26

Integration of reverse engineering and rapid prototyping with data reduction

Y. H. Chen, C. T. Ng, Y. Z. Wang
Dept. of Mechanical Engineering, The University of Hong Kong
Pokfulam Road, Hong Kong,
Tel: 852-28597910, Fax: 852-28585415,
E-mail: YHCHEN@hkucc.hku.hk

Abstract

Reverse engineering is a methodology for constructing CAD models of physical parts by digitizing an existing part, creating a computer model and then using it to manufacture the component. When a digitized part is to be manufactured by means of rapid prototyping machines such as stereolithography apparatus (SLA) and selective laser sintering equipment (SLS), etc., it is not necessary to construct the CAD model of a digitized part. This will be described by the proposed novel method which can construct a STL file (the de facto file format for rapid prototyping machines) directly from digitized part data. Further more, the STL file can even be constructed in a way that significant data reduction can be achieved at the users' discretion.

Keywords

Reverse engineering, rapid prototyping, triangulation, STL

1 INTRODUCTION

Reverse engineering is an important process in the design and manufacturing context. This can be illustrated by two examples. First, a product that initially exists in a designer's medium such as clay and wood, must have its surfaces

Computer Applications in Production and Engineering. F. Plonka and G. Olling (Eds.)

digitized in order to convert the model to a computer-based representation(compatible with CAD/CAM systems). Secondly, if part drawings are not available in CAD form such as some proven old designs or antiques, CAD models of such parts can be constructed through reverse engineering techniques. Given a CAD model of a part, traditional methods such as CNC machining can be used to produce the parts. However, it is sometimes impossible for CNC machines to machine complicated surfaces both rapidly and cost-effectively. Over the past a few years, rapid prototyping machines have been widely used in industry. A variety of rapid prototyping technologies have emerged(Yan and Gu, 1989, Miller and Grote, 1995, Wallet et al., 1992). They include Stereolithography (SLA), Selective Laser Sintering (SLS), Fused Deposition Manufacturing (FDM), Laminated Object Manufacturing (LOM), Ballistic Particle Manufacturing (BPM), and Three Dimensional Printing (3D Printing). These technologies are capable of directly generating physical objects from a CAD model. They have an important common feature: physical parts are produced by adding materials layer by layer. This is opposite to traditional machining methods which make physical parts by removing material.

In rapid prototyping, STL file format has become the de facto standard(3D Systems, 1988). An STL file consists of a list of triangular facet data. Each facet is uniquely identified by a unit normal and three vertices. The normal and each vertex are specified by three coordinates each, so there is a total of 12 numbers stored for each facet. Apart from this, triangular facets in an STL file must also obey the following two rules.

Facet orientation

The facets define the surfaces of a 3-dimensional object. As such, each facet is part of the boundary between the interior and the exterior of the object. The orientation of the facet is specified redundantly in two ways which must be consistent. First, the direction of the normal is outward. Second, the vertices are listed in counter-clockwise order when looking at the object from the outside. This rule is illustrated in Figure 1.

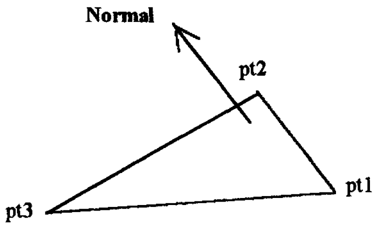


Figure 1 Orientation of a triangular facet.

Vertex-to-vertex
Each triangular facet must share two vertices with each of its adjacent triangles. In other words, a vertex of one triangle cannot lie on the side of another. This rule is illustrated in Figure 2.

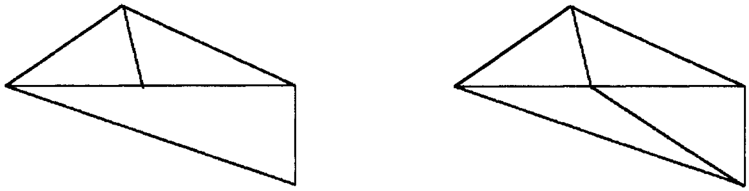

(a) Violation of vertex-to-vertex rule (b) Correct triangulation

Figure 2 The vertex-to-vertex rule.

