Visualization of Grinding Processes

Markus Fiege, Gerik Scheuermann, Michael Münchhofen, and Hans Hagen

Institute for Computer Graphics and CAGD
Department of Computer Science
University of Kaiserslautern, Postfach 3049, D-67653 Kaiserslautern, Germany
E-mail: {scheuer,mmuench,hagen}@informatik.uni-kl.de,
markus_fiege@yahoo.de
WWW home page: http://davinci.informatik.uni-kl.de/~m_fiege

Abstract. In grinding technology, the application of superabrasives and increasing demands for higher productivity and higher quality require an appropriate selection of optimum set-up parameters. An efficient way to determine and test these parameters is modeling and simulating the grinding process. A visualization of the results can support the choice of the parameters and increase the knowledge of the complex grinding process.

This paper describes a web-based visualization tool on the basis of a kinematic simulation. The tool allows the visualization of the surface of an already ground workpiece as well as the changing shape of the workpiece during the grinding process. Two methods for the visualization of the grinding-objects are implemented. One method describes the scene with the Virtual Reality Modeling Language, the other one uses a renderer to create the images.

Keywords: Internet, visualization, VRML, grinding

1 Introduction

The application of superabrasives and increasing demands for higher productivity and higher quality requires the optimum selection of set-up parameters for high performance grinding processes, in order to use the complete potential of this manufacturing process. One possible way of reducing the time consuming and costly grinding tests to determine the set-up parameters, is modeling and simulating the grinding process. The Institute of Manufacturing Engineering and Production Management of the University of Kaiserslautern has developed such a kinematic simulation [1],[7],[8].

A visualization of the simulation gives the possibility to view parts of the grinding process that are not or hardly ascertainable in reality. For example, the engagement of grinding wheel and workpiece in the contact area can be viewed. This is covered in the real process and therefore not visible. The knowledge of the grinding process can be increased by such detail-examinations. Thereby the set-up parameters and the choice of tools can be optimized in order to use the full potential of the grinding process.

The presented paper describes a web-based visualization tool [3], based on the data of a kinematic simulation. The visualization includes two different aspects of the grinding process. On one hand, an already ground workpiece can be visualized, called in the following *static grinding-object*. This gives the user the possibility to view the surface of the workpiece in detail. On the other hand, the workpiece can be shown during the grinding process (called *dynamic grinding-object*) so the user is able to view the changing shape of the workpiece.

Two methods for the visualization of the grinding-objects are implemented. The first one describes the scene with the Virtual Reality Modeling Language (VRML). This gives the facility of an interactive exploration of the workpiece in a VRML-viewer. The second method uses a renderer to create images. This has the advantage of a more realistic view of the scene and the possibility to use a higher level of detail.

2 Data formats and format-transformations

From the raw data, that means the output data of the simulation, to the final images, several data transformations have to be done. Figure 1 gives an overview about the structure of these transformations.

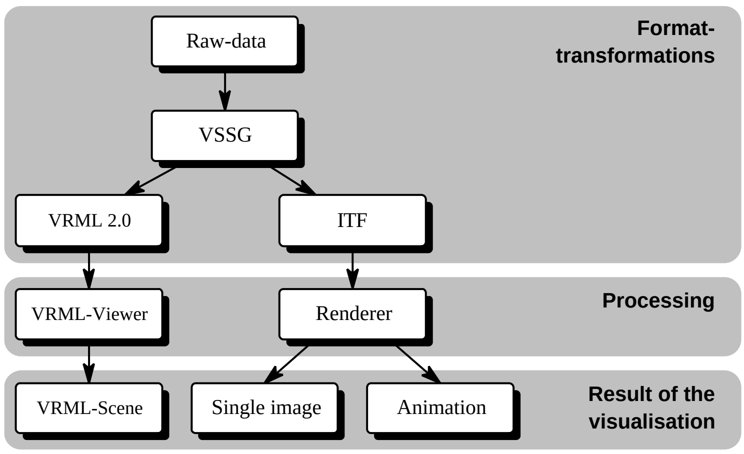

Fig. 1. The format transformations for the visualization

2.1 Raw data

The raw data is given from the simulation in an ASCII-file that describes the ground surface of the workpiece. The file consists of a header, describing the extensions of the workpiece, and a body, that defines the surface. It is defined by a height field over a rectangular grid (see Fig. 2). Therefore just one coordinate of each data point must be stored, the other ones could be calculated by the index and the distance of the grid.

For the grinding-objects the following coordinate-system is used. The x-axis is pointing in grinding direction. The y-axis defines the height of the workpiece, the depth is described by the z-axis. The data points of one line in grinding direction, that means with the same z-value, are defining a *grinding-track*.

