Int J Adv Manuf Technol (2008) 35:1090–1099
DOI 10.1007/s00170-006-0791-0

ORIGINAL ARTICLE

Real-time simulation and visualization of robotic belt grinding processes

Xiangyang Ren · Bernd Kuhlenkötter

Received: 17 May 2006 / Accepted: 4 September 2006 / Published online: 22 November 2006

Abstract Real time simulation and visualization are important for robot programmers to verify and optimize the path planning for the robotic belt grinding process. A new free-form surface representation based on discrete surfel element is developed to facilitate the system implementation, which exploits the advantage of the new development of point-based rendering technology in computer graphics. A local process model is integrated to calculate the material removal rate by considering the local geometry information and non-uniform force distribution. The final surface grinding error is easy to be assessed and visualized for quality evaluation. The experiments show that the simulation error is below 15%, even for a non-uniform contact under stable cutting conditions.

Keywords Simulation · Robotic belt grinding · Surfel-based modeling

1 Introduction

Grinding is a machining process which utilizes hard abrasive particles to fabricate desired parts from raw stocks. It is widely used to produce high precision surfaces and handle hard and brittle material. However, finish grinding is usually found to be costlier than other machining processes per unit volume of material removal. In addition, it is quite difficult to carry out the experiment grinding on inexpensive material because the resulting surface quality is strongly related to material type. Simulation method is a natural choice to check grinding paths for potential problems. It uses modern computer graphics technologies and the grinding process can be virtually displayed in the screen prior to actual production to improve the efficiency and accuracy.

1.1 Belt grinding processes

Belt grinding was developed in the 1960s for its high efficiency and low cost. Compared with traditional wheel grinding, belt grinding process has some unique characteristics. The machining cell usually has a tool motor, one or more contact wheels, a tension wheel and some other accessories. The abrasive belt covers the contact wheel and is driven by the tool motor with some pre-load generated by the tension wheel. During the machining, the workpiece is pushed against the contact wheel. The distributed abrasive grains, which are bounded by special mediums on a bedding layer of the belt, act as the cutting edges to remove materials. A typical situation of belt grinding is illustrated in Fig. 1 and it can be regarded as milling using a "cutter" with a large number of teeth of irregular shape, size and spacing [1]. With the introduction of industrial robot as manipulator, such a manufacturing cell is especially suitable to process parts with complex geometry like water-taps and turbine blades.

In the simulation of free-form surface grinding, finding out how to determine the material removal is one of the most important aspects. The whole simulation system is driven by incrementally subtracting material from the workpiece stock. Unlike turning or milling processes, it can not apply Boolean set operations between the tool envelop and workpiece. Instead, the calculation should

X. Ren (✉) · B. Kuhlenkötter
Robotics Research Institute, University of Dortmund,
Otto-Hahn-Str. 8,
44227 Dortmund, Germany
e-mail: xiangyang.ren@uni-dortmund.de

B. Kuhlenkötter
e-mail: bernd.kuhlenkoetter@udo.edu

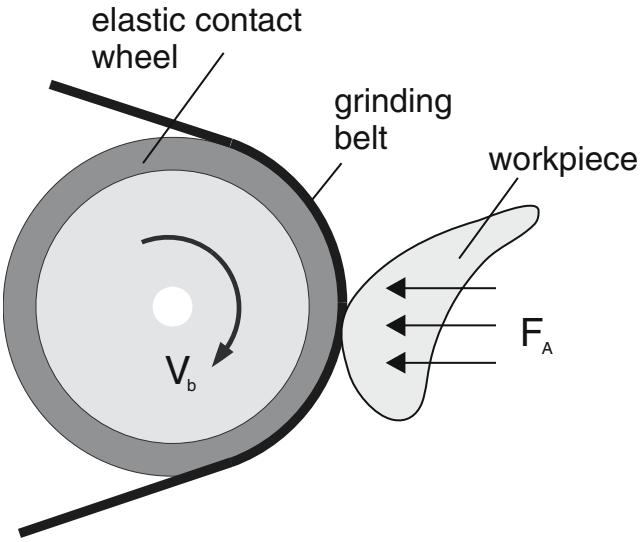

Fig. 1 Belt grinding process

based on an empirical model taking into account many influencing parameters [2]. Hammann once developed a linear global model especially for belt grinding processes [3]. But in free-form surface grinding, things are much more complicated. Particularly, the local non-uniform force distribution in the contact area must be considered and the influence of other manufacturing parameters also need to be studied.

1.2 Related works

Normally, NC cutting process simulation can be divided into two groups: dynamics and geometry simulation. Geometry simulation does not consider the influence of cutting force and other physical factors. It just simulates the relative movement between the tool and workpiece to verify the correctness of NC program. Dynamics simulation is one kind of physical simulation. It is often used to predict the tool broken and system vibration and to control cutting parameters through simulating system dynamic features. In our application, although all the cutting and environmental parameters should be taken into account when deciding the real material removal, it still belongs to geometry simulation because these parameters are only used to form the removal volume.

