Int J Adv Manuf Technol (2006) 30: 254–260
DOI 10.1007/s00170-005-0081-2

ORIGINAL ARTICLE

Zhengji Teng · Hsi-Yung Feng · Abdullahil Azeem

Generating efficient tool paths from point cloud data via machining area segmentation

Received: 28 September 2004 / Accepted: 8 March 2005 / Published online: 12 November 2005

Abstract This paper presents a method of generating efficient three-axis ball-end milling tool paths directly from point cloud data. The primary objective is to achieve high efficiency in the machining of free-form surface geometry having isolated complex machining area. The high machining efficiency is attained by segmenting the entire machining domain into distinct areas according to the geometric complexity of the data points and by using cutters of different sizes for the segmented machining areas. An iterative numerical procedure is derived to determine the critical complexity that separates the data points with higher complexity (the complex points) from those with lower complexity (the non-complex points). A larger and more efficient ball-end mill is used to machine the area defined by the non-complex points. The gouging condition of all the data points is then evaluated with respect to the given ball-end mill. The isolated complex machining area is established by enclosing both the complex points and the gouge points. The smaller and gouge-free ball-end mill for the isolated complex machining area is subsequently selected from the standard commercial cutter series. Implementation of the presented method clearly demonstrates the high efficiency of the generated tool paths.

Keywords Complexity · Efficiency · Gouging · Point cloud data · Tool path generation

1 Introduction

In modern manufacturing, geometrically complex parts such as dies and molds are often characterized with free-form surfaces. Three-axis ball-end milling is currently the most widely used machining process to produce these complex surfaces. Many methods have been developed to generate applicable three-axis ball-end milling tool paths from the continuous mathematical representation of a 3D parametric surface. The popular iso-parametric tool path generation method was initially presented by Loney and Ozsoy [1] and later improved as an adaptive method by Elber and Cohen [2]. The cutter is driven along the iso-parametric curves on the surface. In general, this method cannot be employed to generate tool paths for compound surfaces. Iso-planar tool paths are commonly used for these compound surfaces. The cutter follows the planar cross-sectional curves generated by intersecting the design surface with a set of parallel vertical planes [3–5]. Park [6] developed a similar method named Z-constant contour machining where tool paths were generated according to a set of parallel horizontal planes. The concept of constant scallop-height machining was proposed to optimize the tool paths in terms of redundancy reduction [7–12]. The iso-phote based tool paths [13] and those to maximize the kinematic performance of the machine tool [14] were also reported. These tool paths attempted to better meet the machining tolerance requirement.

Methods to generate three-axis ball-end milling tool paths directly from discrete data point sets, that eliminate the critical challenges of reverse engineering techniques in precision surface fitting, were reported only recently. Initial work by Lin and Liu [15] generated the iso-planar tool paths based on the Z-map model of the data points. By constructing triangular meshes from the discrete data points, algorithms have been developed to generate roughing [16] and finishing [17] tool paths. Park and Chung [18] generated their iso-planar cutter location tool paths from measured data points based on 2D curve offsetting and polygonal chain intersection to avoid time-consuming 3D computations. Feng and Teng [19] presented a new tool path generation method with explicit consideration of the machining error and machined surface finish requirements. The machining error was defined and evaluated according to the adjacent tool swept surfaces of the ball-end mill. The specified scallop-height constraint was used to determine the intervals between adjacent tool paths. It should be noted

Z. Teng · H.-Y. Feng (✉) · A. Azeem
Department of Mechanical and Materials Engineering,
The University of Western Ontario,
London, Ontario, N6A 5B9, Canada
e-mail: sfeng@eng.uwo.ca
Tel.: +1-519-6612111
Fax: +1-519-6613020

that the primary objective of these previous studies is mostly to generate precision tool paths, resulting in acceptable machining errors. Increasing the overall machining efficiency, while it is one of the most important concerns in practice, has not been fully investigated.