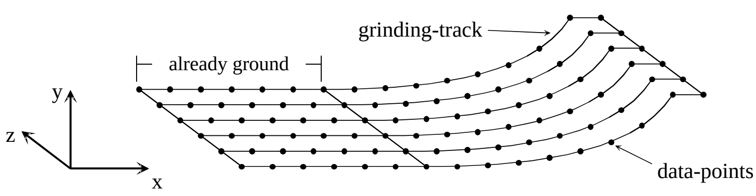

Fig. 2. Height field over a rectangular grid with coordinate system

To describe a static grinding-object all information is given by such a raw data file. The surface of a dynamic grinding-object changes with the time. For such objects, the shape of the workpiece is described at different times. Here the surface at any time is determined by a linear interpolation.

2.2 The Viper Symbolic Scene Graph (VSSG) / VIPER-System

The VSSG is a dynamic application independent data structure, developed in the VIPER-Project [2],[5]. All data is treated as *data objects* (so-called VSSG-Nodes), i.e. instances of generalized data types – *classes* – that may have attributes, which are either elementary or (reference to) other data objects. Classes are organized in an inheritance hierarchy. Type-specific behavior is added through modular actions (methods) which may be attached to data types individually for each application. The VSSG is chosen as the basic data structure for the visualization.

2.3 VRML and ITF

VRML [4] is a language to describe virtual environments, which recently has become the standard for interactive 3D-graphic in the Internet. One reason is the independence of hardware. With a Plug-In, for example the CosmoPlayer from Silicon Graphics, a common web-browser can handle VRML-scenes.

The Indexed Triangle Format (ITF) is a data structure to describe surfaces with triangles. The ITF is the input format for a renderer, developed at our institute.

2.4 The transformation out of the VSSG

The process of the transformation from VSSG into VRML or ITF is the same. The data processing is done by new actions, built in the VIPER-system. For every grinding-object the following steps are done:

1. Define the materials
2. Generate the geometry:
   - Reduce the data points
   - Define the data points
   - Create a triangulation of the data points
   - Define the faces by using the triangulation
3. Define the illumination
4. Define the viewpoints

A default material is assigned to all grinding-objects, for that a metal according to the usual material of a workpiece is chosen. To create an illumination corresponding to daylight, a distance light with a high part of ambient light is used to illuminate the scene. This should give the user an impression of the ground workpiece as realistic as possible.

For examining the workpiece, different viewpoints are defined. Some viewpoints are giving an overview of the grinding-object, others are focusing special details like the grinding edge. As special feature for the static grinding-objects an animated flight of the camera is implemented.

3 Data processing

3.1 Data reduction

A raw data set of a grinding-object is huge, it contains millions of data points. Especially the data files describing dynamic grinding-objects can reach a size of several hundred Mbytes. To reduce the amount of data points to a sufficient size, a data reduction is necessary. Two requirements are made:

- The data must only be reduced in grinding direction.
- The conversion into the VSSG should be made without any reduction.

The first consideration is made because the topological characteristic of the workpiece should be kept for every grit of the grinding-wheel. Each grinding-track represents an engagement of one or more grits. A reduction in direction of the z-axis would cause a reduction over the grits. In addition the height differences in grinding direction are very small in comparison with the changes across this direction – a reduction in both directions would not gain much.

The second requirement follows the idea, to create the VSSG as a 1:1 copy of the raw data. Only the data processing after the VSSG should contain a data reduction, which can be individually optimized according to the further application. In practice this proceeding isn't always optimal. Especially with the data sets of a dynamic grinding-object, a small reduction is sometimes useful.The following two algorithms for data reduction are implemented for the different steps in the data processing.

Data independent reduction. The data independent reduction is very easy and fast. Simply whole "rows" are removed out of the height grid, i.e. only every second (or third, fourth, ...) data point of a grinding-track is used. Despite its simplicity and missing consideration of the data, this proceeding turns out to be useful. The changes in height of the data points in grinding direction are very small and regular, so the result does not differ much from the result of a data dependent reduction. A reduction up to $\frac{1}{10}$ or $\frac{1}{20}$ of the original data shows no considerable difference between the data independent and data dependent algorithms. It is a further advantage of this method, that after the reduction all data points are still ordered above a regular rectangular grid.

For these reasons, the data independent reduction is used for the (possible necessary) reduction from the raw data into the VSSG. In comparison with the whole data reduction just a small reduction is required, a reduction up to the factor ten should be enough.

Data dependent reduction. The data dependent reduction is more costly than the method described above, because the values of the data points are taken in consideration. Instead it allows very strong reductions, because only the points with the least influence on the shape are removed. This algorithm for data reduction is used for the conversion from the VSSG into the ITF or VRML.

The data points of a grinding-track can be seen as an open polygon, whose vertices are represented by the data points. The idea of this method consists of creating a line from the beginning of the grinding-track as long as possible, without exceeding a maximum predefined error between the points and the line (see Fig. 3). From the final data point, a new line is restarted. Because in grinding direction the difference in height of the data points is very small and often some points are located on a line, this procedure removes many data points.

