PERGAMON

Computers & Industrial Engineering 38 (2000) 21–38

computers & industrial engineering

www.elsevier.com/locate/dsw

Direct integration of reverse engineering and rapid prototyping

Kwan H. Lee*, H. Woo

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

---

Abstract

Reverse engineering (RE) technology enables us to create CAD models of new or existing products by capturing surface data. Rapid prototyping (RP) is another emerging technology that allows us to promptly fabricate the physical prototype of a new product using a layered manufacturing technique. In this research, a new method that creates a direct link between these two technologies, is proposed. In RE, an enormous amount of point data is gathered during data acquisition. This leads to a huge file size that requires a large execution time. Surface modeling using these point data is time-consuming and requires expert modeling skills. Some researchers suggested creating an STL file directly from the point cloud data to avoid surface modeling tasks. The STL file, however, has many drawbacks. In this research, algorithms that greatly reduce point cloud data are developed, and thereby, the data file sizes are decreased considerably. The efficiency of the algorithms is demonstrated by comparing them with existing ones. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Rapid prototyping; Reverse engineering; Point cloud data reduction; STL file; Rapid product development

---

1. Introduction

As product varieties increase and life cycles shorten, the need to reduce product development time becomes more critical to maintain competitiveness in the market. The reduction of

---

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

PII: S0360-8352(00)00017-6

product development time, therefore, requires revolutionary improvements rather than gradual changes in technology. Both reverse engineering (RE) and rapid prototyping (RP) are emerging technologies that can play a promising role in reducing the product development time.

RE refers to the process of creating engineering design data from existing parts [1–3]. It recreates or clones an existing part by acquiring the surface data of an existing part using a scanning or measurement device. It is useful in recreating the CAD model of an existing part when the engineering design is lost or when the model has gone through many design changes. It enables us to capture the surfaces of design models that are otherwise impossible to determine. It also saves us from performing tedious manual dimensioning and tracing work. When a designer creates a new design using a mock-up, it is necessary to construct the CAD model of the mock-up for further use of the design data in analysis and manufacturing. The manual operation involved in RE requires a great amount of time and operator skills and is also subject to error. Coordinate measuring machines have been used to extract surface data but their data capturing operation is very slow for parts having complex free-form surfaces. In recent years, the laser scanning technology has improved significantly, and it has become a powerful tool in capturing the geometry of complicated design models.

Surface modeling in the RE process, however, is a challenging task. It takes a significant amount of time and skill to generate an accurate surface model from the point cloud data. This step is usually not automated and involves frequent manual interaction with the user even with a well-developed surface modeling software package [4,5].

The CAD model developed by an RE process can be converted to the physical prototype using an RP technique. Generally, in RP, physical parts are fabricated layer by layer. It uses additive manufacturing processes, which do not require any tools or set-ups compared to the subtractive techniques used in the traditional machining operation. It allows us to fabricate features that are difficult or impossible to fabricate by machining operation. Different fabrication methods exist for RP, but nearly all use the same geometry input format, called STL (Stereo Lithography), which consists of a list of triangular facet data. The STL format has advantages due to its simple structure and ease of use, but it also has serious drawbacks. It requires a large amount of memory as the accuracy of a part increases and also takes a significant amount of repair time when it has flaws such as gaps, overlaps, and mixed normal vectors. In order to bridge RP and RE technologies, efficient point cloud handling methods have to be developed first. Second, an accurate geometry input format for RP machines needs to be prepared. This paper proposes a procedure that allows fabricating RP parts directly from reverse engineered geometric data. With the procedure, algorithms that reduce the size of point cloud are developed.

2. Related work

Though both RE and RP are emerging technologies that have been well developed, little research has been done in integrating the two. The research activities described below deal with interfacing RE data to RP.

Hosni, et al. [6] developed a laser based system for capturing the geometric details of the object and tried to apply the captured data to RP using the STL format. Their research, however, was focused on developing an experimental scanning system for RE. Schoene and Hoffmann [7] used a triangulation method to generate the STL file directly from the scanned data using a 5-axis digitizer. Their objective was to develop an algorithm that can process point cloud data with undercuts and to handle unordered set of points. Vail et al. [8] investigated the issues related to interfacing the point cloud data directly to manufacturing processes, especially, RP. They used a modified Delauney triangulation algorithm combined with the marching cube algorithm to generate triangular meshes. It required post-processing procedures such as removing spurious triangles and gaps, harmonizing normal vectors, and detecting mesh intersections.

The shortcomings of the STL format have been discussed by many researchers. Yan and Gu [9] categorized the common sources of errors in RP and stated that the approximation used in the STL format as one of the major errors. Other researchers such as Jamieson and Hacker [10] and Qiming and Yungan [11] investigated the pros and cons of using original CAD files compared to using STL files. They claimed that the slice data directly generated from the original CAD file showed better accuracy. The STL files often had problems such as gaps, overlaps, and mixed normals. Many researchers (Makela and Dolenc [12], Rock and Wozny [13] and Bohn and Wozny [14]) discussed STL file problems, and currently dozens of commercial software packages exist that provide repair routines.

In establishing the link between RE and RP, another major challenge is to handle the huge amount of point data generated by scanning devices. The scanners have become more accurate and the speed of data acquisition has increased significantly. It becomes quite important to reduce the amount of acquired data and to convert it into a format required by manufacturing processes while maintaining the accuracy of the data. Fujimoto and Kariya [15] suggested an improved sequential data reduction method for 2D digitized point data. Their method guaranteed that the error from the reduced data set remained within the given angle and distance tolerance. Hamann [16] presented a new reduction method for triangulation files based on an iterative triangle removal principle. As a measure of reduction in the file size, each triangulation is weighted according to the principal curvature estimates at its vertices and the interior angles. Chen et al. [17] proposed a method that reduced the number of triangles used in an STL file generated from a point cloud. Triangles in planar or near planar regions were removed by comparing the normal vectors of corresponding triangles, and re-triangulation was performed. They showed the reduction of file size using the STL file of a human face digitized by a coordinate measuring machine. Veron and Léeon [18] introduced an approach to reduce the number of points of a polyhedral model using error zones assigned to each point of the initial polyhedron so that the simplified polyhedron intersects with each error zone. Hamann and Chen [19] proposed a method to reduce point data through various planar curves, compressing 2D images, and visualizing volumes. In their method, points were selected with respect to local absolute curvature estimates for a piecewise linear curve approximation. The degree of reduction was controlled either by the number of points to be selected or by the tolerance. Martin et al. [20] proposed a data reduction method using a uniform grid structure with a median filtering method. In their procedure, point clouds are subdivided and assigned to individual grids and the median point in each grid is selected to represent the grid.