Direct solid modeling approaches, such as constructive solid geometry (CSG) and boundary representation (B_Rep), are first developed for manufacturing verification. They are capable of simulating the material removal process through a series of regularized Boolean difference operations to subtract successive tool swept volumes from the workpiece [4, 5]. Although these approaches can theoretically simulate and verify the process accurately, their application remains limited by the complexity of swept volume formulation and the time-consuming rendering process. Dexel [6–8] was developed for real-time shaded display of a model which is suitable for manufacturing simulation. The workpiece and tool geometries are represented by *dexels* and the geometry update is achieved by Boolean set operations on the one-dimensional dexels. Takafumi et al. [9] also used an extension of the z-buffer method (called G-buffer) to simulate NC machining. Jerard et al. [10] proposed a completely different approach. The designed surface is approximated by a set of points. The workpiece is represented by these points and the normal vectors associated with them. The vectors are shortened to the amount of over- or undercutting error when a tool moves over them. Ayasse [11] used a very coarse mesh as the support, on which a continuous vector field is standing and each vector length is decided by a discrete height field. The simulation procedure is implemented using V-projection and V-shooting. Glaeser et al. [12] applied differential geometric techniques to efficiently generate the swept volume of a moving cutter. Intersection calculations are carried out with a data structure called Γ-buffer. A comparison of these approaches can also be found in this paper.

1.3 Motivation

To implement the belt grinding simulation system, an appropriate workpiece representation method should be developed, which is suitable for the material removal calculation and process rendering while still maintain the advantage of the above simulation techniques. The goal is to provide realistic displaying and grinding error assessment so that robot programmers can verify and optimize pre-planned paths.

2 Surfel-based workpiece representation

Robotic belt grinding cell is usually used to process parts with free-form surfaces, which are given as a set of smooth parametric patches in B-Rep representation. So we take the designed surfaces as the basis and developed a representation technique using surface elements (Surfels) to facilitate the simulation and rendering.

2.1 Surfel definition

The term “surfel” is an abbreviation for surface element or surface voxel in the volume rendering and discrete topology literature. Herman [13] defines a surfel as oriented (d-1)-dimensional object in R^d. For d=3, this corresponds to an oriented unit square (voxel face) and is consistent with thinking of voxels as little cubes. Pfister et al. [14] modified

the definition as a zero-dimensional n-tuple with shape and shade attributes that locally approximate an object surface. In keeping with the convention of dexel or G-buffer, we adapt it to our application and redefine it as:

> **A surfel** is an oriented round disc, which locally approximates an object surface, extends in the normal direction and includes the information about shape, shade, displacement and neighborhood attributes.

Accordingly, the design of the surfel data structure should include the position and orientation, texture information, local differential geometry, neighborhood and some modification flags. Actually, it looks more like a dexel representation method in the parameterized space as shown in Fig. 2.

2.2 Uniform surface sampling

The first step to generate the surfel data structure of smooth surfaces is to approximate the surfaces by a set of sampled points. From the approximation view, they are piecewise constant surface approximants. Hence, the resulting approximation power is linear, i.e., with an average spacing h between the samples p_i, the approximation error with respect to each coordinate function is of the order $O(h)$. The resulting point number is square proportional to the required precision. Two aspects should be considered when deciding the sampling density. One is the simulation error of the manufacturing process. If we sample with a resolution of $n \approx h^{-1}$, we need $O(n^2)$ samples to cover a surface. So for a given simulation error, a over-sampled point set will greatly increase the computation, storage and display burden while under-sampled one does not meet the requirements. Another aspect is the visual artifact. The distribution of the approximate points should be dense enough for displaying.

Layered depth cube (LDC) and splat are the most popular methods for sample surfaces. LDC uses three orthogonal cubes to intersect the given surface. Ray casting records all intersections and generates samples at intersected points [14]. The surface normals are perturbed by bump and displacement mapping. Ellipse splat has first been proposed for rendering purposes by Zwicker et al. [15]. According to the differential geometry, it is the best linear approximation to a smooth surface. However, it is not suitable for material removal process simulation, which requires very dense sampling of workpiece surfaces to ensure the required simulation precision.

In our application, most free-form surfaces to be processed are represented as parametric surfaces, which means they can be regarded as two-dimensional u-v domains and the vertices of regular grids in these domains can be taken as the point set candidate. The more grids are generated by u and v isolines, the more density of the surface sampling we get. However, by purely using the (u, v) mesh vertices as samples, the isolines must be dense enough, especially in areas with high curvature in order to meet the simulation and rendering requirements. The resulting point set is not compact because the surface is heavily over-sampled in flat area as illustrated in Fig. 3. To reduce the redundancy, we developed an recursive surface sampling technology based on the spatial domain subdivision. Binary space partitioning tree (BSP-tree) is adopted due to the easy construction and fast leaf browse. The basic algorithm to decompose a smooth surface is shown as in Table 1. Here we introduced a shape factor *Thinness Ratio* to describe the patch shape. In case of a slim surface, even the area is much smaller than the threshold of the recursion terminate condition of patch area, it still has to be resampled to meet the requirements. The definition of thinness ratio is:

$$Thinness\ Ratio = 16\left(\frac{A}{p^2}\right) \tag{1}$$

where A is area and p is perimeter. It is usually used to describe the regularity of a shape. The idea shape for

