Error-based segmentation of cloud data for direct rapid prototyping

G.H. Liu, Y.S. Wong*, Y.F. Zhang, H.T. Loh

Department of Mechanical and Production Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore, Singapore 119260

Received 21 June 2001; revised 22 March 2002; accepted 25 March 2002

Abstract

This paper proposes an error-based segmentation approach for direct rapid prototyping (RP) of random cloud data. The objective is to fully integrate reverse engineering and RP for rapid product development. By constructing an intermediate point-based curve model (IPCM), a layer-based RP model is directly generated from the cloud data and served as the input to the RP machine for fabrication. In this process, neither a surface model nor an STL file is generated. This is accomplished via three steps. First, the cloud data is adaptively subdivided into a set of regions according to a given subdivision error, and the data in each region is compressed by keeping the feature points (FPs) within the user-defined shape tolerance using a digital image based reduction method. Second, based on the FPs of each region, an IPCM is constructed, and RP layer contours are then directly extracted from the models. Finally, the RP layer contours are faired with a discrete curvature based fairing method and subsequently closed to generate the final layer-based RP model. This RP model can be directly submitted to the RP machine for prototype manufacturing. Two case studies are presented to illustrate the efficacy of the approach.

Keywords: Rapid product development; Reverse engineering; Rapid prototyping; Cloud data; Segmentation

1. Introduction

The goals in rapid development of products with complex surfaces are towards: (1) substantial time reduction of product development; and (2) cost-effective manufacturing of small batch-size products and a variety of products with complex surfaces [1,2]. To satisfy the demanding requirements of rapid product development (RPD), reverse engineering (RE) and rapid prototyping (RP) are two important technologies that have drawn much research interests.

RE refers to the process of creating a CAD model from a physical part or prototype [3]. In the conventional product development cycle, the initial conceptual or aesthetic design of sculptured surfaces is often described by stylists who formalize their ideas by making clay or wooden models. Recently, RE is increasingly employed in medical applications. For example, physical models are molded from a part of the body, such as the stump of an amputated limb, and an appropriate model is then captured and developed from the mold and used to custom fit an artificial limb. Necessary CAD models can be generated using RE technology based on the corresponding geometric database, which is usually created by capturing the shape information from the original physical model with mechanical contact or optical non-contact measuring techniques. The mechanical contact measurement equipment, such as a coordinate measurement machine (CMM) equipped with a touch probe, is an important digitizing tool for data acquisition because its measurement accuracy can achieve up to 10 μm or better. However, compared with the optical non-contact measurement equipment, such as a laser scanner, the contact measurement speed is very slow. Furthermore, the measurement tends to be time-consuming and even difficult for free-form surfaces. A laser scanner is an effective digitizing device for 3D geometric shapes, from which sufficient view scans of an object can be executed rapidly and a so-called cloud data is obtained. Fig. 1 shows a cloud data set obtained by digitizing a facial mask with a laser scanner. The data set is combined with four range data patches. The darkest and lightest patches are from low-resolution scans of the part. The other two patches are from high-resolution scans, particularly for the mouth, nose and eye areas. This cloud data contains 104,175 points. In the RE process, the faster non-contact optical measuring techniques have significantly reduced the time for data acquisition. However, the captured cloud data is usually highly dense, randomly distributed and partially overlapping. Rapid creation of a CAD model from such a data set is still a difficult problem [4,5].
RP is an emerging technology to build parts layer by layer [1]. The use of RP can significantly reduce not only the prototyping time, but also the impact of the geometric complexity of the part on the fabrication process. RP involves a process of faceting the 3D object and then extracting cross-sections from the faceted model (STL model). However, in current CAD systems, the resulting STL model suffers from topological problems such as degenerated facets and undesirable gaps or flipped normals, leading to incomplete cross-sections that cannot be manufactured as layers [6]. Time-consuming processes to repair the facets are therefore required. Moreover, when a higher resolution is required, the faceting time becomes longer and a much larger STL file is generated, which may cause storage and computing problems.

On the other hand, current RE and RP processes rely primarily on two different CAD representations, and therefore are two separate stages. In RE, a fitting algorithm is applied to reconstruct surface representation from the sampled cloud data. In RP, a faceting algorithm is applied to triangulate the surfaces for mesh representation. These two processes result in rather expensive remodeling computations, large file transmissions, and highly accumulated approximation errors.

