

Reverse Engineering Employing a Wrap-Around Surface Triangle Set

Colin Bradley (cbr@me.uvic.ca)
University of Victoria, Canada

Vincent Chan
University of Victoria, Canada

Key words: reverse engineering, 5-axis NC machining, sculptured-surface machining

Abstract: This paper presents a reverse engineering technique for modelling objects digitized by a cloud of three-dimensional (3-D) data points. A 3-D machine vision system, integrated with a computer numerically controlled (CNC) machine tool, and the associated software for generating a surface-triangle set (STS) is described. Generation of the STS model is accomplished utilizing a data independent, angle-optimal triangulation of the cloud data. The STS model is capable of modelling the cloud data sets to within the tolerance determined by the digitization density. A technique for producing a CNC tool path from the STS model is also presented.

1. INTRODUCTION

The process of capturing object form through surface digitization and generating a CAD model, of the object, is termed reverse engineering. In the context of general manufacturing methods, reverse engineering is an important process for instances where a product initially exists as a designer's model in a medium, such as styling foam or modelling clay. The model's surface form must be digitised and the data transformed to a representation compatible with CNC machining or CAD software. The digitisation process can be achieved by a co-ordinate measuring machine (CMM) or computer vision system. Compared to CMM touch probes, 3-D vision systems have the advantages of high data collection speed and non-contact measurement. The primary limitations are the trade-off between the sensor's accuracy and depth of field and that the sensor data is un-structured and requires extensive initial processing to construct smooth and accurate surface models. The remainder of the paper describes a new approach to handling a frequent requirement in die-cavity machining; the manufacture of die and mold tooling from a physical, three-dimensional part model in an efficient manner. The geometrical model produced by the process described here can be utilised for generating CNC machine tool paths.

2. MODELING 3-D DIGITIZED DATA

A method for meshing a set of 3-D cloud data points, $X = \{x_1, \dots, x_n\} \subset R^3$, on an unknown surface, U , was proposed by Hoppel. A surface normal vector is calculated at every data point from a local approximating plane derived from neighboring points. A uniform and regular 3-D grid of data is generated from the normal vectors and data points. A triangular mesh is created by connecting the points in adjacent rows and columns present in the regularized grid. The method is limited by the requirement that the input data points x_i must be uniformly and densely distributed across the given surface, U . The method is also computationally intensive due to the large number of calculations performed for each point. The method cannot be used for sparse or irregularly distributed cloud data due to errors induced by interpolating between sparse data. This will result in erroneous data points that do not accurately model the surface. Other researchers have developed 3-D cloud data meshing

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35392-0_40](https://doi.org/10.1007/978-0-387-35392-0_40)

algorithms, Milroy2 and Turk and Levoy3, however, they capitalize on connectivity information inherent in the 3-D data generated by a specific machine vision system.

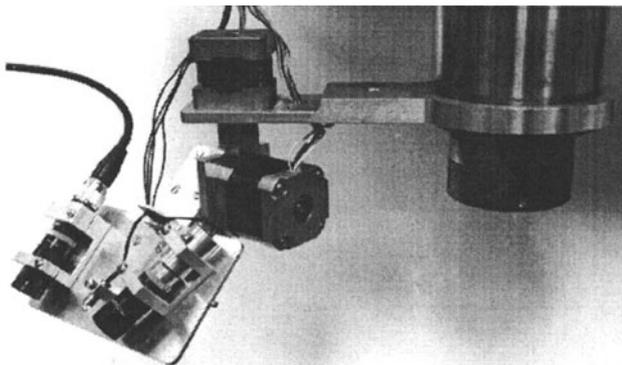


Figure 1. Range sensor mounted on the CNC machine tool.

