01 Jul 2008

A New Contour Reconstruction Approach from Dexel Data in Virtual Sculpting

Kemal Yuksek

Weihan Zhang

Boryslaw Iwo Ridzalski

Ming-Chuan Leu
Missouri University of Science and Technology, mleu@mst.edu

Recommended Citation

K. Yuksek et al., "A New Contour Reconstruction Approach from Dexel Data in Virtual Sculpting," *Proceedings of the 3rd International Conference on Geometric Modeling and Imaging*, Institute of Electrical and Electronics Engineers (IEEE), Jul 2008.
The definitive version is available at https://doi.org/10.1109/GMAI.2008.27

A New Contour Reconstruction Approach from Dexel Data in Virtual Sculpting

*Kemal Yuksek, Weihan Zhang, Boryslaw Iwo Ridzalski, and Ming C Leu

* Department of Computer Engineering
Istanbul Kultur University
Atakoy Campus, Bakirkoy 34156 Istanbul/Turkey

Department of Mechanical and Aerospace Engineering
Missouri University of Science and Technology
Rolla, Missouri 65409, USA

{k.yuksek@iku.edu.tr, wzxq6, birtzd, mleu@mst.edu}

Abstract

This paper presents a novel method of contour reconstruction from dexel data solving the shape anomalies for the complex geometry in virtual sculpting. Grouping and traversing processes are developed to find connectivity between dexels along every two adjacent rays. After traveling through all the rays on one slice, sub-boundaries are connected into full boundaries which are desired contours. The complexity of the new method has been investigated and determined as O(n). We also demonstrate the ability of the described method for viewing a sculpted model from different directions.

1. Introduction

Virtual sculpting is a process in which the user creates a 3D model on a computer screen by interactively carving a workpiece like a real sculptor would do on a piece of clay, wax or wood. It is well suited for the design of parts with freeform geometry, especially at the conceptual design stage [1]. In a conceptual design, the exact dimensions of the design part are not determined initially, and the designer is more interested in creating part shapes and features. Commercial CAD systems such as Unigraphics, Ideas, Catia, PRO/E, etc. are powerful geometric modeling tools, but they require precise data for designing objects and thus do not allow the users to implement their ideas on shape and feature design in an intuitive manner.

We have developed an experimental virtual sculpting system [2,3]. The schematic of the virtual sculpting system configuration is shown in Fig. 1. The goal of this experimental system is to provide the designer with an intuitive virtual environment including stereo viewing and haptic interface capabilities such that s/he can focus on the design intent. Interactive modeling is implemented with VR hardware and software to allow the user creating and modifying 3D freeform objects.

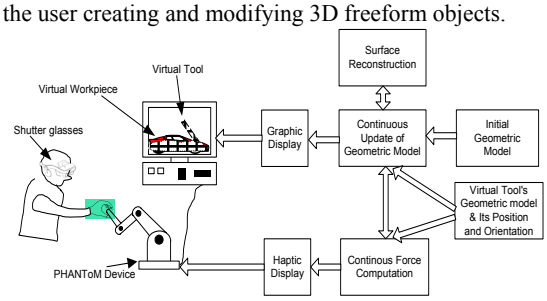

Figure 1 Schematic of the virtual sculpting system

The geometry modeling diagram is shown in Fig. 2. Both the tool and the stock (initial workpiece) are represented by polyhedral boundary representation, where the object surface is a faceted approximation composed of connected, non-overlapping triangles. The tool location is specified by a translation and a rotation tracked by the PHANToM^{TM}. The tool swept volume between two consecutive sampling times is represented by boundary triangular meshes. The workpiece and tool swept volumes are scan-converted to obtain their dexel representations. Boolean operations on dexels are obtained by comparing and merging the z-ranges of the involved dexels. In the process of the sculpting, the surface reconstruction module can be executed to convert the dexel model to a triangular mesh model for viewing the designed model from different angles and also export the model to other CAD/CAM packages.