Hence, to fully integrate RE and RP for an efficient RPD, this paper proposes a novel point-based curve representation, from which the RP compatible layer contours can be directly extracted. An error-based segmentation algorithm is also developed to accomplish a direct RP of cloud data. In this process, neither a surface model nor an STL file is generated. The overall procedure is shown in Fig. 2. Case studies are presented to illustrate the efficacy of the proposed algorithm.

2. Related work

Approaches to transforming a cloud data to a CAD representation can be classified into three categories: triangular polyhedral mesh based method, segment-and-fit based method, and direct prototyping method [2,3,7].

In the first method, an initial triangular mesh is constructed to capture the unknown topological structure of the scattered data. The mesh is then optimized to reduce redundant vertices and thereafter a curvature-continuous surface is reconstructed. Many triangulation techniques have been reported, e.g. delaunay triangulation algorithm [8,9], triangulation based on signed distance function [5] and triangulation based on α-shapes [10]. Triangulation for cloud data is, however, a computationally inefficient process [5,11].

In the segment-and-fit method, the cloud data is divided into a suitable patchwork of surface regions to which an appropriate single surface is fitted [4,12]. Chen and Liu [13] proposed a layer-based segmentation technique based on 2D planar curve using a filtered approximate curvature analysis. Milroy et al. [12] presented a semi-automatic edge-based segmentation method using Darboux frame for edge detection and active contours for edge linking. Extending the same method, Yang and Lee [4] employed a parametric quadric surface approximation, instead of Darboux frame, to identify the edge points. A different boundary-detection method was proposed by Sarkar and Menq [14] based on an image processing technique, in which the z-coordinate at each point replaces the grey level intensity at the pixels of an image. Recently, Liu and Ma [15] proposed a two-stage approach for high-level segmentation of 3D CT-contour points using a layer-based seed-growing strategy. Nevertheless, present segmentation algorithms are sensitive, computationally complex or can only be applied to simple topology data [4,15]. Hence, intelligent division of the entire measured data points into regions according to shape-change detection has been a challenging and long-standing research issue [4].

Unlike the earlier two methods, the direct prototyping method avoids the process of surface reconstruction. A point-based model (curve or mesh model) is first constructed from the cloud data and then served as the direct

![Fig. 1. Cloud data of facial mask.](image)

![Fig. 2. A flowchart of the proposed approach.](image)
input to machines for fabrication. Because surface reconstruction is generally a computationally complex and time-consuming process, this point-based method is faster and attracting greater attention in recent years [2,6,11].

Approaches for direct prototyping of cloud data can be categorized into two directions, i.e. direct NC machining and direct RP. For direct NC machining, the commonly used approach is based on a Z-map model [11] or a mesh model [16,17]. In the former approach, the z-coordinates of the densely distributed data points in a grid space are conserved in a matrix called the Z-map. The Z-map is then offset to generate the corresponding cutter-path. In the latter, the gouging-free cutter-path is directly planned based on the mesh model and generated by using some automatic collision detection algorithms. For direct RP, in essence, this is an integration of RE and RP. Some direct RP techniques have been proposed. Lee and Woo [2] proposed a cross-section based, followed by a curvature-based compression, strategy to process regularly scanned cloud data for direct RP. Fischer [6] proposed a hierarchical LOD (multi-level of detail) structure to bridge the mesh model constructed in RE for a multi-level RP. However, little work has been done for the direct RP of a dense, random and overlapping cloud data. Hence, in this paper, we are concerned with the work in this aspect.

3. Cloud data segmentation

Cloud data segmentation is to extract RP layer contours directly from cloud data by constructing an intermediate point-based curve model (IPCM), such that the layer-based RP model that is submitted to RP machines can be rapidly generated. This is accomplished in three main stages: data preprocessing (data subdivision and reduction), data modeling (IPCM construction), and extraction of the RP layer contours.

3.1. Data preprocessing

The random cloud data is an extremely large set of data and therefore, must be appropriately thinned prior to further modeling. A conventional reduction method for cloud data is the voxel binning method [18], where the data is first bounded with a box and subdivided into many small regions with uniform cubes (bins). Then, for each bin, only the point closest to the center of the bin is retained. The voxel binning method is simple and highly efficient. However, the bin size is arbitrarily user-specified. This may not ensure necessary accuracy between the constructed model and the original data. Sun et al. [19] improved this method by employing a local surface interpolation to automatically determine the best bin size. However, their method is not computationally efficient and can only handle simple surface data. Here, we proposed a new and highly efficient method for cloud data reduction, in which the cloud data is firstly subdivided into many regions and then reduced by individually compressing the data in each region.