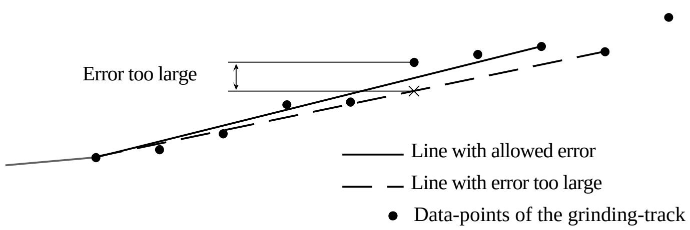

Fig. 3. The data dependent reduction with approximated calculation of the distance

The calculation of the distance between the data points and the line is the most costly part, because every point between beginning and end of the line must be checked. Therefore, two different algorithms are implemented where the user can choose the one he prefers according to his purpose.

Calculation of distances. The first algorithm approximates the exact distance by calculating the distance between the data point and the point on the line, that has the same x-coordinate. In addition, data point and line are on the same grinding-track, so they have the same z-coordinate. Therefore the distance is the difference between the y-coordinate of both points (see Fig. 3). So the algorithm just determines both y-coordinates, followed by calculating the difference – simple and fast. It overestimates the error, so no necessary points are removed.

The exact algorithm calculates the distance between data point and line with Hesse's Normal Form. Because of the exact distances the optimum reduction result is reached for a given error with this reduction algorithm. Despite the bigger costs for the calculation, for very strong reductions the exact calculation of the distance is often useful.

Reduction for dynamic grinding-objects. In the dynamic case each time-slice has a separate height field with the same grid structure – thus any data point has different heights for the different times. The reduction should keep this property. This means that all data points should exist in all time-slices, so only the height is allowed to change. A time-dependent grid would increase the computing time without improving the visual appearance.

The data independent reduction can be used for dynamic grinding-objects without any changes. However, the data dependent reduction must be used in a special way to keep the requirement. Each time-slice is checked parallel. For that, a line is built with the "common" procedure from the "same" data point of each time-slice. As soon as the first line reaches its full length, its final data point is kept in all time-slices as necessary. From this data point the algorithm is restarted.

3.2 Creating a triangulation

The surface of a workpiece in VRML as well as in the ITF format is composed of triangles. To extract the triangles from the data points, a triangulation is used. The triangulation is not created over the whole grinding-object. A VRML-viewer and the renderer are doing a shading over a face. Such a shading across the whole surface is not desired. In grinding direction a shading is desirable, because the edges between the triangles are smoothed according to the real workpiece. In direction of the z-axis a shading would smooth the engagements of the single grits. Certainly it is decisive to view the structure of the single grinding-tracks. Therefore the surface is divided into several stripes, with the area between two neighboring grinding-tracks building a stripe. In this way, the shading is done only in grinding direction. Now for each stripe a triangulation is created.

4 Conclusion and Future Work

Our tool allows the visualization of an already ground workpiece as well as the examination of a workpiece during the grinding process. The two implemented ways of visualization are by VRML (see Fig. 4) or by images, created by a renderer (see Fig. 5). The used data formats and the data transformations are explained. To carry out the transformations, two algorithms for data reduction are presented. For the data dependent reduction, two implemented strategies for calculating the distance are described.

One target during the development of the visualization tool was to use the World-Wide-Web for the visualization. So the programs are created with the ability to create a web-based client/server architecture.

In the future, some additional features can be integrated in the visualization system. Besides the geometry, a number of cutting parameters can be determined as the grinding forces, the grinding power or the heat flux in the contact area. To increase the knowledge of the grinding process, these quantities can be shown by special visualization techniques. A further improvement could be the creation of a material database.

Acknowledgments

The simulation data is courtesy of the Institute of Manufacturing Engineering and Production Management. We like to thank for providing the data. Especially, we want to thank Udo Zitt, Oliver Braun and Günther Warnecke for helpful discussions, their time and inspiration.