978-0-7695-3270-7/08 $25.00 © 2008 IEEE
DOI 10.1109/GMAI.2008.27

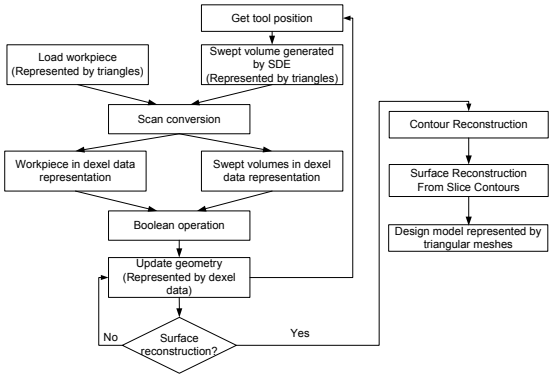

Figure 2 Geometry modeling diagram of the virtual sculpting system

The dexel representation of a solid (Fig. 3 (b)) is constructed via computing ray intersections with the solid slice by slice (Fig. 3 (a)). However, the view-dependence problem is the main limitation of the dexel based method due to the need of viewing simulation model in different view directions and exporting the model to other commercial software after the simulation. If parallel slices of contours can be reconstructed from dexel data, existing techniques [4] can be used to generate surface from contours. The difficulties of reconstructing contours from dexels lie on how to represent the relations between dexels among adjacent rays and how to connect dexels to form correct boundaries. We have developed a novel *contour reconstruction algorithm*. It consists two processes: **firstly**, a grouping criterion is developed to represent the relation between dexels and the gaps between dexels among adjacent rays. The gaps between dexels are verified to find out whether they belong to the inner contour which is the valid situation. Only valid gaps are kept in groups. **Secondly,** dexels and gaps in the same group are traversed and connected according to their overlapping relations to form contours with a contour clockwise sequence.

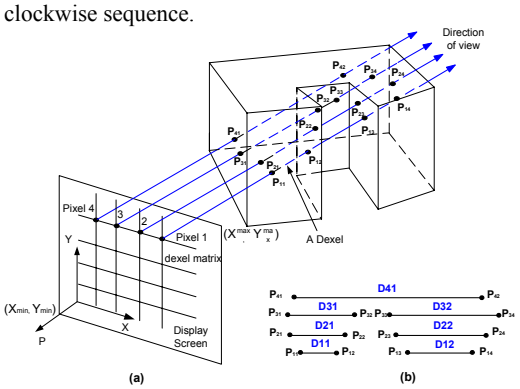

Figure 3 Ray casting process and the generated dexel model

2. Related Work

2.1. Dexel Representation

The dexel representation (also called the ray representation) of a solid is a set of line segments that lie inside the solid, obtained by classifying a grid of parallel lines with respect to the solid [5,6]. The line segments are called dexels and the intersection points are called dexel points. Dexels may also contain tags which are symbolic data associated with each line segment. Tags carry properties of the interior of a solid, or describe characteristics of a solid's boundary to a segment's endpoint. Dexels are generated by the RayCast Engine (RCE), which is a highly parallel computer algorithm that classifies grids of parallel lines against solids represented in triangular facets. The dexels that the RCE produces can be thought of as sampled boundary representations.

Because of its simple implementation procedure, low calculation cost, fast Boolean operation and proved completeness [6], dexel based methods are widely used in virtual sculpting, NC machining simulation, and other real-time simulation applications. Van Hook [7] developed a real-time shaded display of a solid model being milled by a cutting tool which follows an NC cutter path. Stifter [8] developed an NC simulation system using the dexel model. Konig and Groller [9] presented an extended ray reps approach in their NC simulation work which achieves real-time simulation and visualization for removal of inhomogeneous materials on low-end graphics hardware. Dexel representation can be easily extended to multi-dexel model [10] which used to perform conversion between CSG model and B-rep model. Muller et al. [11] presented the idea of multi-dexel volumes, which uses more than on dexel volume to represent a solid in their NC machining simulation.

2.2. Curve Reconstruction

Our work is similar to the curve reconstruction problem where the given data is a set of points on a smooth curve other than a set of dexels on a plane. The curve reconstruction problem has drawn a lot of attention from researchers over the last three decades [12,13] because of its many applications in computer vision, image processing and pattern recognition. If the curve is closed and uniformly sampled, a number of methods is known to work ranging over minimum spanning tree [14], α-shapes [15], β-skeleton [16], and γ-regular shapes[17]. A survey on these techniques appears in [18]. Recently, Dey et al., [19] presented an algorithm that comes with a guarantee for any set P of input points. The algorithm constructs a polygonal reconstruction G and a smooth curve T that justifies G as the reconstruction from P. Contour reconstruction from dexel data can be seen as a special case of curve reconstruction problem where relations between dexels are given as input. However, general curve reconstruction methods cannot be directly applied to

dexel data due to the nature of the input dexel data which does not sample the model uniformly.