As the cloud data is a dense data set, if the subdivision accuracy is well controlled, the shape represented by the points of each region can be considered as a curve or multi-curves. The basic shape of a curve is exhibited by its feature points (FPs), i.e. corners or high curvature points [20]. Hence, our data reduction is to search and select all the FPs in the region and remove the rest. Randrup [21] employed a digital image reduction technology to process projection data for a ruled surface approximation, which turned out a so-called skeleton of the digital image that could be used to construct the necessary directrix curves. We also apply this approach to deal with the data reduction for each region.

3.1.1. Data subdivision

The aim of data subdivision is to convert the 3D cloud data into a set of 2D regions by adaptively subdividing the cloud data with a cluster of parallel planes. This cluster of subdivision planes is defined according to a user-specified reference plane. Denoting the user-specified reference plane and its normal direction as \(T\) and \(\mathbf{n}\), respectively, we take the facial mask cloud data as an example to explain the determination of the reference plane for the cluster of the subdivision planes. As shown in Fig. 3(a) and (b), the variation of the basic shape of the cloud data can be observed from the side view of \(YcZc\) and \(ZcXc\). The reference plane can be chosen as planes \(XcYc\), \(YcZc\), and \(ZcXc\), respectively. \(YcZc\) and \(ZcXc\) are better than \(XcYc\) because the latter would give rise to multi-curves in some subdivided regions, which is less likely with the former two. \(ZcXc\) is better than \(YcZc\) because \(ZcXc\) would ensure that most of the generated subdivision planes would be perpendicular to the main variation of the basic shape of the cloud data; hence, would be more effective in keeping the original shape of the cloud data in the subdivided regions. Therefore, among these three reference planes, \(ZcXc\) is the best to define the cluster of subdivision planes.

The cloud data is first subdivided into several initial regions based on the defined reference plane \(T\) and its normal direction \(\mathbf{n}\). The user determines these after examining the visually observable shape of the cloud data. Then, for each region of the data set, a new subdivision plane passing the central position of the region is generated, as shown in Fig. 3(a). This plane is also treated as the projection plane for the data points of the region. Denoting the region of data set and the corresponding central position as \(\mathbf{P}\) and \(\mathbf{O}\), respectively, the distance from a given point in the region, \(\mathbf{p} \in \mathbf{P}\), to the projection plane \(T\), can be expressed as:

\[
\text{dist}(\mathbf{p}, T) = \|\mathbf{n}^T(\mathbf{p} - O)\|
\]

(1)

We estimate the average projection error of the data points in the region to the projection plane with the following error
indicator:

$$\text{ave}_\text{proj} \text{err} = \left[ \frac{1}{n} \sum_{p \in P} (\text{dist}(p, T))^2 \right]^{1/2}$$ (2)

where $n$ is the total number of the points in the region. In the subdivision algorithm, if the calculated average projection error of the points in the region is greater than a given subdivision error, this region is further subdivided into two halves from the central position of the region. This subdivision process continues for each subdivided region until all the average projection errors of the regions are within the desired subdivision error. Thus, the original 3D cloud data is converted to a set of 2D subdivided regions, as shown in Fig. 3(c).

3.1.2. Data reduction

A digital image technology is employed for data reduction. The related concepts are presented in Appendix A. For the data points of each subdivided region, we consider a grid structure in the projection plane. Each square in this grid is associated with the value 1 (black point) if the intersection between the square and the projection of the point in the region is not empty. Otherwise, it is 0 (white point). In practice, the square size of the grid is normally chosen to be sufficiently large to avoid holes (white points among black points) in the digital plane [21]. The digital plane of the projection plane that maps to the 3D points in the region is thus constructed, as shown in Fig. 4(a).