As discussed above, the importance of interfacing RE to RP was raised but the past research lacked in establishing a framework that provided the direct interfacing of the two technologies.

3. Interfacing RE to RP

Bridging RE and RP technologies can facilitate the process of new product development. RE techniques focus on extracting the design data, while RP techniques concentrate on verifying or evaluating the design. Both technologies use 3D CAD models. In RE, 3D CAD models are created from physical parts, whereas in RP, 3D CAD models are utilized to make physical parts. Combining these two technologies is ideal since one helps in the design phase and the other in the prototyping phase and both use 3D CAD models. The modes of interface can be critical in terms of efficiency, as discussed below.

3.1. Interfacing modes between RE and RP

RE data can be linked to RP in three different ways as shown in Fig. 1. In the first path, a 3D surface model is created from point clouds and then converted to an STL file. The STL file is then sliced to generate a series of layers for RP fabrication. This path is generally followed by RE practitioners since most RE software packages concentrate on creating a surface model from point clouds. Once the surface model is created, CAD packages are used to convert it into the STL file.

Some researchers, as described in Section 2, have investigated the possibility of creating the STL file of a model directly from point clouds. This is shown in the second path of Fig. 1. This link bypasses the creation of surface models, which involves tedious manual operation.

The third path goes directly from point clouds to an RP slice file. In this path, the initial scan data is reorganized and reduced to make the contour data. This contour data can be categorized as an RP slice file since the layers of an RP part can be directly fabricated using these contour data. This link eliminates both the surface modeling task and the STL file generation task.

3.2. Problems of existing approaches

Each path described in Section 3.1 has both advantages and disadvantages. The first path has advantages in that the RE process can be carried out using the existing programs that are

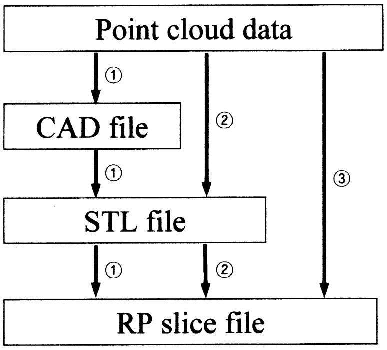


Fig. 1. Interfacing modes between RE and RP.

already available. The surface model created in this path is useful in many design and manufacturing applications, such as design modification, engineering analysis, and tool path generation. The task of creating surfaces from the point clouds, however, is difficult and time-consuming even with the help of surface modeling programs.

In the RE process, the initial point clouds require preprocessing operations which include the filtering of outliers, smoothing and blending of existing points, and registration of point clouds into one coordinate system. For surface modeling operation, the preprocessed point clouds generally need to be subdivided into many patches since the entire model cannot be approximated by a single mathematical surface. Then the boundary curves for each patch are created. By increasing the number of control points and the degree of the curves, more accurate shapes can be generated. Finally, all surface patches have to be connected to each other smoothly. The surface modeling task requires a considerable amount of manual operation and computing time. It is reported that the surface modeling task accounts for 90–95% of the RE time compared to 5–10% of that for the digitizing task [7]. Though highly skilled operators are required in performing these tasks, the resulting surfaces often are subject to error and generally show some deviation from the point cloud data. The difference between the point cloud data and the created surfaces can be shown using an error map in which the amount of deviation is represented by colored regions.

Both the first and the second path in Fig. 1 use the STL format. The STL format is the de facto industry standard and is being used by most RP vendors. The format is good in terms of the ease of use, the compatibility of files between different CAD packages, the simplicity of the data structure, and the efficiency for viewing and intersecting algorithms. The STL format, however, has many problems. It includes unsorted and unrelated triangles and does not retain any surface or feature data. Three vertices and a normal vector are required to represent each triangle but the vertices are redundant in the database. The format does not necessarily represent a complete object either. Some of the problems include having gaps, overlapping of triangles, and reversed normal vectors. The biggest problem comes from the error caused by approximating the model using planar triangular patches. Decreasing the size of triangles can reduce the amount of approximation error but this in turn, increases the computing time as well as the file size. Large size triangles, on the other hand, result in unacceptable parts due to the approximation error.

The third path illustrated in Fig. 1 can be a viable option when an RP part has to be fabricated from RE data. This path facilitates the RE process and reduces the time to make an RP part. However, the successful implementation of this path has some challenges that need to be overcome. In order to efficiently generate the slices for RP fabrication, the number of point clouds needs to be reduced. The enormous number of point clouds generated by scanners greatly increases the computing loads of the RE process, and often leads to situations that are practically unjustifiable to proceed with computation. Point data reduction methods for RP fabrication are described in the subsequent sections.

4. Point clouds reduction methods

In layered manufacturing, the slices with a constant thickness are used in part construction.

Therefore, only cross-sectional data at every fixed height are needed for part fabrication regardless of the density of the scanned data. This cross-sectional data will result in a series of stair-steps when a curved feature along the z-axis of a part is built. This implies that neither the entire data needs to be maintained nor a complete surface fitting is needed when RE data are utilized in RP fabrication. The point data for cross-sections that are within the tolerance of the previous cross-section need not be kept. The proposed algorithm reduces the amount of point data by eliminating these redundant cross-sections. The overall procedure from RE to RP, including the reduction in the number of point clouds, is shown in Fig. 2

Once the point clouds are acquired, the curvature changes of a part along z-axis are identified and recorded, and then they are used to divide the point data model into subregions. The point data for each cross-section within a subregion can be tested and either eliminated or maintained based on the boundary contours (or cross-sections). A ruled surface is assumed between these boundary contours and an intermediate contour can be calculated using the straight-line homotopy method. When the deviation between the intermediate contour of a subregion and the corresponding point data is within tolerance, the subregion of the model can be accurately represented by boundary contours and all intermediate point data can be removed. When the deviation is greater than the tolerance, the intermediate contour becomes a boundary contour and the corresponding subregion is divided into two subregions. The point data reduction process is then applied to each new subregion in a recursive manner. Upon completion of this procedure, the minimum number of contours that represents the part shape are determined, and the slices that are needed for part fabrication are generated from these data. The data reduction procedure is described in the following sections.