2 Area-by-area machining

Tool paths can almost always be generated from a part surface with sufficient accuracy. The efficiency of the generated tool paths by the existing methods, however, depends on the geometric complexity of the part surface. The proposed work attempts to address one common type of surface complexity: a part surface containing an isolated complex machining area. In general, an isolated complex machining area is an island area with significantly increased convexity and/or concavity. Figure 1 shows a typical part surface containing such an isolated complex machining area in the middle. In order to meet the specified scallop-height requirement, distances between adjacent tool paths need to be constrained. The maximum allowable tool path interval in the flatter area of the part surface is always larger than that in the isolated complex machining area. As a result, intervals between tool paths in the flatter area, which also pass through the isolated complex machining area, need to be significantly reduced. This evidently creates redundant tool paths in the flatter areas and leads to low machining efficiency.

The concept of area-by-area machining of complex shaped parts has been previously proposed to increase machining efficiency. Veeramani and Gau [20] optimized the machining sequence of different surface patches and their tool entry and exit points to minimize the total machining time. Park and Choi [5] developed an extraction procedure of machining areas such as pockets and holes from a Z-map model for sculptured surface machining using the boundary extraction algorithm. The machining area boundaries were extracted and the inclusion relationships among the boundaries were identified from a run-length coded binary image. In a recent study, Ding et al. [21] proposed an adaptive iso-planar tool path generation method to increase the machining efficiency by partitioning the surfaces into different areas according to their slopes with the vertical intersecting planes based on the concept of iso-phote. Arbitrary user defined threshold values were used to partition the surfaces in these previous studies [5, 21]. Cutting tool gouging was not taken into consideration and only one tool was used to generate the tool paths for the various machining areas.

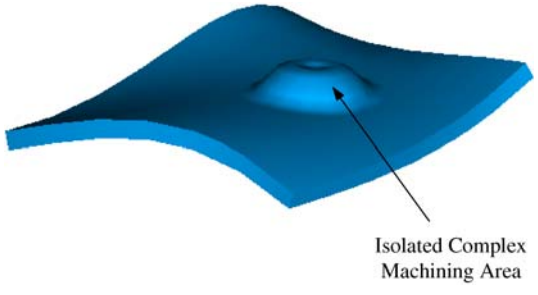

Fig. 1 A typical part surface with an isolated complex machining area

Although direct tool path generation from discrete data points eliminates the existing challenges in the reconstruction and fitting of 3D surfaces, segmentation of the point cloud data for efficient machining presents new challenges compared to the segmentation of continuous part surfaces. The normal vector and curvature at a point on the continuous surface are readily known from the mathematical expression of the surface. Nevertheless, these differential geometric properties can only be estimated with unknown estimation errors for a discrete data point set. The objective of the present work is to generate efficient tool paths from a discrete data point set by reducing redundant tool paths via machining area segmentation. The point cloud data is to be partitioned based on the geometric complexity of the data points and their gouging to the larger ball-end mill for the less complex machining area. Instead of the user defined threshold value to segment the machining areas, an adaptive procedure to automatically determine the threshold value is developed and used.

3 Efficient tool path planning

The proposed efficient tool path generation method starts with identifying the points within the isolated complex machining area from all the data points. The data points are first classified into two different groups based on their geometric complexity: the complex points (points with higher complexity) and the non-complex points (points with lower complexity). The geometric complexity considered here attempts to quantify the combined effect of two local geometric properties on tool path interval determination: normal vector orientation and curvature. A larger angle between the normal vector and the cutter axis and larger curvature will result in higher geometric complexity. This means that allowable tool path intervals for the machining area containing points of higher geometric complexity will be smaller than those for the remaining machining area.

The data points are further examined regarding their gouging with the given ball-end mill for the non-complex machining area, which is often larger and more efficient. The complex points and the gouge points are then grouped together to form the isolated complex machining area, which is distinctly separated from the rest of the machining area. The data points within the isolated complex area are analyzed to determine the corresponding gouge-free cutter size. Tool paths for the two distinct machining areas are generated separately to ensure high machining efficiency.