Data reduction is firstly employed to derive a skeletal curve from the black points in the digital plane by removing all the black points that are deletable. Here, we employ 28 of...
When all the deletable black points in the digital plane have been removed, a skeletal form of the digital image is generated, as shown in Figs. 4(c) and 5(a). We select 3D points in the region that correspond to the centers of the black points (black squares) and define them as the FPs of the region (Fig. 5(a)). Based on the 2D topology of the black squares of the skeleton in the digital plane, the selected FPs are ordered and connected. A 3D skeletal curve is therefore generated, as shown in Fig. 5(a). This generated skeletal curve can be used to roughly cover (or represent) the basic shape of the region because some of the important shape points of the region may not be included in these selected FPs. This is due to the determination of the square size of the grid for the digital plane. Hence, the following proposed maximum-distance-error (MDE) based algorithm has been further implemented to restore the potentially ignored shape points from the rest of the points in the region.

The proposed MDE algorithm is based on the generated 3D skeletal curve and is to check whether the rest of the points in the region are correlated to this curve within a user-defined error bound \( \pm \epsilon \). For example, as shown in Fig. 5(b), suppose that \( P_k \) is one of the rest points in the region and, its most correlated point in the skeletal curve is \( P_i \) (this can be achieved by calculating the minimal Euclidean distance), the MDE algorithm first calculates the distances between \( P_k \) and two segments \( P_{i-1}P_i \) and \( PP_{i+1} \) and then compares the larger one with \( \epsilon \), if the distance is less than \( \epsilon \), \( P_i \) is termed as a redundant point. Otherwise, it is defined to be uncorrelated to the skeletal curve. In this way, all the remaining points, except the FPs in the skeletal curve, can be classified to be either redundant or not. All the redundant points are then removed. Those retained points, together with the FPs in the skeletal curve, are considered as the new set of FPs of the region. The shape of the region is depicted with these FPs.

Thus, the dense unstructured cloud data is compressed and segmented into regions of FPs. Fig. 6 shows an application example. In this example, we have only one region of points. The number of the points in the original data is 1004. The shape tolerance and the grid size are chosen to be 0.1 and 2 mm, respectively. After compression, the number of the retained FPs is 67 (Fig. 6(c)). Following the MDE algorithm, the final number of the points is increased to 115, i.e. 48 additional FPs have been restored (Fig. 6(d)). The presented thinning algorithm is highly efficient and will be discussed further in Section 4. Note that we use subdivision error to control the thickness of the subdivided regions and the shape tolerance to control the shape accuracy of each region. When the method is applied to reduce the cloud data, the user-defined subdivision error cannot be too small, because this may lead to the cloud data being subdivided into too many regions. In addition, some regions may not contain sufficient points, such that a skeletal curve cannot be derived and the reduction becomes similar to voxel binning.

3 × 3 thinning templates (Appendix A) proposed by Jang and Chin [22] to compress the data of each region, which is a parallel thinning method. The 28 thinning templates define 28 kinds of black points that can be removed from the digital plane. In the thinning process, black points in the digital plane are firstly compared with the templates. Once a compared black point matches one of the defined templates, the black point is removed (to be a white point). The comparison is then resumed. This process iterates until no deletable black point can be identified from the digital plane with the defined thinning templates. In practice, to improve the efficiency and robustness of the thinning algorithm, the defined 28 thinning templates, \( \Gamma \), are classified into three levels in this paper, i.e. the digital plane is compressed using the templates with a priority of \( \Gamma_1 > \Gamma_2 > \Gamma_3 \), as shown in Table 1.

Compared with the reduction method used by Randrup [21], our proposed method is more efficient in thinning the data and is robust. The reduction results are compared and shown in Fig. 4(b) and (c).

![Diagram](image)

**Fig. 5.** Data reduction and the retrieving of new FPs by MDE: white dots represent the FPs of the skeletal curve and black dots represent the rest of the points in the region.

![Diagram](image)

**Table 1.** Level-based compression templates

<table>
<thead>
<tr>
<th>Configuration of B(P)</th>
<th>Template (( \Gamma_1 &gt; \Gamma_2 &gt; \Gamma_3 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \Gamma_1 )</td>
</tr>
<tr>
<td></td>
<td>(224,56,14,131)</td>
</tr>
<tr>
<td></td>
<td>(52,22,13,133)</td>
</tr>
<tr>
<td></td>
<td>(225,120,30,15)</td>
</tr>
</tbody>
</table>

When all the deletable black points in the digital plane have been removed, a skeletal form of the digital image is generated, as shown in Figs. 4(c) and 5(a). We select 3D points in the region that correspond to the centers of the black points (black squares) and define them as the FPs of the region (Fig. 5(a)). Based on the 2D topology of the black squares of the skeleton in the digital plane, the selected FPs are ordered and connected. A 3D skeletal curve is therefore generated, as shown in Fig. 5(a). This generated skeletal curve can be used to roughly cover (or represent) the basic shape of the region because some of the important shape points of the region may not be included in these selected FPs. This is due to the determination of the square size of the grid for the digital plane. Hence, the following proposed maximum-distance-error (MDE) based algorithm has been further implemented to restore the potentially ignored shape points from the rest of the points in the region.