3. THREE DIMENSIONAL VISION SYSTEM

The reverse engineering system consists of a Moiré interferometer range sensor head mounted on a 5-axis CNC milling machine, as shown in the photograph of Figure 1. The milling machine collet mount includes two stepper motors, that rotates the sensor head in two rotational degrees of freedom. Overall, control is provided by a PC-based programmable motion control (PMC) board. The system can position the sensor head, around an object placed on the machine tool deck, within a total work volume of 75 cm x 50 cm x 24 cm. The sensor positioning system is utilized in ensuring that the minimum number of images is taken for complete object coverage and each range image is sequentially added to the overall 3-D data file. Further details of the machine vision system can be found in reference 4.

The CNC machine provides x, y, and z motion through its closed loop positioning system. The PMC card also controls the 8 oz-in and 12 oz-in stepper motors, located on the sensor mount, as shown in Figure 1. Supervisory control software was developed and implemented on the PC to perform the following functions:

- Integration of the cloud data, generated by the sensor in each viewing location, with the absolute position obtained from the 3 CNC machine tool motor encoders and the 2 rotational stepper motors on the sensor head mount. The control software was written in the PMC language and generates a cloud data file in one unified global reference frame.
- Providing a user interface for the sensor, location feedback and data file organization. Digitized data from an object can be displayed in a number of formats including orthographic and isometric views.
- Providing motor control through the user interface. Position commands are passed to the custom motor control program that calculates the number of motor shaft turns, for each axis, to correctly position the sensor at the desired 3D spatial co-ordinate.

4. CLOUD DATA MESHING ALGORITHM

The digitized data, $D = \{d_1, \dots, d_n\}$, is a set of 3D coordinates of points, $d_i = \{x_i, y_i, z_i\} \in R^3$, on the surface of the object, S_o . The data set D forms the sampled representation of S_o , it can be from any 3-D digitizing device and is not restricted with respect to object shape. The triangular mesh M_o is constructed from D by first reducing the size of D , growing triangular

mesh patches over the reduced data set, and optimising the mesh to best represent the object's shape.

a) Initial Mesh Generation. The objective of mesh generation is to cover the reduced point set, R , with an initial surface mesh of triangular facets, M_i . A triangular mesh surface is comprised of one, or several, mesh patches, p_i , and each patch is grown over R from a starting seed point, $r_s \in R$. Any point can be selected as the seed and the first patch, p_1 , grows expands until all points in R are meshed. Complex objects require each distinct region, on the object, to be individually meshed and all patches subsequently join to form a continuous surface, M_i . To illustrate, a multiple patch mesh is shown in Figure 2.

The meshing process, illustrated in the flowchart of Figure 3, is initiated with the selection of a seed point, r_s , and all remaining points R form the initial set of valid points for triangulation. The set R is sorted in increasing Euclidean distance from r_s forming the sorted subset $R_s = \{r_{s1}, \dots, r_{sn}\} \subset R$. The seed point r_s becomes r_{s1} , the next nearest point becomes r_{s2} , and so on. Data from previously meshed patches are removed from R and only the boundary points of existing patches remain.

b) Mesh Optimisation. A triangular mesh surface patch must be optimized to further improve the object representation. The optimized mesh, M_o , contains the same vertices and the same number of triangles as the initial mesh, M_i . Edges are spatially relocated to either increase the mesh smoothness or enhance object edges present in the mesh. The optimization algorithm iteratively examines every edge in the mesh surface, and applies an appropriate criterion for ensuring mesh improvement. Each non-boundary edge is shared by two triangles, which together form a quadrilateral, having the target edge as a diagonal. If the quadrilateral is convex, the target edge can be swapped to the opposite diagonal, to create two different triangles, between the same four vertices, as shown in Figure 4. The two optimization criteria are:

1) Mesh smoothness. Edge locations that maximizes the smallest interior angle for the two triangles and results in a mesh containing approximately equilateral triangles.

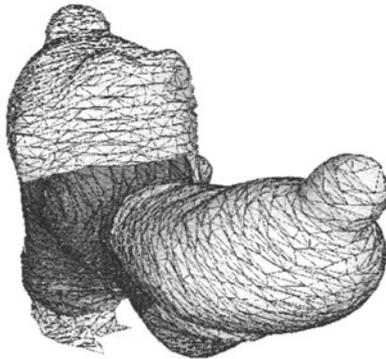


Figure 2. Six contiguous surface mesh patches defining an object.