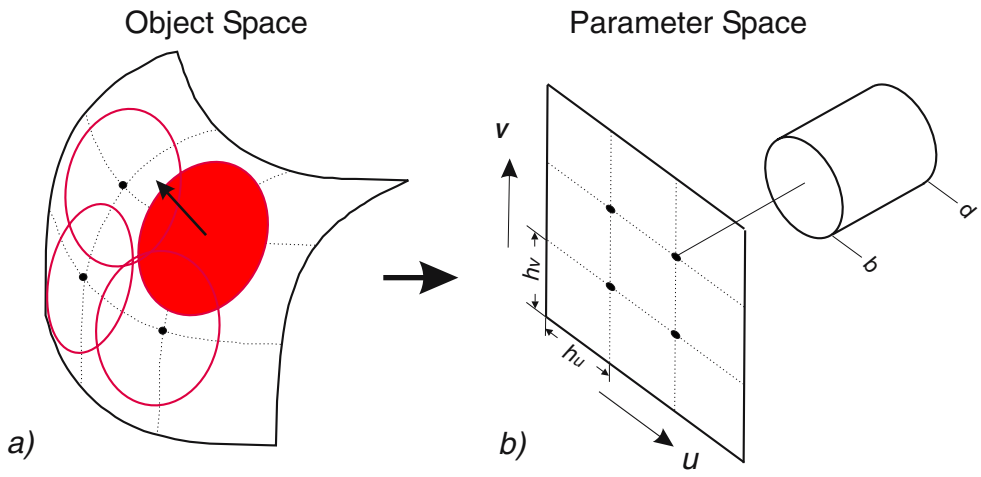

Fig. 2 Parameterized surfel coordinate system

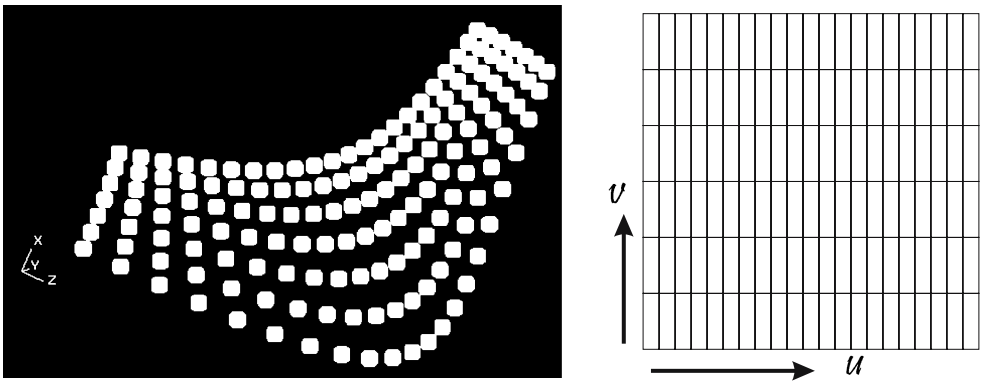

Fig. 3 Surface sampling based on *u-v* mesh

sampling is square. So the closer the thinness ratio is to 1, the more like a square the object is. We use a tolerance to indicate how far a sub-patch can deviate from a square and the final defined threshold for the shape factor is 1-*tolerance*. An example of uniform surface sampling is shown in Fig. 4, the irregular division in the *u-v* space causes a uniformly distributed samples in the surface.

2.3 Surfel elements generation

In surfel-based representation, the support basis is the approximate point set of the designed surface and the workpiece stock is represented by offsetting each point along the normal direction. So after the discretization of the designed surfaces, the next step is to compute the displacement. Since we have already got the position and orientation of each point, a corresponding ray is formed and used to calculate the intersection with stock surfaces. The procedure is quite similar to ray casting. Although it is a one-time process prior to simulation, the generation speed is still an important problem, especially for large and complex workpieces. Many optimization methods are available to accelerate the computation, such as regular cell-based algorithm and bounding box-based algorithms including oct-tree, kd-tree, BSP, etc.

2.4 Visualization

Triangle meshes are still the most common surface representation in many computer graphics applications because of their simplicity and flexibility. They can represent surfaces of any shape and topology and are accelerated in most of current hardware. But to approximate the smooth surface with a small error, one may generate a large number of triangles and in some cases, they may be smaller than a pixel when displaying. Therefore, the point-based rendering technology has been gaining popularity. The idea of using points as display primitives for continuous surfaces was introduced in 1985 by Levoy and Whitted [16], which inspired Pfister et al. [14] to develop the concept of surfels. More recently significant research has been performed on efficient high quality rendering of point-based geometry [15, 17, 18]. This makes it possible for us to directly display the material removal process using surfel points.

Table 1 Recursively decomposition steps

Decomposition steps
1. The surface is decomposed into sub-patches
2. If the child is bigger than the threshold, then subdivide
3. Else if the shape factor is over the limit, then subdivide
4. The process is repeated until all the surface is sampled

The surfel elements keep changing during the simulation, so one of the influence aspects about rendering is the normal vector estimation of each modified surfel. Due to the short displacement from the designed surface to the stock surface, normal perturbation techniques are appropriate methods. Bump mapping [19] has some drawbacks on the silhouette and shadow since the geometry remains unchanged. Displacement mapping overcomes this problem and is able to present surface details by defining an offset (displacement) from a base surface. Doggett et al. [20] define this in the mathematical framework as shown in Fig. 5. The base surface is defined by a vector function *P(u, v)* that defines 3D points *(x, y, z)* on the surface. Normals for the base surface are defined by $\widehat{N}(u,v)$ and the displacement scale fields by *D(u,v)*. Both of these are defined over the same domain as the base surface *P(u,v)*. Following these notations the points on the new displaced surface $P_n(u,v)$ are defined by

$$P_n(u,v) = P(u,v) + D(u,v) \cdot \widehat{N}(u,v) \tag{2}$$

where $\widehat{N}(u,v) = \frac{N(u,v)}{|N(u,v)|}$. We can get the designed surface normal *N(u,v)* in the uniform sampling process and calculate the height field *D(u, v)* when generating the surfel elements. According to Doggett [20], the normal vector of