The proposed MDE algorithm is based on the generated 3D skeletal curve and is to check whether the rest of the points in the region are correlated to this curve within a user-defined error bound \( \pm \epsilon \). For example, as shown in Fig. 5(b), suppose that \( P_k \) is one of the rest points in the region and, its most correlated point in the skeletal curve is \( P_i \) (this can be achieved by calculating the minimal Euclidean distance), the MDE algorithm first calculates the distances between \( P_k \) and two segments \( P_{i-1}P_i \) and \( PP_{i+1} \) and then compares the larger one with \( \epsilon \), if the distance is less than \( \epsilon \), \( P_i \) is termed as a redundant point. Otherwise, it is defined to be uncorrelated to the skeletal curve. In this way, all the remaining points, except the FPs in the skeletal curve, can be classified to be either redundant or not. All the redundant points are then removed. Those retained points, together with the FPs in the skeletal curve, are considered as the new set of FPs of the region. The shape of the region is depicted with these FPs.

Thus, the dense unstructured cloud data is compressed and segmented into regions of FPs. Fig. 6 shows an application example. In this example, we have only one region of points. The number of the points in the original data is 1004. The shape tolerance and the grid size are chosen to be 0.1 and 2 mm, respectively. After compression, the number of the retained FPs is 67 (Fig. 6(c)). Following the MDE algorithm, the final number of the points is increased to 115, i.e. 48 additional FPs have been restored (Fig. 6(d)). The presented thinning algorithm is highly efficient and will be discussed further in Section 4. Note that we use subdivision error to control the thickness of the subdivided regions and the shape tolerance to control the shape accuracy of each region. When the method is applied to reduce the cloud data, the user-defined subdivision error cannot be too small, because this may lead to the cloud data being subdivided into too many regions. In addition, some regions may not contain sufficient points, such that a skeletal curve cannot be derived and the reduction becomes similar to voxel binning.
3.2. IPCM construction

For modeling, we propose a novel point-based curve model (IPCM) instead of a more commonly used triangular mesh model, which is simpler but effective and more computationally efficient. The data structure for the IPCM is shown in Fig. 7. In this data structure, pointers head and tail are used to record the FPs in the same region, while pointers front and next are used to connect FPs in different regions to generate the IPCM. The construction of the IPCM starts...
from one of the end regions (the front most or the back most) and is a progressively growing process. This growing process consists of two stages: the first is across the direction of the regions, and the second (reversed growing) is in the reversed direction across the regions, both of which are carried out region by region.

We refer to Fig. 8 for the elaboration on the algorithm. Suppose that the cloud data is compressed into five regions of FPs, i.e. R1, R2, R3, R4, and R5 where R1, R5 are the two end regions. The growing of IPCM starts from R1 and is firstly carried out in the direction front_end → back_end (Fig. 8(b)) and then in the reversed direction of back_end → front_end (Fig. 8(c)). For an FP in R1, its next neighboring FP is searched from all the FPs in the succeeding four regions (R2, R3, R4, and R5) within a user-defined tolerance sphere. Once a FP that is closest to the given FP is retrieved, the two FPs are connected with straight-line segment and the IPCM between R1 and R2 is generated. This growing of IPCM continues to FPs in R2. Likewise, by retrieving all the FPs in the succeeding three regions (R3, R4, and R5) within the tolerance sphere, the closest one is selected and connected as the next neighboring FPs of the given FPs in R2, and thereby the IPCM between R2 and R3 is generated. In this way, region by region, an initial IPCM across the direction of regions: front_end → back_end, is obtained.