Most recent commercial CAD/CAM software systems are capable of generating STL files directly from a surface model. The majority of triangulation methods are based on a known surface model(Schumaker, 1993, Sheng and Hirrch, 1992). Triangulation of scattered data in 3D space often has an objective of constructing smooth surfaces(Oxley, 1985, Park and Kim, 1995). Literature about optimized STL file generation from reverse engineering data cannot be found. The most relevant publication is found in Hamann paper(Humann, 1994). His proposal removes a triangle based on curvatures at the three vertices of a triangle. Again, his research has smooth surface fitting in mind. This paper describes a data reduction method for automatic STL file generation directly from reverse engineering data. The proposed method is based on surface normals of triangles that share a data point.

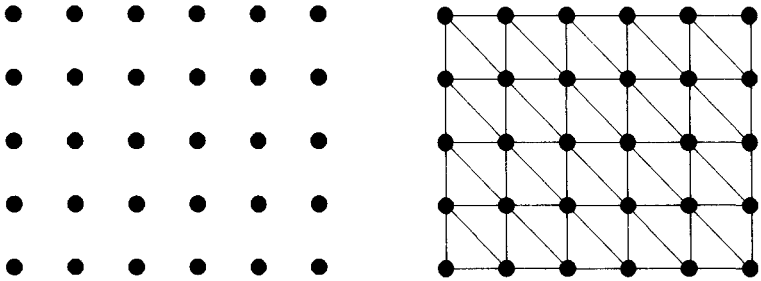
2 AUTOMATED STL FILE GENERATION

There are various methods for part digitization. The presented research utilizes a Mitutoyo BLN 122 coordinate measuring machine (CMM) to digitize a part. In order not to miss any detail of part geometry, a large number of measurement points are normally defined. If all points are used in STL file generation, the file can easily becomes huge that the whole rapid prototyping process will be slowed down significantly. In fact, keeping lots of data points in nearly planar regions is rather unsophisticated. Removal of data points in these regions would not affect the accuracy for rapid prototyping machines. A criterion based on normals of neighboring triangles has been defined as to which points can be removed in accordance with users' requirement.

Point data from CMM measurement is usually very regular and can be arranged in matrix form. If all measurement points are to be used for STL file construction, the problem becomes quite simple. In fact, the proposed method initially uses all points to form triangles in the following simple way:

```
Given a data point matrix $M(i,j)$
    where $i = 1, \ldots m; j = 1, \ldots n$
    For $j$ = 1 to $n$
        for $i$ = 1 to $(m-1)$
            join $M(i,j)$ to $M(i+1,j)$;
    For $j$ = 1 to $(n - 1)$
        for $i$ = 1 to $m$
            join $M(i,j)$ to $M(i,j+1)$;
        for $i$ = 1 to $(m-1)$
            join $M(i,j)$ to $M(i+1,j+1)$;
End.
```

With the above procedure, data points can be triangulated as shown in Figure 3. It can be seen that triangles formed in this ways conform to the STL file vertex-to-vertex rule.



(a) Data points

(b) Triangulation

Figure 3 Initial triangulation.

To generate a complete STL file, normalized surface normals of triangles must be specified. Given a triangle as shown in Figure 1 (pt_1, pt_2 and pt_3 must be arranged in counter-clockwise direction), normalized normals can be easily calculated from the following procedure.

(a) Form two vectors $\mathbf{V}_{31}$ and $\mathbf{V}_{32}$ by

$$\mathbf{V}_{31} = \mathbf{Pt}_1 - \mathbf{Pt}_3$$
$$\mathbf{V}_{32} = \mathbf{Pt}_2 - \mathbf{Pt}_3 \qquad (1)$$

(b) Calculate surface normal from the cross product.

$$\mathbf{n} = \mathbf{V}_{31} \times \mathbf{V}_{32} \tag{2}$$

(c) Normalize **n** through the division.