3.1 Data point classification by complexity

All the data points are first classified according to their local surface complexity. Figure 2 depicts the geometry of

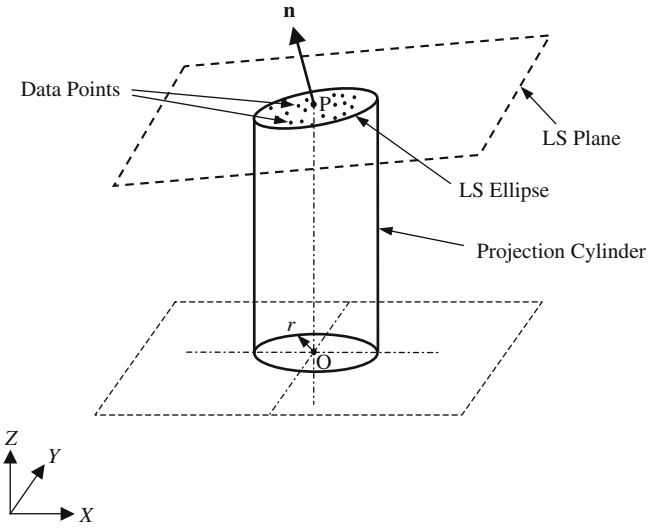

Fig. 2 Geometry of complexity definition and determination

defining and determining the complexity of a typical data point, for example, point P. Let O be the projection of P on the *XY* plane. An initial circle centered at O is formed such that three nearby points are within the corresponding projection cylinder. A least-squares plane is then fitted for P, which is formulated as:

$$Z = aX + bY + c. \tag{1}$$

The coefficients of the least-squares plane a, b and c in Eq. 1 can be easily solved from the following matrix equation:

$$\begin{bmatrix} \Sigma X_i^2 & \Sigma X_i Y_i & \Sigma X_i \\ \Sigma X_i Y_i & \Sigma Y_i^2 & \Sigma Y_i \\ \Sigma X_i & \Sigma Y_i & N \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \Sigma X_i Z_i \\ \Sigma Y_i Z_i \\ \Sigma Z_i \end{bmatrix}, \tag{2}$$

where (X_i, Y_i, Z_i) represents the Cartesian coordinates of the data points within the projection cylinder and N is the total number of these data points. Varying the radius of the initial circle and thus its projection cylinder, the root mean square (rms) values of the fitted residuals of the least-squares plane vary with changing N. The valid radius is obtained when the rms value of the fitted residuals is equivalent to the preset termination criterion. The value of the preset termination criterion is determined from the specified machining tolerance. It is taken in the present work as one third of the machining tolerance, based on the normal distribution assumption of the fitted residuals.

In essence, the fitted least-squares plane is used to approximate the local surface geometry of the data point. The normal vector **n** at P is estimated using the normal vector of the fitted plane. The intersection curve of the projection cylinder of radius r with the least-squares plane is an ellipse, which is named as the least-squares ellipse. The area S of the least-squares ellipse closely corresponds to the complexity of the local surface geometry. A larger S indicates flatter local geometry and a smaller S indicates changing geometric shape in the neighborhood of the data point. The complexity C of the data point P is thus defined as the reciprocal of S as:

$$C = \frac{1}{S} = \frac{1}{\pi r^2 \sqrt{a^2 + b^2 + 1}}. \tag{3}$$

The geometric shape of the point cloud data set can be interpreted and established using the derived complexity of each data point. The data points are to be classified into the complex and non-complex points. The critical complexity C_c, separating the complex data points representing the isolated complex machining area from the non-complex data points representing the rest of the machining area, is determined by an iterative numerical procedure using the dynamic complexity variation range Δ_{cv}:

$$\Delta_{cv} = C_{\max} - C_{\min}, \tag{4}$$

where $C_{\max}$ is the maximum complexity value and $C_{\min}$ the minimum. Δ_{cv} is divided by 10 to form 10 subsets of the data points containing the same complexity variation range. The ratio of the number of data points within each subset to the number of data points with higher complexity is then calculated. This ratio represents the potential percentage increase to the number of complex points if the corresponding data point subset is deemed geometrically complex. Since the critical complexity C_c is considered as the limiting complexity value for the set of complex points, it falls within a data point subset that will result in the minimum increase to the total number of complex points. Because of this, the data point subset with the minimum ratio is found and C_c is considered to be within the corresponding complexity variation range.