The reversed growing of the IPCM is to further supplement the connection of the FPs in different regions that may be bypassed in the first growing due to different amounts of FPs in different regions. In this stage, the starting region is R5 and the reversed direction is back_end → front_end. For a FP in R5, if its front neighboring FP is still empty, all the FPs in the preceding four regions (R4, R3, R2, and R1) are retrieved within the defined tolerance sphere and, the FP that is closest to the given FP is continuously connected as the corresponding front FP. As in the first growing, region by region in a reversed order, the second
growing continues until region $R_1$ is reached. Then the whole growing process is completed and the final IPCM is constructed.

Compared with the conventional triangular mesh model for cloud data, the proposed IPCM model is much simpler but effective when it is applied to extract layer contours for RP. An example of IPCM that is constructed from a carton toy cloud data is shown in Fig. 9. In this example, the cloud data is subdivided with a cluster of plane $XcYc$ and the IPCM is constructed along the direction of $Zc$.

3.3. Extraction of RP layer contours

The RP layer contours are extracted by intersecting the constructed IPCM with a set of user-defined planes (slicing planes) that are parallel and uniformly distributed. The number of RP layers is determined according to a user-given layer thickness, which corresponds to the allowable building thickness in RP. Suppose that the cluster of slicing planes (intersection planes) is positioned with a sequence of $z$-coordinates: $Z = (z_0, z_1, ..., z_n)$ and $z_0 < z_1 < \cdots < z_n$, the value of $z_k \in Z$ is determined based on the maximum and minimum $z$-coordinates of the FPs in the IPCM. As shown in Fig. 10(a), denoting the user-given layer thickness, the maximum and minimum $z$-coordinates of the IPCM as $z_{\text{pitch}}$, $z_{\text{min}}$, and $z_{\text{max}}$, respectively, the value of $z_k \in Z$ is calculated as given below:

$$z_0 = z_{\text{min}}, \quad z_n = z_{\text{max}}, \quad z_k = z_0 + i \frac{z_n - z_0}{z_{\text{pitch}}}$$

with $i = 1, \ldots, n - 1$

The intersection between the slicing planes and the constructed IPCM is determined by comparing the sequence $Z$ with the $z$-coordinates of the FP pairs (FP with its front
neighboring FP or FP with its next neighboring FP) in the IPCM. For example, supposed that the z-coordinate of a FP pair is $z_{FP}$ and $z_{neighboring}$, respectively, and if there exists a value $z_k \in \mathbb{Z}$, such that $z_{neighboring} \leq z_k \leq z_{FP}$ or $z_k \leq z_{FP}$, then there exists an intersection between the slicing plane and the IPCM. The intersection point is calculated by linearly interpolating the two FPs of the pair.

As the algorithm retrieves all the FPs of the IPCM to check for the intersection, repeated intersection points may be produced because the two FP in the pair are not ordered, e.g. FP pair (FP1, FP2) and (FP2, FP1). To avoid such repetitions, a status value of either 1 or 0 is applied in the IPCM data structure. The proposed extraction algorithm is described as follows:

1. Initialization. Retrieve all the FP pairs of the IPCM by checking whether the following condition is satisfied:

$$\{ \text{FP} \rightarrow \text{next} = \text{FP}' \quad \text{or} \quad \text{FP} \rightarrow \text{next} = \text{NULL} \}$$

If the above condition is satisfied, the status value of the FP is initialized as $\text{FP} \rightarrow \text{status} = 0$, otherwise $\text{FP} \rightarrow \text{status} = 1$. For $\text{FP} \rightarrow \text{status} = 0$, the FP with its front neighboring FP is used to form a FP pair and compared with the $Z$ sequence to determine the intersection. For $\text{FP} \rightarrow \text{status} = 1$, both the FP with its front neighbor and the FP with its next neighbor, are each to form a FP pair and make a comparison with the sequence $Z$.

2. Calculate the layer contour points (intersection points). Suppose that an intersection is determined between plane $z_k \in \mathbb{Z}$ and the IPCM, the identified two FPs are $\text{FP}_i (R_m)$ and $\text{FP}_i (R_n)$, and the corresponding $z$-coordinates are $z_m$ and $z_n$, as shown in Fig. 10(b), the intersection point $P_i (z_k)$ for the RP layer $z_k$ is calculated as follows:

$$P_i (z_k) = \frac{Z_k - Z_m}{Z_n - Z_m} \text{FP}_i (R_m) + \frac{Z_n - Z_k}{Z_n - Z_m} \text{FP}_i (R_n) \quad (4)$$

3. Generate the layer contours. The intersection points having the same $z$-coordinate correspond to the contour points of a RP layer. By sorting and connecting these 2D points, an RP layer is generated. However, due to the processing errors in the segmentation, the extracted contour/contours of each RP layer may not be sufficiently smooth. Hence, a discrete curvature based fairing algorithm has been developed to automatically smoothen these layer contour/contours [23].