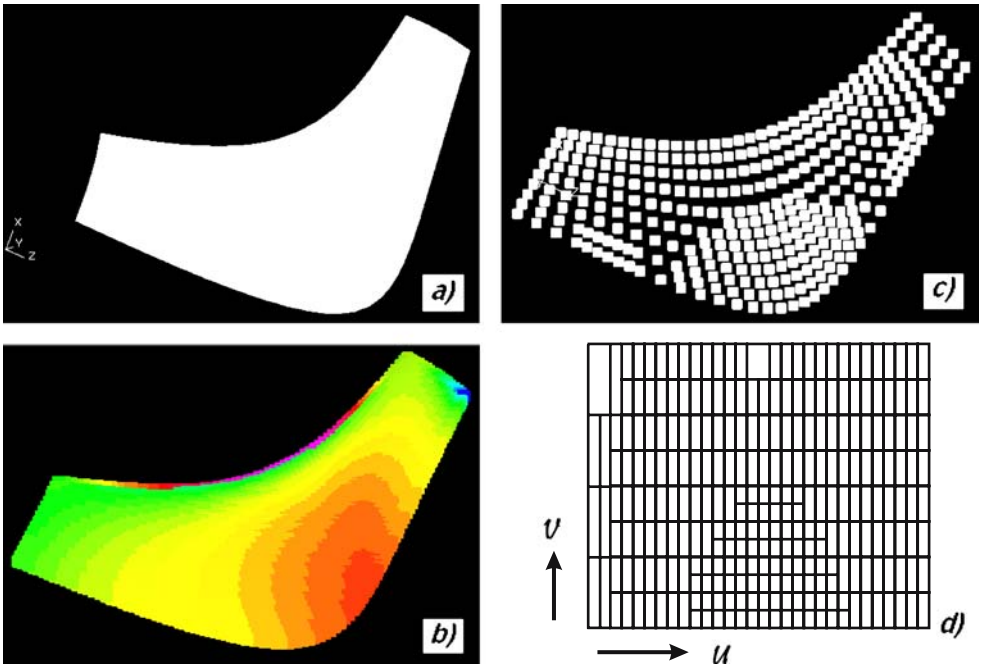

Fig. 4 Uniform surface sampling. **a**) a parametric surface **b**) surface mean curvature analysis **c**) uniform sampled points **d**) *u*-*v* mesh subdivision corresponding to uniform sampling

this new surface is derived by taking the cross product of its partial derivatives.

$$N_n = P'_{n^u} \times P'_{n^v} = N + D'_u \cdot \left(\widehat{N} \times P'_v\right) + D'_v \cdot \left(\widehat{N} \times P'_u\right) \quad (3)$$

Since the normal vectors of affected areas keep varying during the machining process, it is important to record the neighbor points in order to compute the perturbation dynamically.

3 Belt grinding process simulation

Two questions have to be solved during the implementation of the grinding process simulation. One is the representation of the workpiece and cutter model, and the other is the determination of material removal. The first one has been described in Sect. 2. The workpiece is discretized into surfel elements and the grinding tool is represented by polygons. The second one is to determine how much material is removed on each grinding point. The whole simulation system is driven by incrementally removing material from the workpiece stock. As mentioned above, it can not use Boolean set operations between the tool envelop and workpiece like turning or milling processes. Instead, the calculation should based on an experience model integrating many influential parameters.

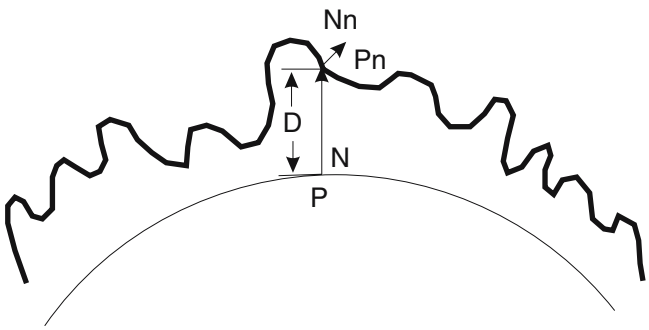

Fig. 5 Displacement mapping algorithm

3.1 Removal determination

In the free-form surface grinding process, the linear global grinding model given by Hamann [3] is not applicable anymore. Particularly, the local non-uniform force distribution in the contact area must be considered and the influence of other manufacturing parameters also needs to be investigated. Generally, the procedure to estimate the removal rate can be divided into three steps: contact situation determination, force distribution calculation and removal computation. The first one describes the geometric information about the intersection between the grinding belt and workpiece, which will be used to obtain the pressure in the contact area in the second phase. Then other parameters are included to get the final removal in the last stage. The whole process can be illustrated as in Fig. 6.