In order to identify the precise C_c value, the dynamic complexity variation range Δ_{cv} needs to be sufficiently small. As a result, Δ_{cv} is further reduced based on the maximum and minimum complexity values of the data point subset. The updated Δ_{cv} is again divided by 10 to form 10 sub-subsets of data points in the associated subset. This iterative procedure continues until the minimum ratio is reasonably small, which leads to the accurate determination of C_c. The data points with higher complexity than C_c are then regarded as the complex points whereas the rest of the data points as the non-complex points.

3.2 Gouging evaluation

A gouging evaluation algorithm is derived to identify the gouge points to a specific ball-end mill. This algorithm is used to examine the data points regarding their gouging with respect to the larger and more efficient ball-end mill for the non-complex machining area, and to determine the gouge-free cutter size for the isolated complex machining area. The gouge points are identified without being associated with a particular tool path in order to provide

the flexibility of tool path generation after the gouging evaluation process. For this purpose, every data point is regarded as a potential cutter contact (CC) point. Figure 3 depicts the gouging evaluation geometry for a typical data point P. As the data point P is taken as a CC point, the approximated unit normal vector **n** at P passes through the cutter location (CL) point P_{CL}. For a cutter with radius R, the coordinates of P_{CL} are simply:

$$\mathbf{P}_{\mathrm{CL}} = \mathbf{P} + R\,\mathbf{n} = \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} + R \begin{bmatrix} -\frac{a}{\sqrt{a^2+b^2+1}} \\ -\frac{b}{\sqrt{a^2+b^2+1}} \\ \frac{1}{\sqrt{a^2+b^2+1}} \end{bmatrix} \tag{5}$$

$$= \begin{bmatrix} X_p - \frac{Ra}{\sqrt{a^2+b^2+1}} \\ Y_p - \frac{Rb}{\sqrt{a^2+b^2+1}} \\ Z_p + \frac{R}{\sqrt{a^2+b^2+1}} \end{bmatrix}.$$

In order to ensure computational efficiency, only the data points within the cutter cylinder are considered in the gouging evaluation process (Fig. 3). The data points inside the projection cylinder of P are not evaluated. These data points have been effectively considered by the machining tolerance based termination criterion for fitting the least-squares plane. A point is deemed a gouge point if its distance to P_{CL} is less than the cutter radius R.

3.3 Machining area segmentation

The isolated complex machining area is separated from the rest of the part surface area for efficient tool path planning. After the complexity-based classification and the gouging evaluation of the data points, the complex points, points with higher complexity than the critical complexity, and the gouge points are grouped together to form the isolated