References

1. Braun, O.: Analyse und Modellierung der Abtrags- und Verschleißmechanismen beim Hochleistungsschleifen. Master-Thesis, Institute of Manufacturing Engineering and Production Management, University of Kaiserslautern (1998)
2. Disch, A.; Jacob, Ph.; Münchhofen, M.: VIPER – A 3D-Modeling- & Visualization-Toolkit for web-based Applications. WWW7 Conference, Brisbane, Australia (1998)
3. Fiege, M.: Visualisierung von Schleifprozessen. Master-Thesis, Institute for Computer Graphics and CAGD, University of Kaiserslautern (1998)
4. Kloss, J. H.; Rockwell, R.; Szabó, K.; Duchrow, M.: VRML 97: Der internationale Standard für interaktive 3-D Welten im World-Wide-Web. Screen-Multimedia Edition, Addison-Wesley (1998)
5. Seck, A.: VSSG nach VRML 2.0 Konvertierung. Projektarbeit, Institute for Computer Graphics and CAGD, University of Kaiserslautern (1997)
6. The Virtual Reality Modeling Language http://vrml.sgi.com/basics/index.html
7. Warnecke, G.; Zitt, U.: Kinematic Simulation for Analyzing and Predicting High-Performance Grinding Processes. Annals of the CIRP, volume 47/1/1998 (1998) 265–270
8. Zitt, U.: Modellierung der Schleifscheibentopographie von CBN-Schleifscheiben zur Simulation und Analyse des Flachschleifprozesses im Hinblick auf prozeßstabilisierende Korrektiven. Report to the research project Wa 501/19-1, Institute of Manufacturing Engineering and Production Management, University of Kaiserslautern (1996)

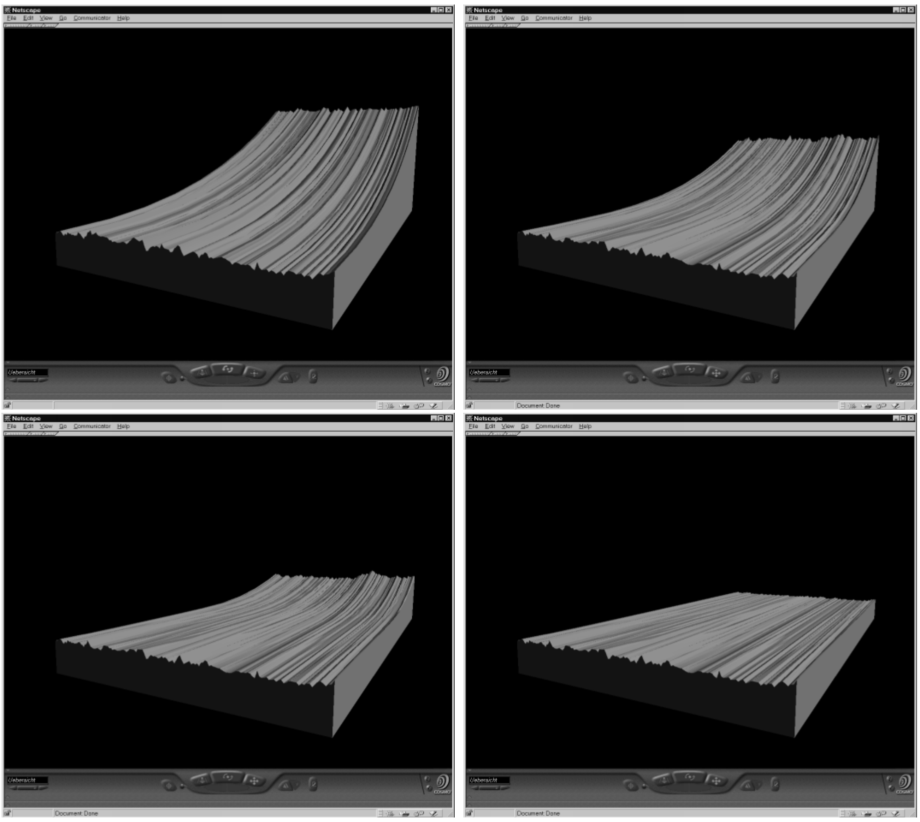

Fig. 4. A VRML-scene with a dynamic grinding-object

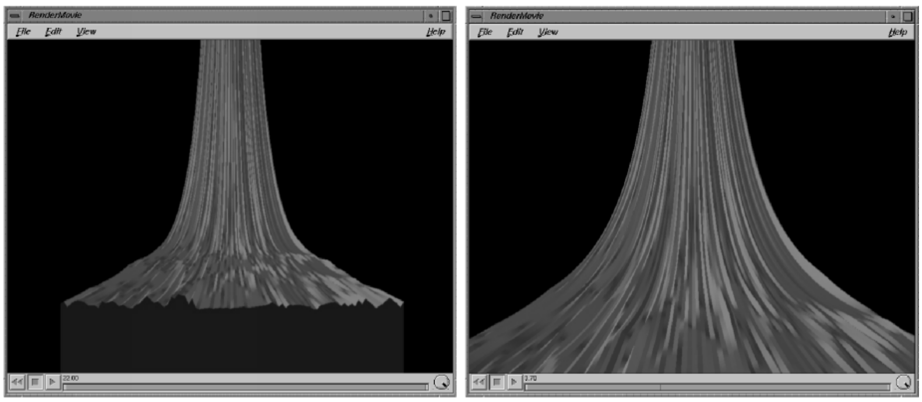

Fig. 5. A scene with a static grinding-object calculated by the renderer