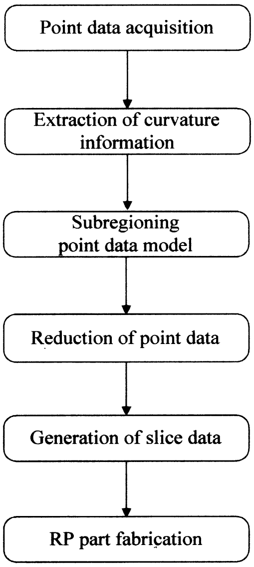


Fig. 2. The overall procedure from RE to RP.

4.1. Part scanning and point data acquisition

In RE, the quality of the data acquisition process is crucial to the accuracy of the resulting CAD model. It starts with preparing the part surface. Parts with dark or shiny surfaces are difficult to digitize and they need to be spray coated. Then the part has to be set up properly so that it can be scanned. If the part cannot be scanned with one orientation, it needs additional setups. When a part is scanned with multiple orientations, the scanned data from each orientation need to be combined and represented in a common coordinate system. This is called the registration. It is achieved by using reference features on the part, such as attaching tooling balls, or fixturing the part on a rotary table. After fixturing is completed, the part can be scanned. However, the initial scanned data cannot generally be used for surface modeling operation. They usually contain spikes, outliers, and poor quality regions. These need to be removed, and further preprocessing operations are required such as merging overlapped point data and performing registration of separate point clouds [21].

4.2. Point data arrangement

The types of point data arrangement can be different based on subsequent manufacturing operations. In this research, point data are arranged as a series of cross-sectional contours. The cross-sectional data format is popular in the construction of medical images such as those obtained by computed tomography or magnetic resonance imaging. This contour data format is selected since it can be directly applied to part fabrication in RP where parts are built layer by layer. The point clouds obtained in the previous step need to be converted to a series of cross-sectional contours. In order to construct each contour, the point clouds are divided by two parallel planes, giving a constant thickness ϵ. The point data contained in each slice are rearranged by projecting them on the cross-sectional planes.

4.3. Extraction of curvature information

In RP, the part is built by sweeping each cross-section vertically by the height of a layer and this results in zero vertical curvature for each slice. The curvature data on the cross-sectional planes is kept but the ones along the vertical direction are approximated due to the stair-steps inherent in layered manufacturing. It is crucial to keep critical features such as the edges and peak points along the z-axis; this can be best achieved by keeping cross-sectional contours at the z-heights where sharp curvature changes occur.

Keeping these critical cross-sectional contours require the acquisition of curvature data of a part along the z-axis. The procedure consists of four steps: (1) generate vertical cross-sections, (2) identify the points of curvature changes, (3) recreate the point data model using extracted points, and (4) calculate the point extraction ratio. The details of each step are described below.

4.3.1. Generation of vertical cross-sections

To obtain the curvature data along the z-direction, cross-sections are made perpendicular to the xy-plane of the model. As shown in Fig. 3, three different ways of vertical cross-sectioning

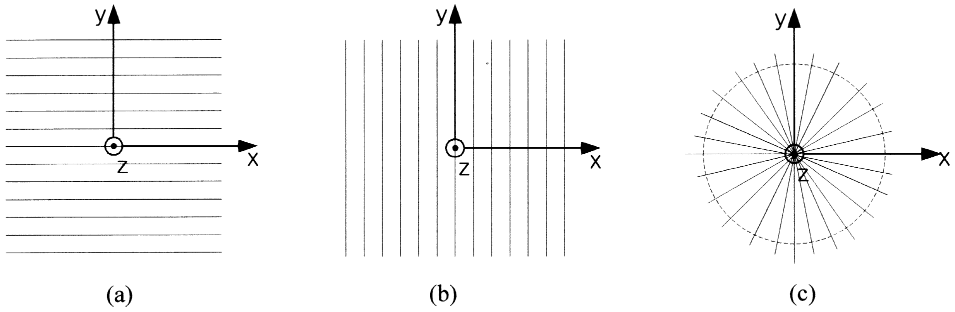


Fig. 3. Methods for generating vertical cross-sections.

are considered. The vertical cross-sectional planes can be generated along the y-axis, x-axis, or around the z-axis. The curvature information of a part is better acquired if the angles made between a vertical cross-sectional plane and the surface normal vectors at the intersection lines are small. One of the methods should be chosen according to the geometry of the part as shown in Fig. 3.

4.3.2. Identification and extraction of curvature changing points

A set of 2D point data is acquired for each vertical cross-sectional plane. For each point in the cross-sectional data set, the angular deviation, θ, can be calculated as shown in Fig. 4(a) using the vectors created from the consecutive points.

The angle θ represents the angular deviation of the current point from the direction formed by the previous two points, and can be used to detect curvature changes of the part on the vertical cross-sectional planes. We call this the 'angular deviation method'. Fig. 4(b) shows the extracted points that represent curvature changes.

4.3.3. Recreation of a point data model using extracted points

Upon completion of extracting points for all vertical cross-sections, they are gathered to recreate the point data model. The edges and features where significant contour changes taking place are detected using this model, and thereby, the overall geometry of the part is identified.