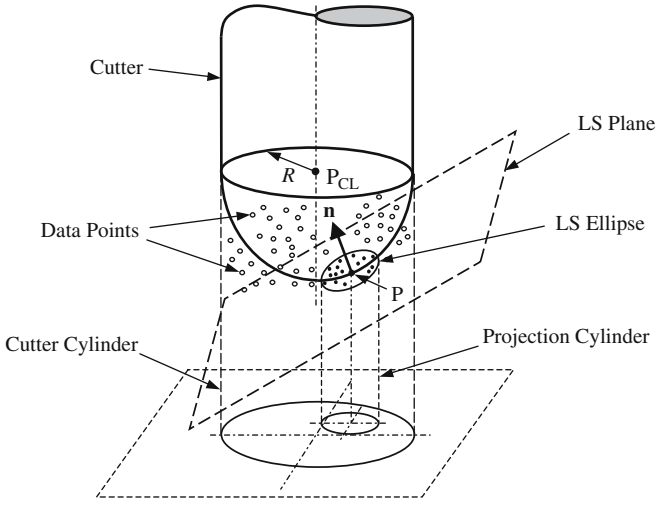

Fig. 3 Gouging evaluation geometry with respect to a specific ball-end mill

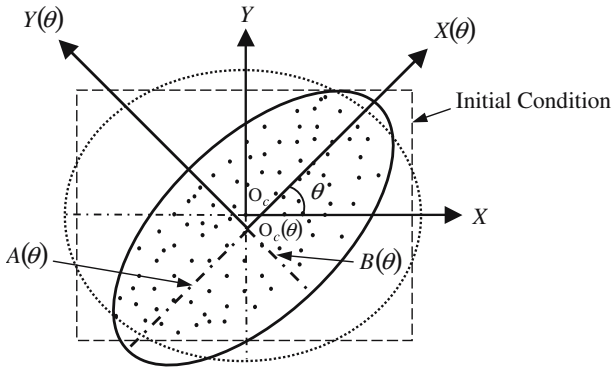

Fig. 4 Elliptical boundary optimization for the isolated complex machining area

complex machining area. An elliptical boundary with minimum XY projection area is employed to establish the isolated complex machining area.

Figure 4 shows the geometry of optimizing the elliptical boundary for the isolated complex machining area. The area of an elliptical boundary, S_c, can be expressed as:

$$S_c = \pi AB, \tag{6}$$

where A and B represent the major and minor radii of the ellipse, respectively. Minimizing S_c involves five optimization parameters: the X and Y coordinates of the ellipse center O_c, the orientation angle θ, and the major and minor radii. To simplify the optimization procedure, the orientation angle θ of the elliptical boundary is set to change from 0 to π incrementally with a small angle step. A 2D rotational coordinate system transformation is then performed according to the incremented orientation angle. For a specific orientation angle θ, the X and Y coordinates of O_c are approximated as the averages of the respective maximum and minimum of the X and Y coordinates of the data points. The major and minor radii of the elliptical boundary are then optimized by the Powell's method to minimize the elliptical area at the given θ. For efficient computational convergence, the numerical iteration starts by using the rectangle that encloses all the associated data points as the initial condition. The area of the minimized elliptical boundary is thus a function of the orientation angle θ:

$$S_c(\theta) = \pi A(\theta)B(\theta). \tag{7}$$

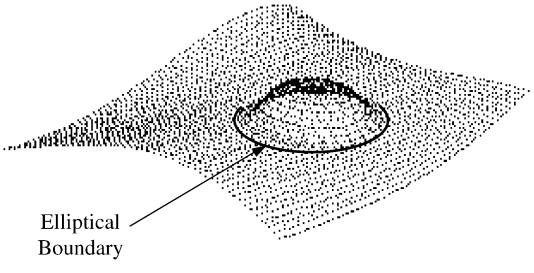

Fig. 5 Machining area segmentation by the elliptical boundary

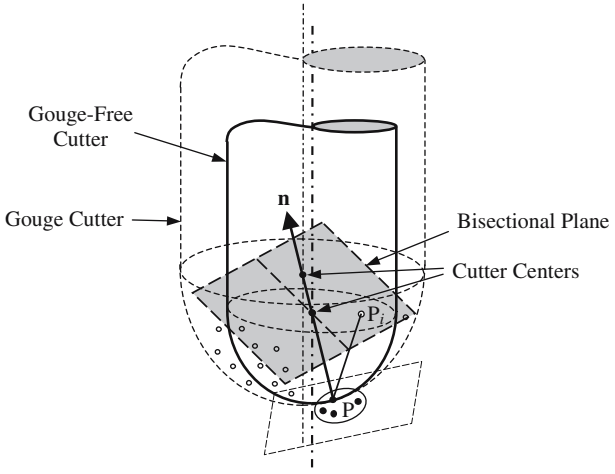

Fig. 6 Determination of the gouge-free cutter size at a typical data point P

The optimal elliptical boundary for the isolated complex machining area corresponds to the minimal $S_c(\theta)$. Figure 5 shows the elliptical boundary on the part surface that separates the two distinct machining areas. The elliptical boundary establishes the isolated complex machining area containing the complex and gouge points. Points outside the elliptical boundary constitute the flatter area of the free-form part surface.