2) Mesh edge enhancement. Mesh edges are aligned with the edges in the object surface to enhance sharp features. Neighboring triangles are examined and the final orientation of the target edge is the one which minimizes the variation in surface normals between the quadrilateral and the neighboring triangles.

For either criterion, the set of edges, E , is examined on each iteration of the algorithm until no further edge movement is evident. The best results have been obtained by applying the criteria sequentially; the initial mesh, M_i , is first optimized for smoothness, then optimized

to enhance the features. This prevents artificial features that exist in the initial mesh, but not on the object surface, from being erroneously enhanced.

5. STS MODELLING EXAMPLES

The plastic toy, shown in Figure 2, typifies the complex surfaces found in mold manufacture and is a difficult object to model for three reasons:

- The surface is too complex to be mapped to a 2D planar domain or any simple 3D geometric domain.
- The surface is not divided into simple distinct regions that can have several separate surface patches fitted and merged together.
- The surface has large smooth regions interrupted by sharp features with high surface curvature.

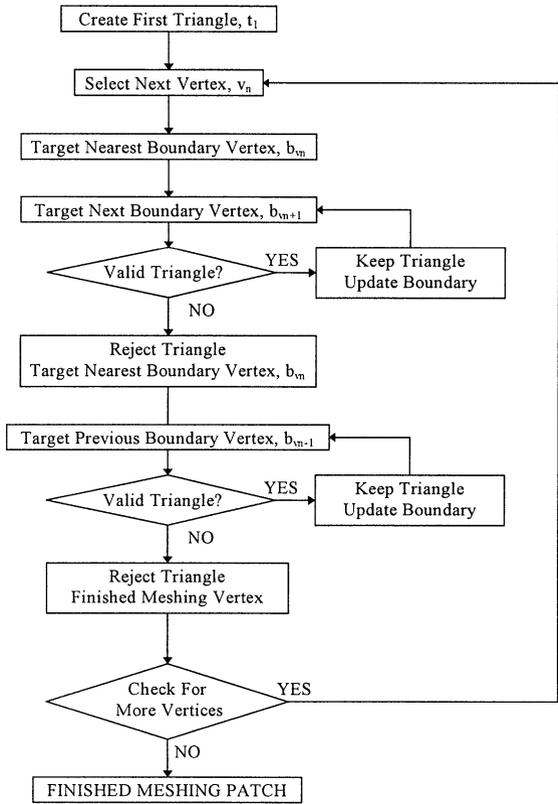
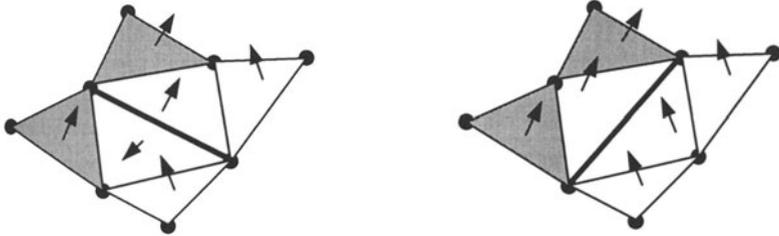


Figure 3. Flowchart of the data meshing algorithm.

The 3D triangular meshing algorithm successfully approximates the toy object using six patches, as shown in Figure 2. The six patches, in the order they were grown, are: nose; top of head; right cheek; left cheek; chin; back of head and neck.

The complexity of the surface makes it difficult to choose successful parameters for patch growth. They must allow large patches that join without gaps or holes, and without forming erroneous triangles. The patches are limited in size by a fixed number of vertices, rather than features on the surface. The seed vertices are selected at the regions of highest curvature, e.g. top of the head, tip of the nose, bottom of the chin, and points of the cheeks. Patches grow more reliably over high curvature regions if they start there. The last patch fills the gap at the back of the head and around the collar.