In the application the workpiece is represented by surfels and the stock is composed by those displaced points of the surfels. On each contact point of the pre-planned path, those surfels whose displaced points are located above the contact tangent plane are to be found. We select the tangent plane as the *X*-*Y* plane and project each affected surfel displaced point to this plane to get its *Z* value. According to

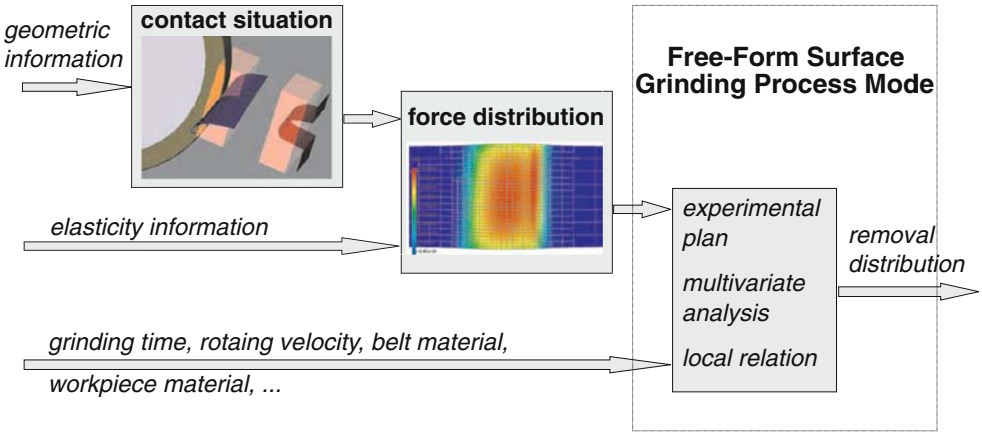

Fig. 6 Removal calculation procedure

the input requirement of FEM, these randomly distributed points are used to generate a regular grid mesh by extrapolating or interpolating Z values in those grid points where no data exist. Kriging [21] is one of the more flexible methods and is useful for gridding almost any type of data set. However, it can be rather slow. So the faster but less precise Shepard's method [22] is adopted to improve the simulation efficiency.

The acting force is not easy to measure because the sensors are difficult to install in order to obtain the fast-changing local force distribution during the high speed grinding. So approximate solutions are very popular in this area if the deformation is small enough. Then it can be regarded as a wholly elastic deformation and the classic theoretical model of strain-stress can be applied. Once the deformation information of the elastic contact wheel is known, the acting force in the contact area can be also computed according to the strain-stress relation. The calculation turns out to be a Signorini problem and the FEM is a self-evident technique in the first place to solve it [23, 24].

Finally, the influence of other parameters should be combined together to produce the results, which is denoted by the term of process model in grinding. The local model does not simply take a single quantity as the force or the removal but present the local situation of the entire contact area using a pattern of values. The local model, which is generalized by Cabaravdic [25], can be expressed as

$$r = C_g \cdot \frac{(V_b)^{e1}}{(V_w)^{e2}} \cdot F_A^{e3} \tag{4}$$

where

$$C_g = C_A \cdot K_A \cdot k_t$$

$$F_A = \begin{pmatrix} F_{11} & \dots & F_{1n} \\ F_{21} & \dots & F_{2n} \\ \vdots & \ddots & \vdots \\ F_{m1} & \dots & F_{mn} \end{pmatrix} r = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ r_{21} & \cdots & r_{2n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \dots & r_{mn} \end{pmatrix}$$

in which r is the removal distribution of chips; C_A is a revised constant of the grinding process; K_A is the combination constant of resistance factor of the workpiece and grinding ability factor of the belt; k_t is the grinding belt wear factor; V_b, V_w and F_A are the grinding rate, the feed-in rate and the normal acting force, respectively. e^1, e^2 and e^3 are the exponent coefficients while m and n are the mesh sizes of the discretized contact area in X-Y directions. Unlike the global model, the force $\mathbf{F_A}$ or the removal rate $\mathbf{r}$ is a matrix that represents the entire discrete information in the contact area. It contains more information and is applicable to the grinding of free-form surfaces. Illustrations of these steps on one grinding point are shown in Fig. 7.

3.2 Process simulation

The pre-defined grinding paths are composed of contact points. In each contact point, the contact situation is determined by a localization procedure in order to calculate the acting force. Then the real removal distribution is computed by the local process model as shown in Fig. 6. The material removal can be simulated by continuously modifying the surfels in affected areas. In this way, the generation of the removal volume is eliminated without sacrificing accuracy if the distance between two contact points is small enough. Each contact point has two attributes: position and orientation. The position is the Cartesian coordinate $P\{x,y,z\}$ in the robot base coordinate system and the orientation is the posture of the local framework on that point. Although different types of robots have different ways to represent the orientation, we choose quaternion $Q\{w,x,y,z\}$ as the standard method. Conversions may be required when applied to various robots. To ensure the approximation accuracy, the distance of neighboring contact points should be small enough. So for pre-planned paths, some extra points may needs to be added. The position of these points can be linearly interpolated. Concerning the orientation, it is a little more complicated.