3.4 Gouge-free cutter size determination

A gouge-free cutter needs to be used to avoid gouging in the machining area. Based on the same assumption made earlier for gouging evaluation that every point of the cloud data is a CC point, determination of the gouge-free cutter size at a typical data point P is illustrated in Fig. 6. For an identified gouge point P_i at P, the current cutter size needs to be reduced so that P_i will not be cut off by the cutter. The largest cutter size without gouging P_i can be determined by considering both P and P_i to be on the hemispherical surface of the ball-end mill. To locate the center of this cutter, P is first connected to P_i by a line segment. As P is the CC point, the center of the associated cutter is to be along the normal vector **n.** Intersecting the bisectional plane of $\overline{PP}_i$, the plane perpendicular to and crossing the middle of $\overline{PP}_i$, with **n** yields the center point of the gouge-free cutter for P_i. The corresponding cutter size is readily available from the distance of the cutter center to P. The same type of calculation repeats for the remaining gouge points at P until no more gouge points can be found.

To determine the applicable gouge-free cutter size for the isolated complex machining area, the minimum of the gouge-free cutter sizes at all the data points in the isolated area is calculated first. The applicable gouge-free cutter is then selected from the commercially available series of ball-end mills. The selected cutter size is less than and closest to the minimum gouge-free cutter size.

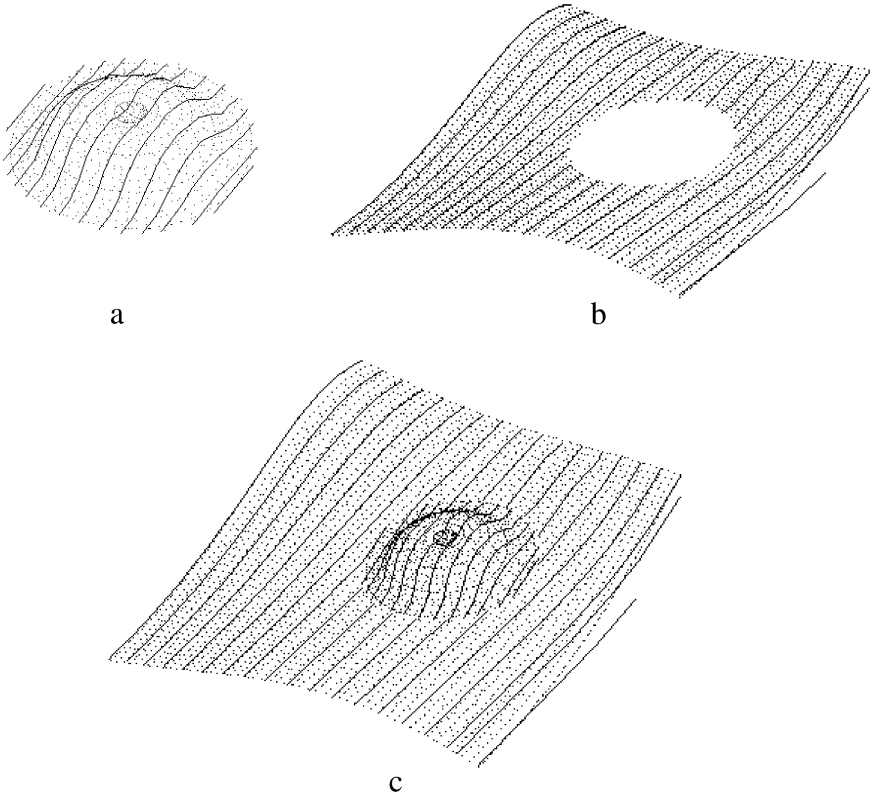

Fig. 7 Generated tool paths for: (**a**) isolated complex machining area; (**b**) non-complex machining area; and (**c**) entire part surface (combined complex and non-complex machining areas)

3.5 Tool path generation

Tool paths are generated separately from the data points within each distinct machining area using the method developed by Feng and Teng [19]. The advantage of this method is that the specified machining tolerance and scallop-height constraints are explicitly considered in generating the ball-end milling tool paths. For the isolated complex machining area, tool paths are generated using the corresponding gouge-free cutter. For the rest of the part surface, tool paths are generated using the larger and more efficient ball-end mill. The elliptical boundary, established for machining area segmentation, is also used to trim the generated tool path segments for both machining areas. The resulting efficient tool paths are the combination of the two sets of tool paths for the inside and outside areas of the elliptical boundary.