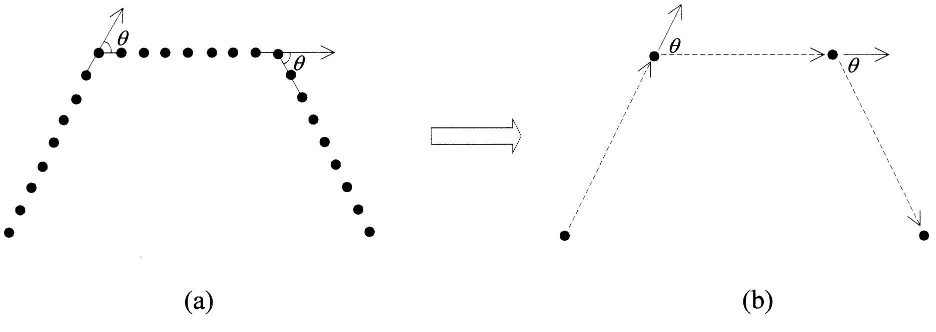


Fig. 4. Angular deviation method for extracting points.

4.3.4. Calculation of point extraction ratio

A point extraction ratio is introduced to identify the edges that involve significant curvature changes of a part on the vertical cross-sectional planes. Between the point data model and the extracted points, the ratio of the number of points at each level of height Z_i can be calculated. The point extraction ratio, r_{z_i}, is the number of extracted points over the total number of points in the horizontal cross-sectional plane at height Z_i as defined below.

$$r_{z_i} = \frac{(\text{The number of extracted points})_{z_i}}{(\text{Total number of points in the cross-section})_{z_i}} \quad \left(0 \le r_{z_i} \le 1\right)$$

A graph that shows the ratio of point extraction at different heights of a part can be created. The value of r_{z_i} is greater when a large amount of curvature changes occur, and vice versa.

4.4. Subregioning of point data models

Dividing the point data model into subregions can facilitate the data reduction process. The subregioning can be performed based on the point extraction ratio described above. Boundary contours of each subregion can be generated by the horizontal cross-sections located at z-heights where the point extraction ratio is greater than the given threshold value. The purpose of subregioning is twofold. First, it ensures accurate fabrication of edges and critical features. These boundary contours reduce the stair step errors. Second, the boundary contours are used as reference data for comparison during the data reduction process.

When a part consists of quadric surfaces and uniform cross-sections, the subregions can be generated using a higher extraction ratio as the threshold. However, when a part has free-form surfaces, a lower extraction ratio has to be used. If the threshold level of point extraction ratio is high for free-form surfaces, it may mistakenly eliminate cross-sectional point data that need to be maintained.

4.5. Reduction of point data

For point data reduction, the points on a horizontal cross-sectional contour have to be compared with those of the neighboring boundary cross-sections. If the deviation is within the given tolerance, the points in that contour can be eliminated; otherwise, they need to be kept. In order to compare the point data with efficiency, a homotopy method is introduced.

4.5.1. Homotopy

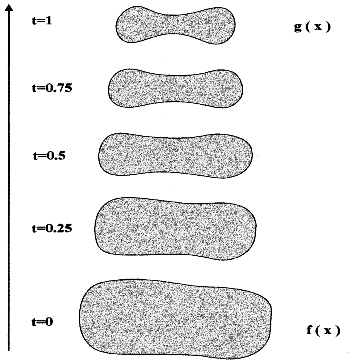
Intermediate contours between the boundary contours can be generated using a homotopy. A homotopy is defined as follows [22].

Let f, $g : X \rightarrow Y$ be maps, where X and Y are topological spaces. Then f is homotopic to g if there exist a map $F : X \times I \rightarrow Y$ such that $F(x, 0) = f(x)$ and $F(x, 1) = g(x)$ for all points $x \in X$ where $I = [0, 1] \subset R$. The map F is called a homotopy from f to g. When F is defined by $F(x, t) = (1 - t)f(x) + tg(x)$, it is called a straight-line homotopy. Fig. 5(a) illustrates the concept of the straight-line homotopy.

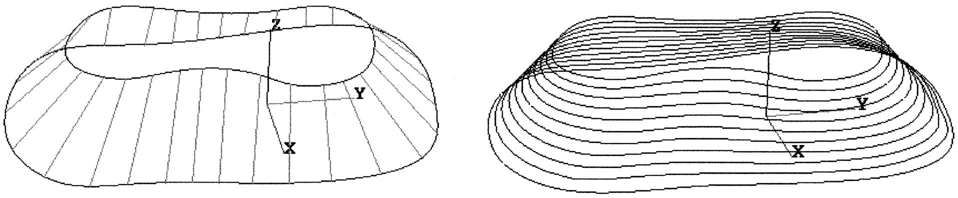
The homotopy can be used in different ways. The homotopy is used to generate intermediate contours. But a surface that connects two boundary contours can be represented as well by the locus of homotopy that transforms one contour to the other. Fig. 5(b) and (c) show boundary contours and the corresponding intermediate contours generated using the homotopy, respectively.

4.5.2. Elimination of point data at intermediate contours

After the point data model is divided into subregions, B-spline curves are generated by approximating the points in the boundary contours. Then, a mid-contour between the consecutive boundary contours is calculated using the homotopy. Distances between the mid-contour and the corresponding cross-sectional point data are calculated. When the deviation of each point from the mid-contour is within tolerance, all cross-sectional point data located at the same *z*-height are eliminated. If the deviation is greater than the tolerance, these cross-sectional point data cannot be removed and they become the boundary contours of new subregions. The process of halving the boundary contours continues until the height of the subregion is smaller than two times the minimum allowable layer thickness, l_{min}. To check the tolerance, both the maximum allowable deviation and the average deviation are used, with these values being set by the user. The algorithm terminates when no further subregioning can be performed. Since the algorithm only checks for the mid-contour in examining the deviation between the point data and the intermediate contours, it may miss some contours that are out



(a) Homotopy



(b) Boundary contours (c) Generation of intermediate contours

Fig. 5. Homotopy and its examples: (a) homotopy, (b) boundary contours, (c) generation of intermediate contours.

of tolerance. This can be avoided by increasing the number of initial subregions by lowering the level of the point extraction ratio as described in the earlier section.

