

[23] Filip D, Magedson R, Markot R. Surface algorithms using bounds on derivatives. Computer Aided Geometric Design 2000;3:295–311.

W. Sun received his BEng from Shanghai Jiao Tong University, China. He is currently a MEng candidate at the Department of Mechanical & Production Engineering, National University of Singapore. His research interest is reverse engineering.



Y.F. Zhang received his BEng from Shanghai Jiao Tong University, China in 1985 and PhD from the University of Bath in 1991. He is currently an Associate Professor at the Department of Mechanical&Production Engineering, National University of Singapore. He is a senior member of SME. His research interests include reverse engineering, multi-axis tool path planning, and the computational intelligence in design and manufacturing.

H.T. Loh is an Associate Professor in the Department of Mechanical and Production Engineering at the National University of Singapore (NUS). He is also the Director of the Centre for Robust Design, which is a multi-disciplinary research centre focussing on quality and reliability issues. He is a Fellow of the Singapore-MIT Alliance, which is an innovative engineering education and research collaboration between MIT, NUS and the Nanyang Technological University, to promote global education and research in engineering. Dr Loh's research interests are in the areas of robust design, rapid prototyping and computer aided design.

Colin Bradley teaches and performs research, in the design and manufacturing area, at the University of Victoria, BC, Canada. The research focuses on the application of emerging technologies to assist the improvement of the design and manufacturing processes. Dr Bradley received a BASc from the University of British Columbia, a MSc from Herriot-Watt University and a PhD from the University of Victoria.