4 Implementation and discussion

The method presented in the previous section to generate efficient three-axis ball-end milling tool paths was implemented through the machining of a typical free-form surface containing an isolated complex machining area (Fig. 1). The point cloud data were sampled from the corresponding CAD model. A normally distributed random error component with a 0.0083 mm standard deviation was added to the sampled data points to simulate the measured point cloud data from a 3D laser scanner. A 12.7 mm ball-end mill was chosen to machine the free-form surface under the machining tolerance of 0.1 mm and the scallop-height constraint of 0.3 mm.

Figure 7 shows the generated tool paths for the two distinct machining areas. For the complex machining area within the elliptical boundary, the maximum gouge-free cutter size was calculated as 4.941 mm. From the standard commercial cutter series, a 3/16 in (4.763 mm) ball-end mill was selected to machine the isolated complex machining area. The given 1/2 in (12.7 mm) ball-end mill was used to machine the rest of the surface area outside the elliptical boundary.

Comparison of the generated ball-end milling tool paths with and without machining area segmentation is shown in Fig. 8. It can be clearly seen that high machining efficiency is achieved with machining area segmentation. This is due to the following two effects: (1) tool path redundancy is significantly reduced by separating the complex area from the part surface, and (2) the non-complex area is machined by the larger and more efficient cutter (the 1/2 in or 12.7 mm ball-end mill). Without machining area segmentation, the smaller and gouge-free cutter for the complex area (the 3/16 in or 4.763 mm ball-end mill) has to be used to machine the entire part surface. This significantly decreases the machining efficiency due to the evident increase of the overall tool path length. More importantly, the machining feed rate needs to be lowered considerably for the smaller ball-end mill in order to constrain cutter deflection and the resulting machining error [22]. For tool paths passing through the complex area, machining efficiency is further compromised by the smaller than required tool path intervals in the non-complex area. The small tool path intervals are in fact only needed in the complex area to meet the specified scallop-height requirement.

5 Conclusions

A new method of generating efficient ball-end milling tool paths directly from point cloud data for the three-axis machining of free-form surface geometry has been presented in this paper. Significantly improved machining efficiency is achieved for geometrically complex part surfaces having isolated complex machining areas. The uniqueness of the presented method is the segmentation of the part surface into distinct machining areas for which cutters of different sizes are used. The resulting high machining efficiency can be attributed to reduced tool path redundancy and increased feed rate for the selected larger cutting tool. The machining area segmentation is based on the estimated normal vector and complexity of each data point, which characterize the corresponding local surface

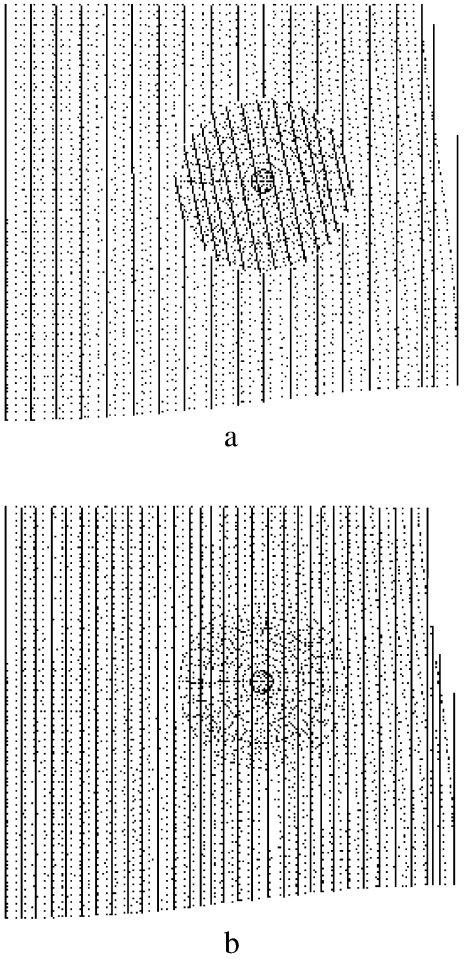

Fig. 8 Generated tool paths: (**a**) with machining area segmentation; and (**b**) without machining area segmentation

geometry. Density of the data points is thus an important concern to the valid estimation of these local geometric properties. The massive data points being collected by modern 3D contact or non-contact scanning equipment are dense and therefore, allow correct implementation of the presented method. Nonetheless, a complete representation of the part surface by the data points is essential. Incomplete data set, for example, due to occlusion in machine vision does not provide adequate information for the machining of the entire part surface.