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{|\mathbf{n}|} \tag{3}$$

With the above information, STL file can be readily generated.

3 DATA REDUCTION

3.1 Data reduction in percentage

If the digitized object has large planar or near planar regions, it is desirable to remove some of the points in these regions. In order to identify which points can be removed, a point weighting scheme based on a point surrounding triangles' normals is developed and described in the following:

(a) Search all triangles Δ that share the point in concern.
(b) Let one of the surface normal **U** in Δ be the reference vector, calculate subtractions of **U** with the rest of surface normals in Δ.

$$\begin{aligned} \mathbf{UV}_1 &= \mathbf{V}_1 - \mathbf{U} \\ \mathbf{UV}_2 &= \mathbf{V}_2 - \mathbf{U} \\ &\bullet \\ &\bullet \\ &\bullet \\ \mathbf{UV}_n &= \mathbf{V}_n - \mathbf{U} \end{aligned} \tag{4}$$

(c) Transform all resulting vectors to first quadrant by taking the absolute values of all vector components, e.g. $\mathbf{UV}_2$ = ai + bj + ck is transformed to $\mathbf{UV}_2$ = |a|i + |b|j + |c|**k**.
(d) Sum up all the resulting vectors.

$$\mathbf{W} = \mathbf{UV}_1 + \mathbf{UV}_2 + \ldots + \mathbf{UV}_n \tag{5}$$

(e) Calculated the average quantity d by

$$d = \frac{|\mathbf{W}|}{n} \mathbf{V}_n \qquad (6)$$

If d is small for a given point, the point stands a higher chance of being removed. Which points are to be removed depends on users' specifications. If a user want 10% of the point data be removed, then points with the lowest 10% of d will be removed.

3.2 Data reduction by bounded error

In computer visualization, polygonal mesh, triangles in particular are usually used to approximate object surfaces. In the triangulation of an object surface, accuracy is often controlled by bounded error(Kalvin and Taylor, 1996). The bounded error concept is also adapted in this research to control errors resulting from data reduction of an STL file in rapid prototyping.

In data reduction, two strategies can be used. One is to remove the point one at a time. The other is to remove several points, or a patch at the same time. After consideration of error accumulation and computation efficiency, patch removal strategy is implemented in this research.

Suppose a patch has k triangles. The normals and areas of the k triangles are $\mathbf{n}_1$ to $\mathbf{n}_k$ and S_1 to S_k respectively. Let