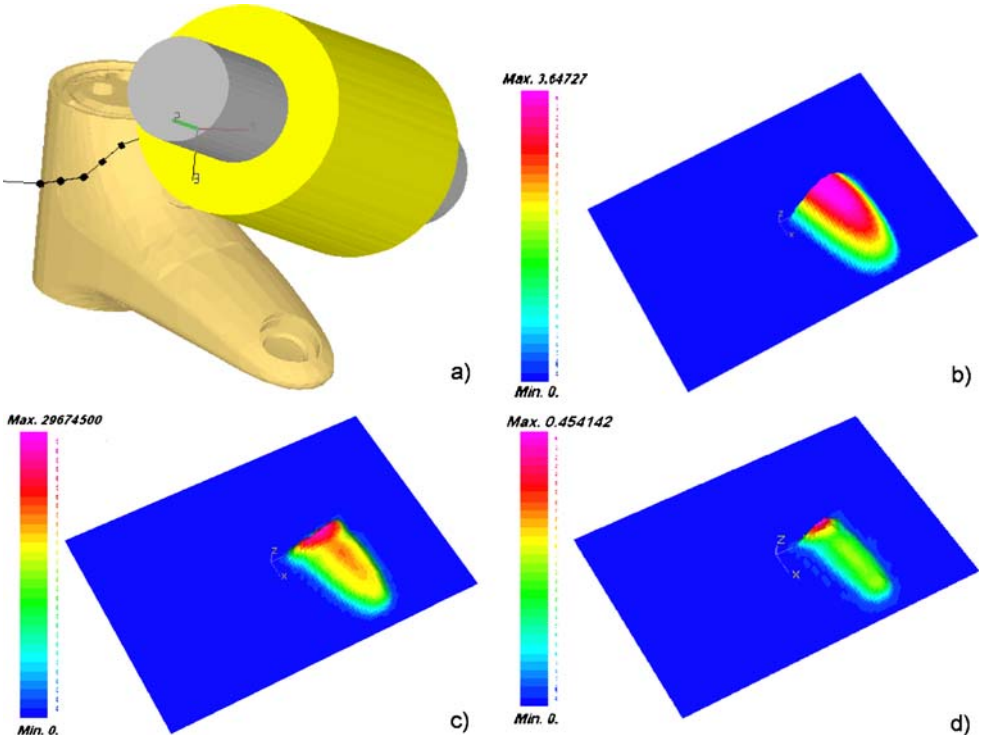

Fig. 7 Removal calculation of one contact point. **a**) The contact point in a grinding path. **b**) Contact situation of that point. **c**) Force distribution computed from the contact situation. **d**) The final removal distribution in that contact area

As we know, all unit quaternions are mapped into a hypersphere in 4D space. The problem with the linear interpolation (lerp) of quaternions is that it interpolates along the secant line between the two quaternions and not their spherical distance. As a result, the interpolated motion does not have smooth velocity: it may speed up too much in some sections. Spherical linear interpolation (slerp) removes this problem by interpolating along the arc lines instead of the secant lines. Although higher order interpolation may achieve C^2 continuity, slerp is able to get a

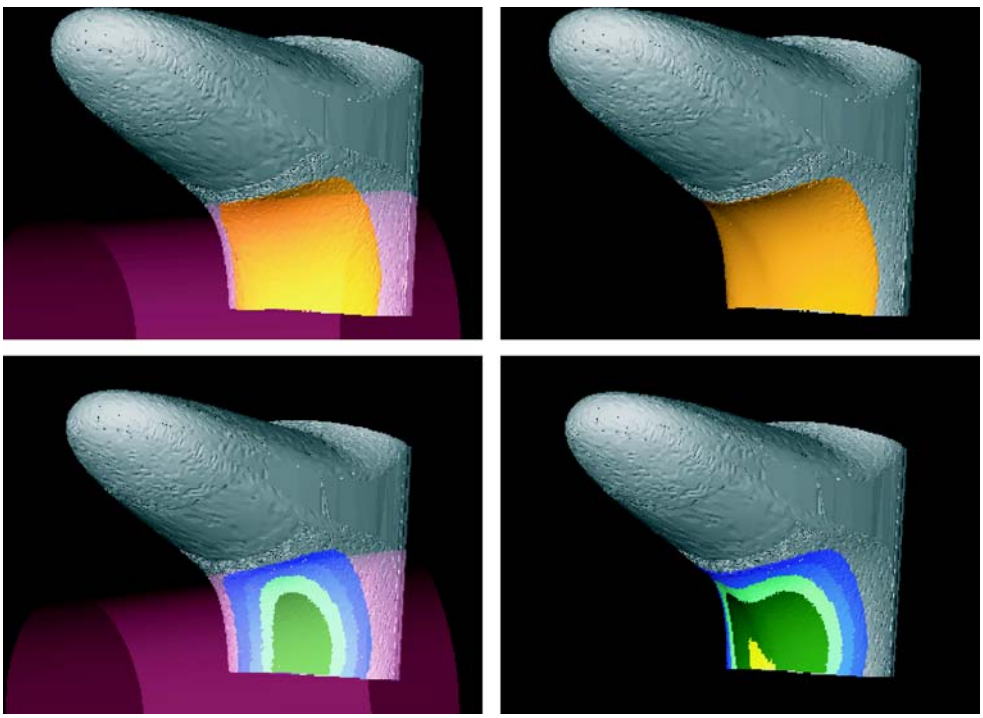

Fig. 8 Grinding process simulation and error assessment

desiring result with relatively low computation costs. For the ith point between Q_1 and Q_2, the interpolation result is:

$$\begin{aligned} \mu &= i/n \\ P_i &= (1-\mu)\cdot P_1 + \mu \cdot P_2 \\ Q_i &= Q_i \cdot \frac{\sin(1-\mu)\theta}{\sin\theta} + Q_2 \cdot \frac{\sin\mu\theta}{\sin\theta} \end{aligned} \tag{5}$$