Fig. 11 shows an IPCM constructed from a facial cloud data and two different kinds of extracted RP layer contours that have been fairied. In this example, the cloud data is
subdivided with reference plane $X_cY_c$ and the IPCM is constructed in the direction of $Z_c$. The reference planes for the slicing planes in Fig. 11(b) and (c) are plane $X_cY_c$ and plane $Z_cX_c$, respectively. With the user-defined layer thickness of 1 mm, the total number of the extracted RP layers in Fig. 11(b) is 242 and in Fig. 11(c) is 48. Compared with these two RP layers, Fig. 11(b) would produce a more accurate RP part than Fig. 11(c). This is because the shape resolution in Fig. 11(b) is much higher than that in Fig. 11(c), although the RP layers in Fig. 11(c) could provide a shorter time. Note that the criteria for the generation of subdivision planes and slicing planes are different. Hence, in the application, when a high resolution of RP part is expected, the best reference plane for the slicing planes should be chosen to be the same as that for subdivision planes.

4. Direct RP

Direct RP refers to direct building of the part from the cloud data without constructing a surface representation and generating an STL file. The aforementioned faired RP layer contours cannot be directly sent to the RP machines for fabrication because the extraction cannot ensure all the layer contours to be closed. Hence, the contour/contours of each RP layer has/have to be checked and closed before the RP fabrication. We termed the complete set of RP layer contours that are faired and closed as the layer-based RP model. This model serves as a direct input to the RP machine.

The algorithms for direct RP of the cloud data have been implemented using C/C++ on a HP-C200 workstation in the Unigraphics environment. In this paper, two case studies are presented to illustrate the efficacy of the proposed algorithms. The chosen experimental RP machine is a Sony Solid Creator JSC2000 SLA machine. In the experiment, the intensity of the laser power is set at 0.250 W and the prototyping material is SCR-310 photo polymer. The generated layer-based RP model is directly sent to the SLA machine for prototype manufacturing.

The first case is a facial mask, which is composed of four range data patches. The original data cloud contains 104 175 points (Fig. 12(a)). The reference plane to define the cluster of subdivision planes and the cluster of slicing planes are all chosen as plane $X_cY_c$, as shown in Fig. 12(a). Based on this defined reference plane, the cloud data is subdivided and compressed and, a faired layer-based RP model is generated along the direction of $Z_c$, as shown in Fig. 12(b) and (c). For the part, the shape tolerance and the layer thickness are, respectively, defined as 0.1 and 0.3 mm. The total number of the layers of the generated RP model is 807. This RP model is directly sent to the SLA machine and the prototyping time takes about 18 h. Fig. 12(d) shows the final RP product.
The second case is the cartoon toy of relatively complex features. The original data cloud contains 26,799 points. In this case, plane $X_cY_c$ is chosen to be the reference plane for both subdivision planes and slicing planes and, the direction of $Z_c$ is used to generate the final layer-based RP model, as shown in Fig. 13(a). In this case study, the original cloud data is initially user-subdivided into three regions (I, II, III) and parts of them are overlapped at the boundary areas, as shown in Fig. 13(a). This is to well preserve the original shape because the shape of region II varies sharply at some areas. In the segmentation, the subdivision errors for regions I and III are chosen to be the same and for region II, we use half the value. The compressed data and the generated RP model are shown in Fig. 13(b) and (c). For the part, the shape tolerance and the layer thickness are chosen to be 0.1 and 0.2 mm, respectively. The total number of the extracted RP layers is 367. After fairing, the generated RP model is directly sent to the SLA machine and the prototyping time is approximately 8 h. The final RP product is shown in Fig. 13(d).

In these two case studies, the proposed segmentation algorithm for direct RP has been proven to be highly efficient. The entire segmentation process took hundreds of seconds for hundreds of thousands of cloud data, while the conventional triangulation algorithm needs hundreds of minutes [24]. A summary of the performance of the algorithm is listed in Table 2.