BEFORE EDGE ENHANCEMENT

AFTER EDGE ENHANCEMENT

Figure 4. Facet edge swapping and edge enhancement process.

The six component patches are merged into a single mesh prior to optimization. The optimized meshes are shown in Figure 5. The most accurate mesh is obtained by optimizing first with the smoothing criterion, shown on the left, then with the feature-enhancing criterion, shown on the right. The final mesh is still smooth at regions of low curvature (e.g. the bulbous nose) but possesses clearly defined features (e.g. the eyebrows, nose wrinkles, and cheeks).

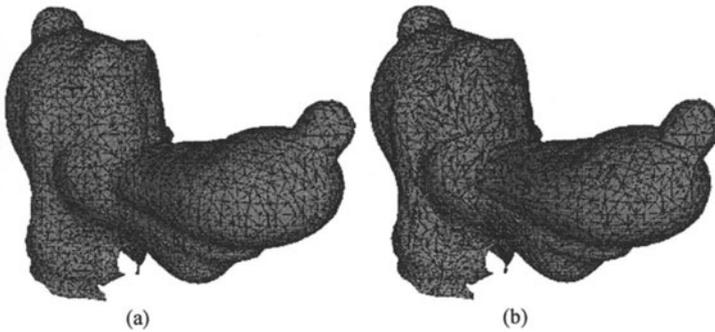


Figure 5. Optimised STS meshes: (a) smoothing criterion (b) feature enhancing criterion.

The largest apparent gap in the mesh, on the throat under the chin, is due to the incomplete digitizing of the Pluto figurine. This gap in the data points is too large for the mesh to span. The error between the optimized mesh and the digitized points is largest at the boundaries between patches, where holes or erroneous triangles exist. The difficulty in choosing growth parameter values to produce large patches on the complex surface means that occasional errors are inevitable. These errors can be corrected by manually inserting and removing individual triangles, if necessary. The size of these errors is approximately equal to the voxel bin size of 1.5 mm.

Due to the complexity of the surface, the Pluto mesh required six patches, using different patch growth parameters. The mesh patches required some trial and error, and several

patches had to be removed and re-grown. The initial digitized surface of 29,798 points was fitted with an optimized mesh of 6240 triangles in approximately three hours. The breakdown of time for each stage of the process is: 10 seconds to import and reduce the initial data set, approximately 2 hours to grow the initial mesh over the object's surface, 21 minutes to perform optimisation with respect to the smoothness criterion and 17 minutes to optimise with respect to the feature enhancement criterion.

6. TOOL PATH CONSTRUCTION

In this section, a method for generating the tool path for 5-axis milling of sculptured surfaces is presented. The method employs the STS model described above and focuses on a technique suitable for machining with a flat-ended cutter. Employing the STS model as a starting point, the following steps are necessary to create the CNC code: cutter contact (CC) point generation, determination of the cutter locations (CL)⁵ and post-processing to obtain the CNC part program. One issue addressed in CL data creation is the use of a configuration space.

6.1 Cutter Contact Point Generation

The STS is employed in creating the CC points on the surface and each triangular facet is described by the spatial co-ordinates of each vertex and the three components of the surface normal vector. The STS topological information, critical to efficient use of the mesh, is organised as follows: a facet vertex specified by a (x, y, z) triplet, an edge specified by two end indices and indices to the two facets that share the edge. Thus, every facet is specified by an index to each vertex, three edges, the three adjacent facets and the normal vector to the facet. For the overall digitized object, the edges that define the boundary are also flagged.

The CC points are created from the STS by means of an intersection plane⁶, as shown in Figure 6.