Fig. 6 shows the point data elimination procedure. Fig. 6(a) shows the initial point cloud model that is divided into three subregions using the point extraction ratio. In subregions I and II, the intermediate cross-section point data are represented by the homotopy of the boundary curves. Therefore, all the point data except the ones at the boundary curves are eliminated. In subregion III, however, the mid-contour cannot be eliminated since it is bigger than the tolerance when compared with the boundary curves, and additional subregions are needed. Fig. 6(c) shows the final point data model obtained upon completion of the data elimination procedure. We call this model a contour data model since each contour is represented by the point data. It can be stored and used for slice data file generation for RP.

5. Slice data file generation and part fabrication

A slice data file contains contour data of all layers that are required in fabricating an RP part. The contour data model created in the previous section can be used for this purpose.

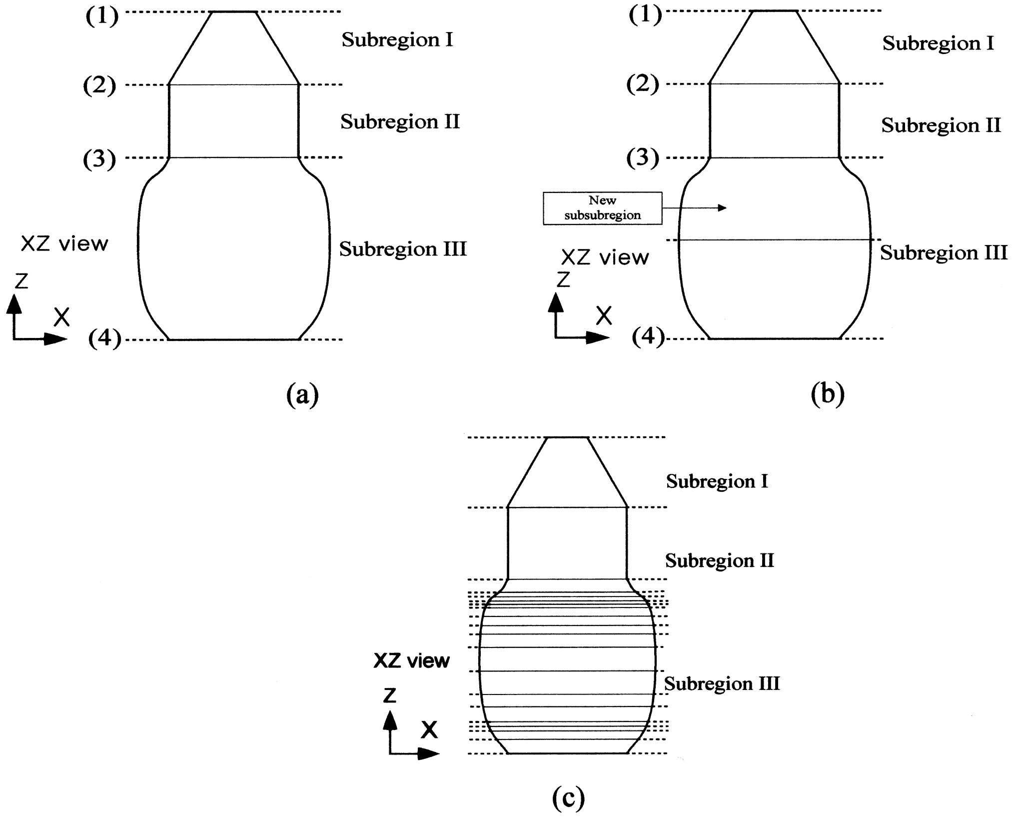


Fig. 6. Point data elimination procedure.

Using the contour data model directly made from the point cloud data, the STL file creation step can be eliminated. The contour data model consists of boundary contours of subregions that remained in the data reduction process. But due to point data reduction, some intervals between consecutive contours are much larger than the layer thickness of a machine. The slice data from these contours cannot be used for fabricating a part since RP machines have limitations in terms of their maximum curing depth. For these thick slices, additional intermediate contours are generated between the boundary contours. These intermediate contours need to be generated without disrupting the continuity of the surface. The straight-line homotopy is used to generate these additional contours between boundary contours. The number of contours generated at this stage coincides with the number of layers in RP fabrication. This contour information constitutes the slice data file, which is finally transferred to the RP machine.

Different slice file formats exist for different types of RP machines, but the slice data file basically consists of a layer file and a scan file. The layer file contains the contour curves of all layers and the scan file contains the laser path in the contour of each layer.

In our research, several RP parts were fabricated using the slice data without the use of STL files. In the next section, the proposed method is applied to example parts and the results are discussed.

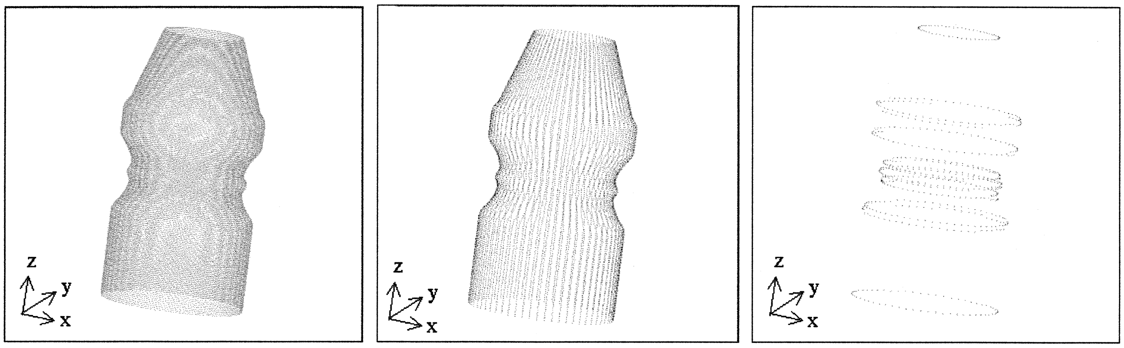
6. Application examples

In implementing the proposed procedure, a laser scanner, Surveyor Model 1200 from Laser Design Inc., was used to acquire the point cloud data and an RP machine, the Sinterstation 2000 system of DTM Corp. was used to make RP parts using the slice data. The algorithms were developed using the open architecture of Surfacer 7.1, the CAD kernel ACIS, and C/C++ language on an SGI Indigo2 workstation. Two examples are presented to demonstrate the efficiency of the proposed procedure.