$$S = \sum_{i=1}^{k} S \qquad (7)$$

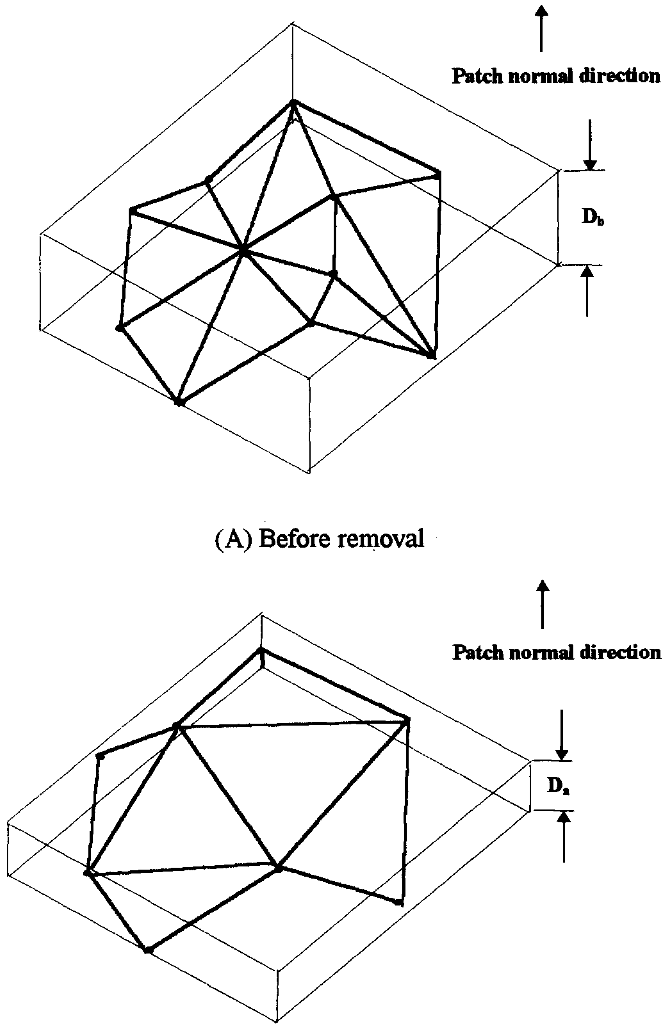
The patch direction normal is calculated by:

$$\mathbf{n} = \sum_{i=1}^{k} \frac{S_i}{S} \mathbf{n}_i \qquad (8)$$

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{|\mathbf{n}|} \qquad (9)$$

The direction normal $\hat{\mathbf{n}}$ is an indication of the direction of maximum projection. Based on the patch direction normal, a minimum enclosing box of a patch can be calculated. For illustration purpose, Figure 4(a) shows a patch and its bounding box. After data reduction (Suppose the three points in the middle are removed), the new patch and its bounding box are shown in Figure 4(b). The bounded error E_p is measured along the patch direction normal, that is

$$E_p = D_b - D_a \qquad (10)$$



(A) Before removal

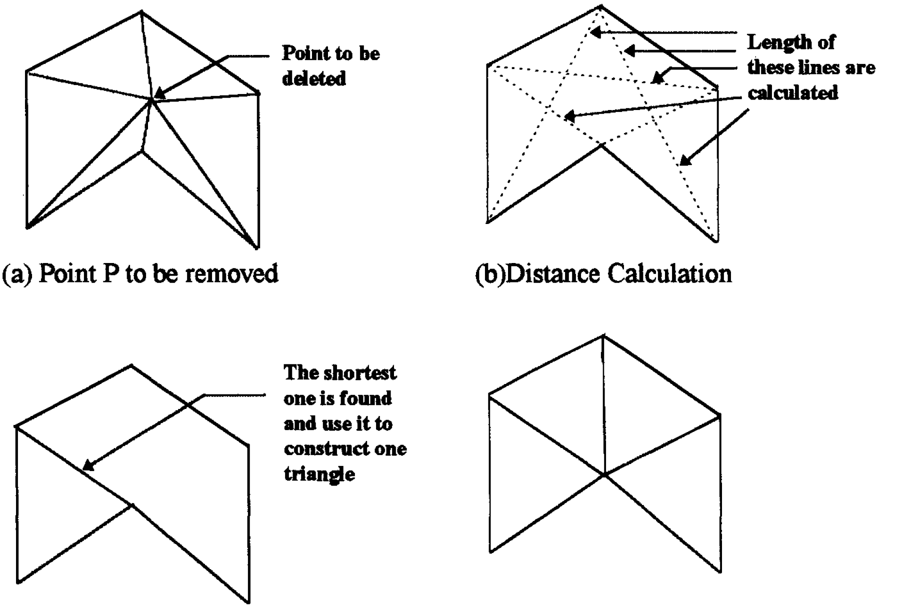
(B) After removal

Figure 4 Error definition.

This error can be specified by the user so that the accuracy after data reduction is still within control.