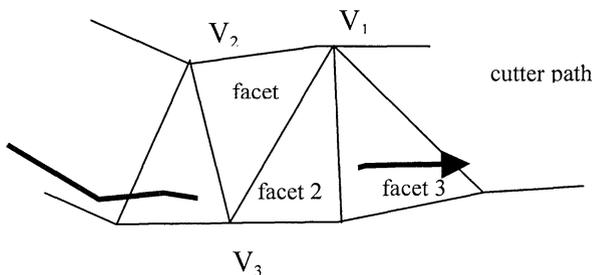


Figure 6. Generation of the CC points from the STS surface model.

The cutting plane intersects with the STS thereby creating a set of points on the edges of each facet where the plane intersects the edge. The corresponding set of points, or edge list, defines a nominal CC line along the surface. The length of the segment between edge points is the step forward distance from one CC point to the next. The STS models the surface to within the pre-defined tolerance and, therefore, the step-forward length restriction is automatically satisfied. The intersection checking, for all candidate edges, along the cutter path is efficient because all edges of each facet are indexed. Successive CC points are determined in this manner until a boundary edge (flagged) is reached.

6.2 Cutter Location Point Generation

The cutter location (CL) is defined as the center of the bottom of the tool, for a flat-end mill. The diagram of Figure 7 shows a tool traversing a path defined by the edge list CC points. For a given position and orientation, the CL data is specified by the location of the tool center point, \mathbf{p} , and the tool axis orientation, \mathbf{u} . A local co-ordinate system can also be defined, at each cutter contact point and is formed by the N-axis, in the direction of the facet normal vector, and the F-axis, in the direction of the cutter feed direction.

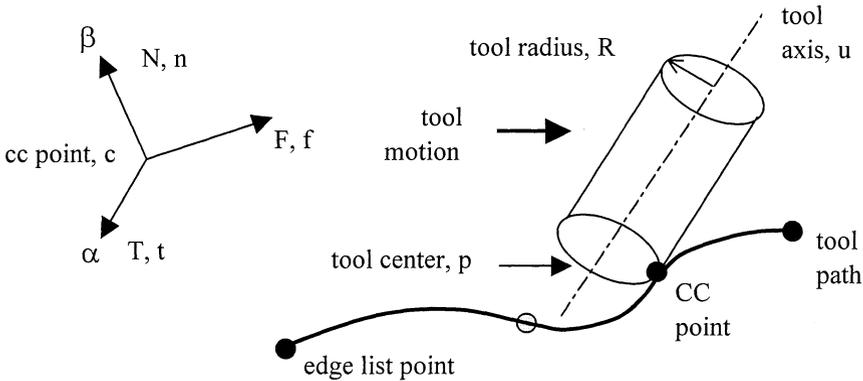


Figure 7. Cutter orientation and local co-ordinate system.

The third axis of the co-ordinate system, T, is defined by the cross product of the unit vectors in the F and N directions, i.e. $\mathbf{t} = \mathbf{f} \otimes \mathbf{n}$. Thus, the three unit vectors (\mathbf{t} , \mathbf{f} , and \mathbf{n}), in the axial directions, and the cutter contact point, \mathbf{c} , define an orthonormal co-ordinate frame. It can also be shown that the cutter center (\mathbf{p}) and the orientation (\mathbf{u}) are a function of the angles α and β , which are the cutter orientation angles with respect to the T-axis and N-axis of the local co-ordinate frame (see Figure 7).

Therefore, for a given cutter contact point the CL-data depends entirely on the angles α and β . The configuration space (C-space) technique⁷ has been used widely in robotic motion planning to represent a moving object as a point in an appropriate parameter space, to which the obstacles are also mapped. The method simplifies the motion-planning problem by mapping a complex geometrical problem into the simpler C-space. The C-space method has also been proposed to simplify the problem of avoiding tool gouging and collision in multi-axis CNC machining.

$$\mathbf{u} = H_1(\alpha, \beta) \quad \text{and} \quad \mathbf{p} = H_2(\mathbf{c}, R, \mathbf{u})$$