where

$$\theta = \arccos(Q_1 \cdot Q_2) \tag{6}$$

One of the simulation goals is to help the robot programmer estimating the final part surface quality. So it is of great importance to assess the grinding error to see whether any problem such as gouge or undercut happens. Grinding error is usually described as the discrepancy between the resulting surface and the designed one. For a surfel element, it means the distance between surfel displaced point and the designed surface. Compared with other data structures, the surfel-based representation has a great advantage in analyzing the grinding error. The base points of surfels are the outputs of the uniform sampling and are exactly located on the designed surface. While the displaced points is merely moved a certain distance along the directions of the surface normal vectors. All we need to do is to compute the distance between the base point and displaced point for each surfel element and compare it with the error tolerance limit. If the final length is longer than the upper limit (assuming that the tolerance range is $[t_l, t_h]$), then undercut happens. Otherwise if it is beneath the lower limit, then it leads to gouge. The results of grinding error assessment can be visualized by several hues depicting the grinding depth. The hue index is built according to the error arrange and for a given grinding error of one surfel, a corresponding color is picked to represent it. There are two ways to implement the assessment: online assessment and post analysis. Online grinding error assessment is to calculate the difference and display it dynamically. This method is more vivid and instant for observers. In contrast, post analysis computes the error after all the grinding paths have been processed. It is obviously more efficient. An example of simulation and error assessment along one grinding path is demonstrated in Fig. 8. The error of the dark part in the ground area is within the tolerance range relative to the nominal designed surface while the outside brighter area represents undercut and the inside brighter one represents the amount of gouge.

Table 2 Process data of the experiment

Process parameters	
Workpiece: faucet	Material: brass
Infeed rate V_b: 30 m/s	Belt type: p100
C_g: 0.09	Grit size: 162 μm
e_1: 0.9	Belt tension: 0,5 N/mm^2
e_2: 1.23	E-module: 3 N/mm^2
e_3: 1.96	Robot type: IRB4400

3.3 Experimental evaluation

To evaluate the simulation system, we carried out a series of experiments using a faucet from GROHE Corporation, a sanitary system producer. The workpiece to be ground consists of 58 patched surfaces, and ranges 160 mm in x extension, 55 mm in y extension and 80 mm in z extension. It is made of brass and the grinding paths are planned by FAMOS, an off-line robot programming system. We sequentially pick some points in the surfaces and then decide their orientations which define the relative poses between the workpiece and contact wheel. Although this kind of trajectory planning heavily relies on programmers' experience, it is widely used in industry. These points form the grinding path and post-processor will generate the corresponding robot program based on the predefined coordinate frames. Extra detections of collision and singularity of the generated program have to be performed. The process data used in the experiment are listed in Table 2. C_g, e_1, e_2 and e_3 are obtained through statistical Design of Experiment (DoE). The others are chosen following the industrial guide and optimized in workshop by operators.

Since the investigation focuses on the approximation degree between the simulation and the real machining process, we pick 10 points out of the path shown in Fig. 7a) and test the positions in the normal direction before and after the grinding to get the final removal. For each point, the machining is an incremental procedure because the neighbor points also have some affection on it. So the final removal is a combined result of all the grindings around the point. The first column of Table 3 shows the material removal in each point after grinding. The beginning and

Table 3 Experimental evaluation of the simulation system

Test point	Removal of grinding (mm)	Removal of simulation (mm)	Approximation error (%)
1	0.019	0.082	331.58%
2	0.202	0.385	90.59%
3	0.322	0.503	56.21%
4	0.720	0.689	4.31%
5	0.894	0.938	4.92%
6	1.050	1.070	1.90%
7	1.397	1.418	1.50%
8	0.852	0.732	14.08%
9	0.619	0.721	16.48%
10	0.423	0.530	25.30%

ending points are located in the infeed and retracting area respectively, So the removal amounts are relatively small. In the middle of the path, with the increasing of the grinding depth, the material removal increases too. But in point 7, overcut can be obviously observed.

During the numerical simulation, the workpiece surfaces are discretized into 61 K points. The average cost is about 0.25 second for each contact point on a personal computer with AMD 2600+CPU and 1 GB memory. Users can adjust the simulation speed by changing the density of the approximation point set of the designed surfaces. The simulation results are displayed in the second column of Table 3. From the comparison between the results of simulation and real machining, we can see that in the normal grinding points such as point 4, 5, 6, 7 and 8, the simulation approximate the grinding process quite well and the error is under 15%. The overcut around point 7 is also accurately predicted. But in those infeeding and retracting areas with little infeed amount, the results are not satisfactory. There are many factors which may cause the problem. But the main reason is the process transition status. The parameters of the above model are achieved by experiments in stable working conditions, which means it can only approximate the removal during a regular grinding cycle. However, contacts during those infeeding and retracting phases contain too many indeterminate factors and lead to the inaccuracy or even invalidation of the process approximation. Further research is needed to improve the performance in such a situation. It is hard to investigate the detailed mechanism during these fast changing phases. A reasonable way is to study again the impact of these influencing factors to the final result and generate another model to more accurately approximate the processes under transition status. The switch of the different models can be trigged by controlling the threshold value.