4 RE-TRIANGULATION

When a point is to be removed, triangles sharing this point must be removed too. This leaves a lank region’ which must be covered through re-triangulation. The implemented re-triangulation method is described in the following:


Point to be deleted
Length of these lines are calculated
The shortest one is found and use it to construct one triangle

(a) Point P to be removed

(b)Distance Calculation

(c) Form a new triangle

(d) The rest of triangles are formed in the same way

Figure 5 Re-triangulation.

(1) Form a point list **L** that are affected by the removal of a point. In figure 5(a), if point P is to be removed, the affected point list will be **L** = {P_1, P_2, P_3, P_4, P_5, P_6}.

(2) Take three consecutive points from the point list **L** and check whether the triangle formed by the three points is inside or outside the region formed by **L**. If not, take another three consecutive points and check again.

(3) Calculate the length of the line formed by the first and the last point of the three points as shown in Figure 5(b) in dashed lines.

(4) Repeat (2) and (3) until all combinations of consecutive three points have been exhausted.

(5) Find the shortest length and use the corresponding three points to form a real triangle as shown in Figure 5(c). Calculate the normal of the newly formed triangle and delete the middle point of the three points from **L** have $\mathbf{L}_1$.

(6) Repeat from (2) with point list **L** until no more than four points left in $\mathbf{L}_1$.

Figure 5(d) shows a re-triangulated region after a point is removed. Point removal and re-triangulation will go on iteratively until certain criteria are met. Two criteria are given for users' specification. One is the percentage of points to be removed and the other is the bounded error caused by points removal.

5 EXPERIMENTAL RESULTS

With the implemented method, a number of examples have been tested. One of the example is a human face. It was digitized using a Mitutoyo BLN 122 coordinate measuring machine as shown in Figure 6. As some parts of the human face has very low curvature change, some data points can be removed. In the experiment, data reduction for 20%, 50% and 80% have been specified for

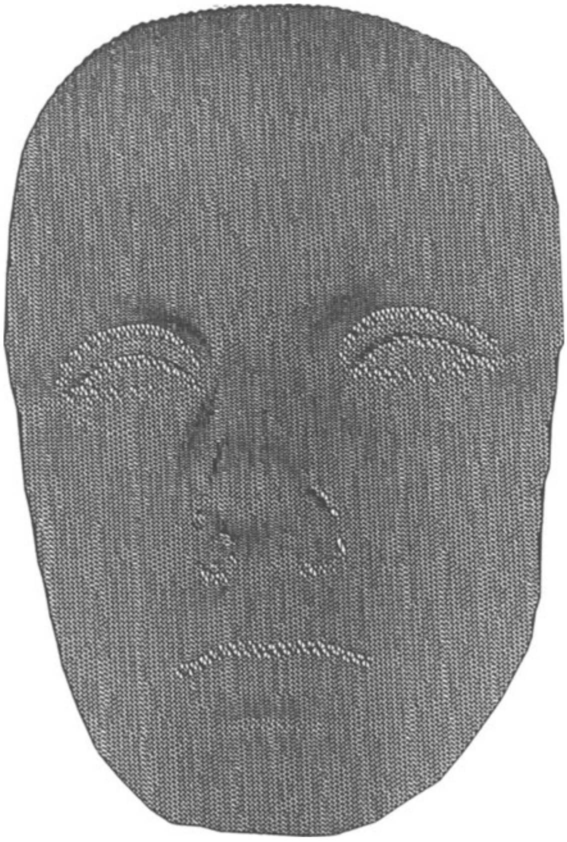


Figure 6 STL file without data reduction.