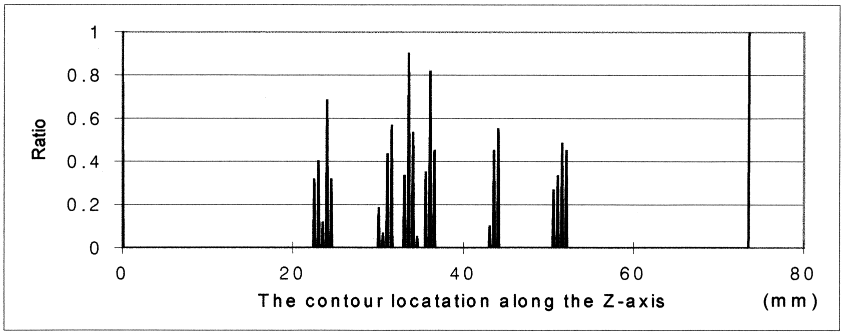
6.1. Example 1: a bottle

In the first example, a part with a bottle shape (75 mm in length) is reverse-engineered in order to illustrate the point data reduction algorithm for parts with a simple geometry. The initial point clouds of the part are shown in Fig. 7(a). The point clouds are sliced vertically by an angle of 5° around the center axis of the part, and the resulting vertical cross-sections are collected to generate the point data model shown in Fig. 7(b). Curvature changing points are identified by applying the angular deviation method and Fig. 7(c) shows the point data model recreated using only the extracted points. From this, the point extraction ratio is calculated. The extraction ratio graph for the part is shown in Fig. 7(d). This graph makes it possible for users to identify the significant contours along the z-axis.

Using the extraction ratio of 0.1 for the above graph, subregions are created as shown in Fig. 8(a). The data reduction algorithm using the straight-line homotopy is applied, and the contour model in Fig. 8(b) is obtained. Table 1 shows that 79% reduction is achieved for this

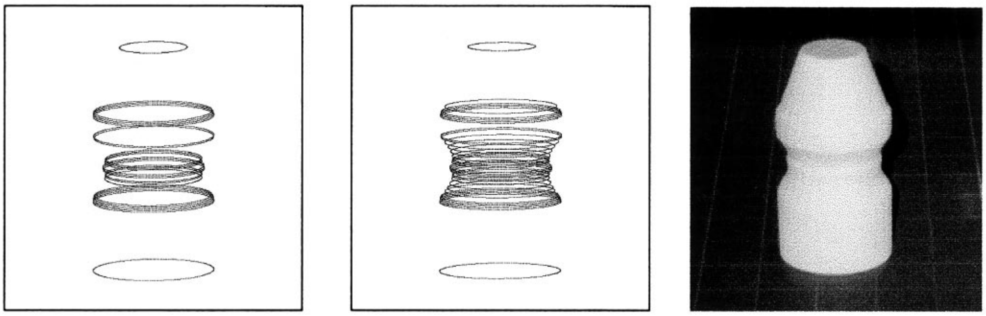


(a) Scanned data (b) Vertical Cross-sections (c) Result of Angular Deviation



(d) Extraction Ratio Graph

Fig. 7. Extraction of curvature information for the bottle part: (a) scanned data; (b) vertical cross-sections; (c) result of angular deviation; (d) extraction ratio graph



(a) Subregions (b) Contour data model (c) RP part

Fig. 8. Data reduction and RP part fabrication: (a) subregions; (b) contour data model; (c) RP part

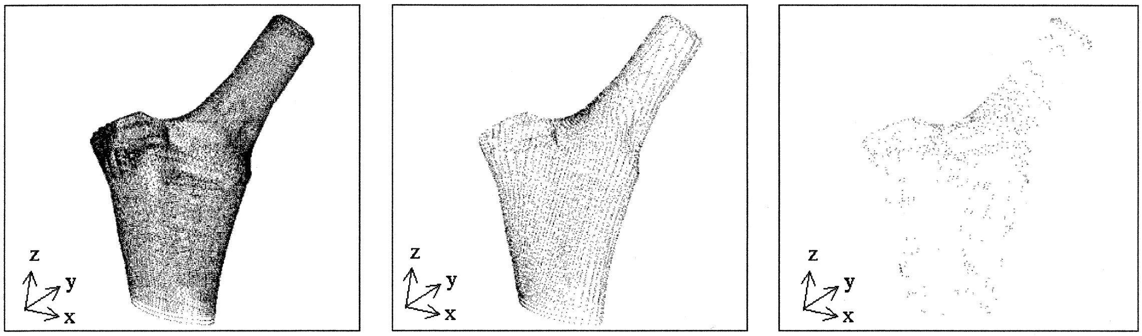
Table 1
Data reduction results for the bottle part

Item	Number of points
The initial number of point data	44,064
The number of points in the contour model	9271
The number of reduced data points	34,793
Reduction ratio	78.96 (%)

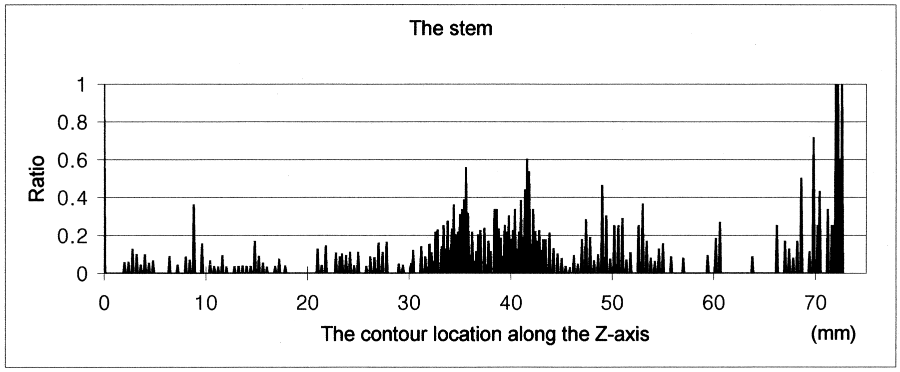
model. The slice data file is then created from the contour model, and the corresponding RP part is shown in Fig. 8(c).