For the machining area segmentation research at its current developmental stage in our laboratory, an elliptical boundary is employed to establish the isolated complex machining area. A 2D polygonal convex hull to enclose the same set of data points has also been considered. The convex hull boundary is versatile and can handle most area contours including the isolated area in the part surface. The challenge is the many sharp corners of the convex hull, which presents a potential difficulty to be resolved in applicable tool path generation.

Acknowledgement The financial support of this work from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

References

1. Loney GC, Ozsoy TM (1987) NC machining of free form surfaces. Comput Aided Des 19(2):85–90
2. Elber G, Cohen E (1994) Toolpath generation for freeform surface models. Comput Aided Des 26(6):490–496
3. Bobrow JE (1985) NC machine tool path generation from CSG part representations. Comput Aided Des 17(2):69–76
4. Huang Y, Oliver JH (1994) Non-constant parameter NC tool path generation on sculptured surfaces. Int J Adv Manuf Technol 9(5):281–290
5. Park SC, Choi BK (2000) Tool-path planning for direction-parallel area milling. Comput Aided Des 32(1):17–25
6. Park SC (2003) Tool-path generation for Z-constant contour machining. Comput Aided Des 35(1):27–36
7. Suresh K, Yang DCH (1994) Constant scallop-height machining of free-form surfaces. ASME J Eng Ind 116(2):253–259
8. Lin RS, Koren Y (1996) Efficient tool-path planning for machining free-form surfaces. ASME J Eng Ind 118(1):20–28
9. Sarma R, Dutta D (1997) The geometry and generation of NC tool paths. ASME J Mech Des 119:253–258
10. Feng HY, Li H (2002) Constant scallop-height tool path generation for three-axis sculptured surface machining. Comput Aided Des 34(9):647–654
11. Tournier C, Duc E (2002) A surface based approach for constant scallop height tool-path generation. Int J Adv Manuf Technol 19:318–324
12. Lee E (2003) Contour offset approach to spiral toolpath generation with constant scallop height. Comput Aided Des 35(6): 511–518
13. Han Z, Yang DCH (1999) Iso-phote based tool-path generation for machining free-form surfaces. ASME J Manuf Sci Eng 121 (4):656–664
14. Kim T, Sarma SE (2002) Toolpath generation along directions of maximum kinematic performance; a first cut at machine-optimal paths. Comput Aided Des 34(6):453–468
15. Lin AC, Liu HT (1998) Automatic generation of NC cutter path from massive data points. Comput Aided Des 30(1):77–90
16. Balasubramaniam M, Laxmiprasad P, Sarma S, Shaikh Z (2000) Generating 5-axis NC roughing paths directly from a tessellated representation. Comput Aided Des 32(4):261–277
17. Chuang CM, Chen CY, Yau HT (2002) A reverse engineering approach to generating interference-free tool paths in three-axis machining from scanned data of physical models. Int J Adv Manuf Technol 19(1):23–31
18. Park SC, Chung YC (2003) Tool-path generation from measured data. Comput Aided Des 35(5):467–475
19. Feng HY, Teng Z (2005) Iso-planar piecewise linear NC tool path generation from discrete measured data points. Comput Aided Des 37(1):55–64
20. Veeramani D, Gau YS (1998) Models for tool-path plan optimization in patch-by-patch machining. Int J Prod Res 36 (6):1633–1651
21. Ding S, Mannan MA, Poo AN, Yang DCH, Han Z (2003) Adaptive iso-planar tool path generation for machining of free-form surfaces. Comput Aided Des 35(2):141–153
22. Feng HY, Menq CH (1996) A flexible ball-end milling system model for cutting force and machining error prediction. ASME J Manuf Sci Eng 118(4):461–469