The C-space is defined as the planar co-ordinate system $\{\alpha, \beta\} \in R^2$, shown in Figure 8, where a point represents the tool rotational β and tilt α angles respectively. That is, a point in C-space represents a single tool position and orientation in the 3D-work space of the tool. The C-space is sub-divided into a collision area and a free area. The free area represents all permissible values of α and β that the tool angles may assume, whereas the collision area represent denotes where tool collision, surface gouging or violation of the limit's of the machine tool may occur. The delineation between the regions is determined by projecting the neighbouring obstacles, surrounding surface regions and machine angular limits onto the C-space. For a 5-axis machine tool, the individual free areas are formed and then an overall feasible machining region is obtained from the union of the three. A point in this region represents a gouge-free, collision-free and attainable (by the machine) tool \mathbf{p} and \mathbf{u} .

As an illustration, Figure 8 shows the limits of the machine tool movement expressed as a C-space region. The physical machine angular limits, of its two revolute joints, are transformed to the local co-ordinate system (\mathbf{t} , \mathbf{f} , and \mathbf{n}) and expressed as maximum and minimum angles in α and β ; (α_0 , α_m) and (β_0 , β_m). To ease the computational burden, the β axis is kept fixed while all feasible ranges of the α axis are examined. Employing the C-space approach, checks can be made at each CC point to determine whether collision or gouging will occur. If they do, then the closest workable values of α and β in the overall feasible C-space can be selected. In this manner, a tool path can be generated across the STS model such that collision, gouging and machine tool limits are all checked and any necessary changes in the commensurate values of α and β made.

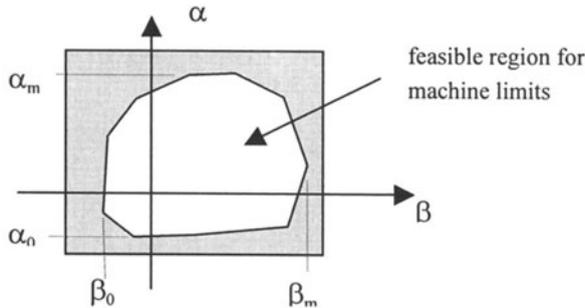


Figure 8. The configuration space derived for the machine limits.

7. CONCLUSIONS

The paper describes a hardware and software system that can perform the reverse engineering of sculptured surface objects. Digitization and generation of the copious cloud data set is performed by a 3D-vision system integrated with a multi-axis CNC machine tool. The cloud data, acquired from numerous views, is modelled by a multi-patch, triangular facet meshing algorithm. The technique creates an accurate model of complex surfaces in a form that is suitable for manufacture by a CNC milling machine. The resulting mesh is employed to directly create cutter contact points, from which the cutter location is ultimately found. Cutter location selection is based on spatial location and orientation that avoids gouging of the underlying surface, for a flat-end mill of a given radius.

REFERENCES

- 1 Hoppe, H.. Surface reconstruction from unorganised points, Ph.D. Thesis, University of Washington, USA, 1994.
- 2 Milroy, M., Bradley, C. and Vickers, G.W.. Automated laser scanning based on orthogonal cross sections. *Machine Vision and Applications*, 1996, Vol. 9, 106-118.
- 3 Turk, G. and Levoy, M.. Zippered polygon meshes from range images. *Computer Graphics (SIGGRAPH '94 Proceedings)*, 1994, 28(3), 311-318.
- 4 Archibald, I.. Adaptive mesh generation from machine vision data. M.Sc. Thesis, University of Victoria, Canada, 1997.
- 5 Choi, B.K., Park, J.W. and Jun, C.S.. Cutter-location data optimization in 5-axis surface machining. *Computer Aided Design*, 1993, 25(6), 377-386.
- 6..Li, S.X. and Jerard, R.B.. 5-Axis machining of sculptured surfaces with a flat-end cutter. *Computer Aided Design*, 1994, 26(3), 165-178.
- 7 Choi, B.K., Kim, D. and Jerard, R.B.. C-space approach to tool-path generation for die and mould machining. *Computer Aided Design*, 1997, 29(9), 657-669.