Fig. 13. Example of cartoon toy: (a) cloud data, (b) compressed model, (c) faired layer-based RP model, and (d) RP fabricated part.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Summary for modeling of each case study</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Volume Size (mm$^3$)</td>
</tr>
<tr>
<td>Facial mask</td>
<td>$158 \times 242 \times 49$</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Carton toy</td>
<td>$74 \times 34 \times 68$</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5. Conclusions and discussion

An error-based segmentation approach for direct RP of random cloud data has been presented. The cloud data is firstly converted into a set of regions of points using an adaptive subdivision algorithm and then compressed into regions of feature points using an efficient digital image based reduction algorithm. By constructing an IPCM from the regions of feature points, a layer-based RP model can be rapidly generated through an efficient extraction algorithm and a discrete curvature based fairing algorithm. This RP model is directly sent to the RP machine for fabrication. In this process, neither a surface representation nor an STL model is generated. The proposed algorithms have been proven to be effective and highly efficient by the demonstrated case studies. However, in the proposed approach, if the user-defined subdivision error is too small, this leads to too many subdivided regions and some regions may not contain sufficient points, which would make the proposed digital image based reduction method degrade to a kind of voxel binning. In addition, the extraction algorithm is well behaved when the defined slicing planes ensure each extracted RP layer contains only one contour. However, if the RP layer contains multi-contours, as shown in Fig. 11(c), some of the multi-contours of the layer may be self-interfered after the sorting and connection of the contour points. This is mainly caused by the subdivision errors and the sharply varied shape areas in the cloud data. Currently, the self-interfered multi-contours are manually corrected and then individually fairied. A desirable extraction algorithm that can automatically handle multi-contours of the RP layer is therefore necessary in the future work.

Appendix A. Brief review of some related concepts on digital topology

The digital plane is the set of points in the Euclidean plane having integer co-ordinates. A digital set is any finite subset of the digital plane. Each digital point in the digital plane has a value of either 1 or 0 and eight neighbors, which are numbered 0–7 according to the following scheme:

\[
\begin{align*}
N_0(P) & \quad N_1(P) & \quad N_2(P) \\
N_3(P) & \quad N_4(P) & \quad N_5(P) \\
N_6(P) & \quad N_7(P) & \quad N_8(P)
\end{align*}
\]

\(N_i(P)\) is the set of all neighbors of \(P\) (without \(P\) itself) and called eight-neighbors of \(P\). This neighborhood can be uniquely characterized by a number \(B(P)\), which is given by the bit-pattern of the neighborhood configuration as follows:

\[B(P) = \sum_{i=0}^{7} N_i(P) \cdot 2^i\]

with \(N_i(P) = 0\) or 1

In a digital set \(S\), digital point \(P\) is termed as a black point if its value equals to 1 and a white point when the value is 0. Two points \(P_0\) and \(P_n\) with a common value are said to be eight-connected, if a sequence of points \(P_0, P_1, \ldots, P_n\) exists in \(S\), such that each two successive points in the sequence are 8-neighbors of each other and all \(P_i (0 \leq i \leq n)\) have the same value. Point \(P\) is deletable from \(S\) if its removal preserves the eight-connectedness of \(S\). Otherwise, the point is said to be undeletable from \(S\).

The 28 possible deletable black points within a 3 \(\times\) 3 window proposed by Jang and Chin [22] are shown in Fig. A.1. These windows are termed as thinning templates.
References


Liu Guohe is currently an engineer in the Design Technology Institute, National University of Singapore. His research work and interests include CAD/CAM/CAE and Mechatronics.

Wong Yoke San is an Associate Professor in the Mechanical Engineering Department at the National University of Singapore (NUS). His teaching and research interests are in process characterization for monitoring and control; rapid product data capture for design and manufacture; and automated and integrated manufacturing system modeling and design. In these areas, Wong has published in refereed journals and proceedings of international conferences, contributed to books, and participated in several local and overseas funded projects.

Zhang Yunfeng received his B.Eng. 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 Engineering, National University of Singapore. His research interests include reverse engineering, multi-axis tool path planning, and the computational intelligence in design and manufacturing. He is a senior member of SME.

Loh Han Tong is an Associate Professor in the Department of Mechanical Engineering at the National University of Singapore (NUS). He is the Deputy Director (Education) of the Design Technology Institute. He is also 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.