6.2. Example 2: a stem

The upper stem of a hip-joint implant was scanned and manufactured as the second example. Fig. 9(a) shows the initial scanned data. In order to extract the curvature



(a) Scanned data (b) Vertical Cross-sections (c) Result of Angular Deviation



(d) Extration Ratio Graph

Fig. 9. Extraction of curvature information for the bottle part: (a) scanned data; (b) vertical cross-sections; (c) result of angular deviation; (d) extraction ratio graph.

Table 2
Data reduction results for the stem part

Item	Number of points
The initial number of point data	22,524
The number of points in the contour model	10,148
The number of reduced data points	12,376
Reduction ratio	54.95 (%)

information, vertical cross-sectioning and angular deviation methods are performed and the corresponding results are shown in Fig. 9(b) and (c), respectively. The point extraction ratio graph given in Fig. 9(d) shows wider peak areas than those for the 'bottle' case due to the free-form surfaces contained in the part. The results of data reduction are summarized in Table 2. It shows a 55% reduction of point data, which is a lower reduction, compared to the bottle case. This is due to the complicated geometry of the part. The subregions created by the extraction ratio graph, the contour data model, and the RP part made from the slice data are shown in Fig. 10.

6.3. Comparison with the STL files

In the previous two examples, it was demonstrated that the data reduction algorithm reduces the amount of point data significantly. In this section, a comparison study in terms of storage requirement and accuracy is performed between the data format used in this research and the STL file.

In this research, the slice file is generated from the point clouds without creating the STL file. To compare the proposed method with an existing method that uses STL files, an STL file was also generated from the point clouds. Fig. 11(a) shows the STL mesh data generated from the initial point clouds of 'the bottle' part. The file sizes of the part in different data formats are summarized in Table 3. Both the initial point clouds and the contour data model are stored

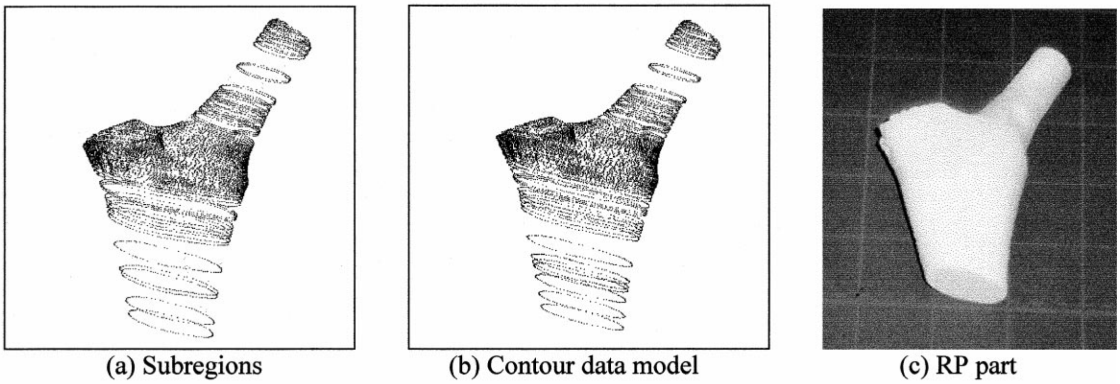
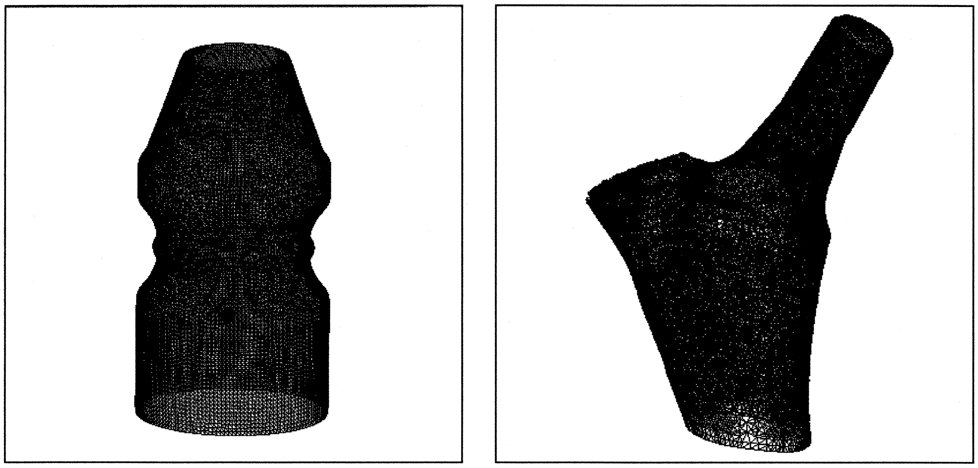

(a) Subregions
(b) Contour data model
(c) RP part

Fig. 10. Data reduction and RP part fabrication: (a) subregions; (b) contour data model; (c) RP part.



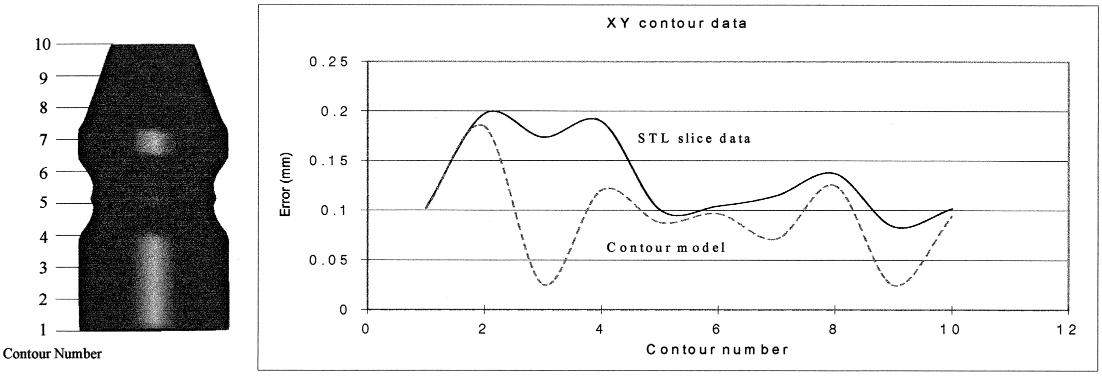
(a) Bottle part

(b) Stem part

Fig. 11. STL mesh data: (a) bottle part; (b) stem part.

in ASCII as well as in the binary format. The STL file was also generated using both formats. The results show that the file size of the contour model is much smaller than that of the STL file in both ASCII and binary formats. Similar results are also obtained for 'the stem' part shown in Fig. 11(b).