4 Conclusion

We have presented a simulation framework of robot-controlled belt grinding cell. An enhanced geometric representation technique using the *surfel* element is developed, which facilitates the simulation implementation and improves the removal calculation efficiency. By integrating a local grinding model, it is able to approximate the removal distribution in a non-uniform contact area with the precision under 15% when the grinding conditions are stable. The average computation cost of the model for each contact situation is around 0.25 second. With this simulation system, the material removal process can be visualized in a interactive way and the planned grinding paths can be optimized or corrected based on the grinding error analysis.

Acknowledgements This project has been sponsored by the Deutsche Forschungsgemeinschaft (DFG) as a part of the research group 366 (Simulation-Aided Offline Process Design and Optimization in Manufacturing Sculptured Surfaces). Special thanks are due to professor Müller in Informatik VII for his constructive comments and advice.

References

1. Kurfess TR (2001) Computer-aided design, engineering, and manufacturing: systems techniques and applications, volume VI, manufacturing systems processes, chapter production of high quality parts by the process of grinding in manufacturing systems. CRC Press
2. Law SS, Wu SM (1973) Simualtion study of the grinding process. Trans ASME 972–978
3. Hammann G (1998) Modellierung des Abtragsverhaltens elastischer robotergefuehrter Schleifwerkzeuge. PhD thesis, University Stuttgart, Stuttgart
4. Kawashima Y, Itoh K, Ishida T, Nonaka S, Ejiri K (1991) A flexible quantitative method for nc machining veri-fication using a space-division based solid model. Vis Comput 7:149–157
5. Wang WP, Wang KK (1986) Geometric modeling for swept volume of moving solids. IEEE Comput Graph Appl 6(12):8–17
6. Van Hook T (1986) Real-time shaded nc milling display. In Proceedings of the 13th annual conference on Computer graphics and interactive techniques, pages 15–20. ACM Press
7. Huang Y, Oliver JH (1994) Nc milling error assessment and tool path correction. In Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pages 287–294. ACM Press
8. Mueller H, Surmann T, Stautner M, Albersmann F, Klaus W (2003) Online sculpting and visualization of multi-dexel volumes. In Proceedings of the eighth ACM symposium on solid modeling and applications, pages 258–261. ACM Press
9. Saito T, Tokiichiro T (1991) NC machining with G-buffer method. In Proceedings of the 18th annual conference on Computer graphics and interactive techniques, pages 207–216. ACM Press
10. Jerard RB, Hussaini SZ, Drysdale RL, Schaudt B (1989) Approximate methods for simulation and verification of numerically controlled machining programs. Vis Comput 5:329–348
11. Ayasse J (2003) Discrete displacement fields: a versatile representation of geometry for simulation in computer-aided manufacturing. PhD thesis, University Dortmund, Dortmund
12. Glaeser G, Gröller E (1998) Efficient volumegeneration during the simulation of NC-milling. In: Hege H-C, Polthier K (eds) Mathematical visualization. Springer, Berlin Heidelberg New York, pp 89–106
13. Herman GT (1992) Discrete multidimensional jordan surfaces. CVGIP: graph. Models Image Process 54(6):507–515
14. Pfister H, Zwicker M, van Baar J, Gross M (2000) Surfels: surface elements as rendering primitives. In: Akeley K (ed) Siggraph 2000, Computer Graphics Proceedings, pages 335–342. ACM Press/ACM SIGGRAPH/Addison Wesley Longman
15. Zwicker M, Pfister H, van Baar J, Gross M (2001) Surface splatting. In: Fiume E (ed) SIGGRAPH 2001, Computer Graphics Proceedings, pages 371–378. ACM Press/ACM SIGGRAPH
16. Levoy M, Whitted T (1985) The use of points as a display primitive. Technical report, Computer Science Department, University of North Carolina at Chapel Hill, January
17. Rusinkiewicz S, Levoy M (2000) QSplat: a multiresolution point rendering system for large meshes. In: Akeley K (ed) Siggraph 2000, Computer Graphics Proceedings, pages 343–352. ACM Press/ACM SIGGRAPH/Addison Wesley Longman

18. Pauly M, Keiser R, Kobbelt LP, Gross M (2003) Shape modeling with point-sampled geometry. ACM Trans Graph 22 (3):641–650
19. Blinn JF (1978) Simulation of wrinkled surfaces. In SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques, pages 286–292, New York, NY, USA, ACM Press
20. Doggett M, Hirche J (2000) Adaptive view dependent tessellation of displacement maps. In HWWS '00: Proceedings of the ACM SIGGRAPH/ EUROGRAPHICS workshop on Graphics hardware, pages 59–66, New York, NY, USA, ACM Press
21. Oliver MA, Webster R (1990) Kriging: a method of interpolation for geographical information system. Int J Geogr Inf Syst 4 (3):313–332
22. Shepard D (1968) A two-dimensional interpolation function for irregularly-spaced data. In Proceedings of the 1968 23rd ACM national conference, pages 517–524, New York, NY, USA, ACM Press
23. Blum H, Suttmeier F-T (2000) An adaptive finite element discretisation for a simplified Signorini problem. Calcolo 37 (2):65–77, June
24. Blum H, Schroeder A, Suttmeier F-T (2003) A posteriori error bounds for finite element schemes for a model friction problem. In Proc. of simulation-aided offline process design and optimization in manufacturing sculptured surfaces, Witten- Bommerholz, Germany
25. Cabaravdic M, Kuhlenkoetter B (2005) Bandschleifprozesse optimieren. mo Metalloberfläche 4:44–47