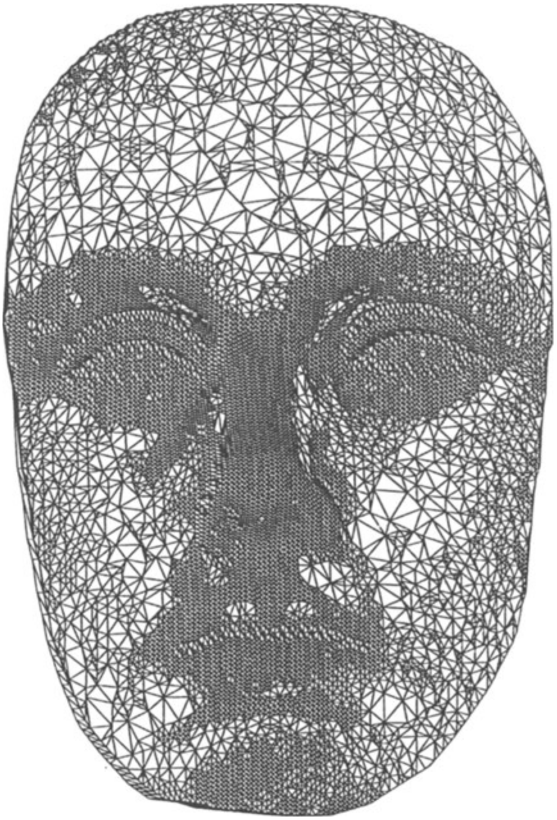


Figure 7 STL file after 50% data reduction.

prototype fabrication. Figure 7 shows the STL file after 50% of data reduction. It can be seen that regions with small curvature changes have large data reduction. This is exactly what is desired.

6 CONCLUSION

This paper has presented a novel method for STL file construction directly from CMM measurement data. When desired, the user can specify either the percentage of points to be removed or the bounded error caused by points removal in order to speed up computation and to reduce storage space. Several prototypes with different amounts of data reduction have been built using an SLS equipment which shows satisfactory results.

7 ACKNOWLEDGEMENT

This research is carried out with a CRCG grant (code:337/064/0023) from the University of Hong Kong.

8 REFERENCES

3D Systems (1988) Stereolithography interface specification. USA.

Hamann, B. (1994) A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design*, **11**, 197-214.

Kalvin, A.D. and Taylor, R.H. (1996) Superfaces: polygonal mesh simplification with bounded error. *IEEE Computer Graphics & Applications*, May, 64-77

Miller, J.L. and Grote, K.H. (1995) Solid freeform manufacturing technologies as an important step in the product development process. *Computers in Industry*, **28**, 11-16.

Oxley, A. (1985) Surface fitting by triangulation. *The Computer Journal*, **28**(3), 335-339.

Park, H. and Kim, K. (1995) An adaptive method for smooth surface approximation to scattered 3D points. *Computer-Aided Design*, **27**, 929-939.

Schumaker, L.L. (1993) Triangulations in CAGD. *IEEE Computer Graphics & Applications*, January, 47-52.

Sheng, X. and Hirrch, B.E. (1992) Triangulation of trimmed surfaces in parametric space. *Computer-Aided Design*, **24**(8), 437-444.

Wall, M.B., Ulrich, K.T. and Flowers, W.C. (1992) Evaluating prototyping technologies for product design. *Research in Engineering Design*, **3**, 163-177.

Yan, X. and Gu, P. (1996) A review of rapid prototyping technologies and systems. *Computer-Aided Design*, **28**(4), 307-318.

9 BIOGRAPH

Dr. Y.H. Chen is an assistant professor in The Department of Mechanical Engineering, The University of Hong Kong since 1993. Before joining The University of Hong Kong, Dr. Chen has worked in Motorola Electronics Pte, Ltd (Singapore), Asia Matshushita Electronics Pte, Led (Singapore), and Swire Technologies Pte, Ltd (Hong Kong) as senior automation engineer, senior research engineer and automation manager respectively. Dr. Chen research interest include CAD/CAM, reverse engineering and rapid prototyping.

Mr. C.T. Ng is an M.Phil research student under Dr. Chen.

Dr. Y.Z. Wang is currently a research associate working with Dr. Chen. He is a lecturer in The Department of Mechanical Engineering, Tianjin University, P.R. China.