The accuracy of the contour model and that of the STL file were also compared. For the purpose of comparison, the bottle part was used again. First, 10 contours from the contour model and the same number of contours from the STL meshed model at the same heights were extracted. Both of these contours were compared, respectively, with those of the original CAD model at each corresponding height as shown in Fig. 12(a). Fig. 12(b) shows the maximum deviations of each model from the original CAD model. It shows that the contour model has less deviation than the STL meshed model from the original CAD model. Therefore, it can be



(a)

(b)

Fig. 12. Comparison of deviation of each model from the original CAD model.

Table 3
File sizes for the bottle shape part

	File sizes (bytes)	
	ASCII format	Binary format
Initial scanned data	1, 336, 424	530, 684
STL mesh data	5, 389, 162	1, 485, 884
Contour data	274, 288	113, 166

concluded that the contour data model generated by the proposed data reduction method shows better accuracy with a lower storage requirement than the one based on the STL meshed model. The STL meshed models have inherent approximation errors as discussed earlier. For high curvature surfaces, therefore, the errors of STL models will increase. In addition, the triangulated meshes are difficult to accurately generate from the scan data due to outliers, undercuts, and unordered data points. The proposed method, on the other hand, generates the slices using the contour model that makes use of the curvature information of the part, which leads to more accurate slice data files.

7. Conclusion

A novel procedure for integrating RE and RP technology is proposed. In integrating RE and RP, a surface model needs to be developed in the RE process and the STL file is generated from the surface model as the CAD input to RP systems. This existing procedure takes a considerable amount of time and expert skills. The new procedure proposes not to develop surface models for RP fabrication but rather suggests the use of contour data model created from the scanned data. In the new procedure, a point data reduction method that can be efficiently used in RE/RP integration is also proposed. The data reduction method uses the curvature information extracted from the point clouds. It also uses the straight-line homotopy method in eliminating redundant point data. The proposed method is applied to two example parts. The results show significant data reduction in both parts. It is also shown that the reduced point data models have smaller storage requirement while achieving better accuracy.

Acknowledgements

This work was supported by grant No. 981-1008-051-2 from the Basic Research Program of the Korean Science and Engineering Foundation.

References

[1] Várady T, Martin RR, Cox J. Reverse engineering of geometric models–an introduction. Computer Aided Design 1997;29(4):255–68.

[2] Gaspardo S. Applications of reverse engineering. In: Rapid Prototyping and Manufacturing '99 Conference, Rosemont, IL, vol. 3. 1999. p. 515–8.

[3] Yau H. Reverse engineering of engine intake ports by digitization and surface approximation. International Journal of Machine Tools Manufacturing 1997;37(6):871–85.

[4] Hoppe H. Surface reconstruction from unorganized points. Ph.D. thesis, University of Washington, Seattle, WA, 1994.

[5] Surfacer, User's Guide, Imageware, 1997.

[6] Hosni Y, Ferreira L, Burjanroppa R. Rapid prototyping through reverse engineering. In: Second Industrial Engineering Research Conference Proceedings. 1991. p. 420–4.

[7] Schoene C, Hoffmann J. Reverse engineering based on multi-axis digitized data. International Conference on Manufacturing Automation (ICMA '97) 1997;1:909–14.

[8] Vail NK, Wilke W, Bieder H, Juneman G. Interfacing reverse engineering data to rapid prototyping. In: Solid Freeform Fabrication Symposium. 1996. p. 481–90.

[9] Yan X, Gu P. A review of rapid prototyping technology and systems. Computer Aided Design 1996;28:307–18.

[10] Jamieson R, Hacker H. Direct slicing of CAD models for rapid prototyping. In: Internet Conference http://www.mcb.co.uk/services/confec95/rapidpd/jamieson/jamieson.htm. 1995.

[11] Qiming H, Yungan W. Direct slicing for rapid prototyping. In: International Conference on Manufacturing Automation (ICMA '97), vol. 1. 1997. p. 890–3.

[12] Makela I, Dolenc A. Some efficient procedures for correcting triangulated models. In: Solid Freeform Fabrication Symposium. 1993. p. 126–34.

[13] Rock SJ, Wozny MJ. Generating topological information from a "bucket of facets". In: Solid Freeform Fabrication Symposium. 1992. p. 251–9.

[14] Bohn JH, Wozny MJ. Automatic CAD-model repair: shell-closure. In: Solid Freeform Fabrication Symposium. 1992. p. 86–94.

[15] Fujimoto M, Kariya K. An improved method for digitized data reduction for computer integrated manufacturing. In: Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation. 1992. p. 896–901.

[16] Hamann B. A data reduction scheme for triangulation surfaces. Computer Aided Geometric Design 1994;11:197–214.

[17] Chen YH, Ng CT, Wang YZ. Data reduction in integrated reverse engineering and rapid prototyping. International Journal of Computer Integrated Manufacturing 1999;12(2):97–103.

[18] Véron P, Léon JC. Static polyhedron simplification using error measurements. Computer Aided Design 1997;29:287–98.

[19] Hamann B, Chen J. Data point selection for piecewise linear curve approximation. Computer Aided Geometric Design 1994;11:289–301.

[20] Martin RR, Stroud IA, Marshall AD, Data reduction for reverse engineering. RECCAD, Deliverable Document 1 COPERNICUS Project No. 1068, 1996, pp.101–113.

[21] DataSulpt User's Manual Version 4.0. Laser Design, 1995.

[22] Kunii TL, Shinagama Y, Gotoda H. Intelligent design and manufacturing. New York: Wiley, 1992.