Huang and Oliver [20] briefly described a contour tracking technique to reconstruct contours from dexel data without detailed development of an algorithm. The boundary of the object was visualized by simply displaying sets of contours extracted from the dexel data. Zhu and Lee [21] presented a visibility sphere marching algorithm for constructing polyhedral models from dexel models for their haptic virtual sculpting. When the algorithm was applied to some complex models, there could be some cracks and holes in the generated mesh due to topology related issues [22]. In this paper, we take advantage of the extra input information of the relations between dexels in the development of a novel contour reconstruction algorithm.

3. Contour Reconstruction Algorithm

3.1. Problem Statement

The problem of the contour reconstruction from dexel data is defined as: Given a set of dexels D sampled on a set of close contours C on the plane, the problem of connecting them according to their adjacencies in order to regenerate the same contour as C is called the contour reconstruction problem.

3.2. Premiers

Our contour reconstruction algorithm uses following definitions:

- **Dexel**: A dexel is a line segment which has two end points that are called dexel points. $D(x, y) : \{x \in [D_h, D_t], y = cons\}$ where (D_h, y) is the head of this dexel and (D_t, y) is the tail of this dexel.
- **Contour**: A set of vertices lining in a sequence of counter clockwise sequence. It has the same definition and data as the closed curve.
- **Real Dexel**: The dexels that are produced by the ray casting algorithm to represent the solid part of the object. For example, in Fig. 4, the line segments between point C and D, E and F, G and H are real dexels.
- **Temporary Dexel**: Temporary dexel is the gap or the distance between the two consecutive real dexels on the same ray. For example, in Fig. 4, the line segments between points D and E, points F and G are temporary dexels.
- **Valid temporary Dexel**: if two points of a temporary dexel belong to an inner contour, then this temporary dexel is valid such as the temporary dexel between points F and G in Fig. 4.
- **Invalid temporary Dexel:** if any of the two points of a temporary dexel belongs to an outer contour, then the temporary dexel is invalid such as the temporary dexel between points D and E in Fig. 4.
- **Overlapping**: If there is a common part in z direction between two dexels on the two adjacent rays, we say that they overlap with each other.
- **Connectivity**: Connection is made between adjacent dexel points. Dexel points p and q are connected we mean there is a path we can go from p to q or q to p.

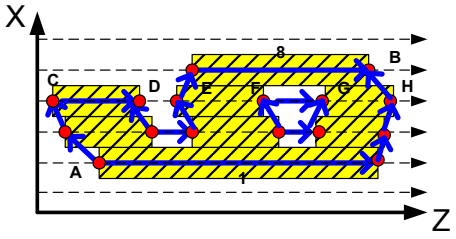

Figure 4 Real and temporary dexels

3.3. Grouping Process

The difficulties of the contour reconstruction problem arise from the case including multiple outer and inner contours. If we can separate vertices which belong to the same contour into the same group, a recursive algorithm can be developed to solve the contour reconstruction problem.

We developed a grouping algorithm. The algorithm starts from the first dexel along z direction with the minimum x value (i.e., dexel 1 in Fig. 4), traveling in z direction then in x direction until reaching the last dexel (i.e., dexel 8 in Fig. 4). The algorithm compares all the dexels including real and temporary dexels between every two adjacent rays. Dexels are categorized into different groups according to the overlapping relations. The overlapping criterion is defined as: if two dexels partly intersect with each other in z direction, we call them "half-intersect" as shown in Fig. 5 (left). If two dexels A and B including real and temporary dexels fully overlap with each other in z direction, we call B is "fully-covered" by A as shown in Fig. 5 (right).

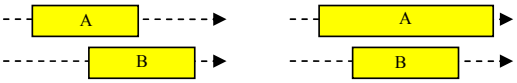

Figure 5 "half-intersect" (left) and "fully-intersect" (right)

There are two different cases: if one real dexel "half-overlaps" or "fully-overlaps" with another real dexel, then these two real dexels are both in the same group. The same rule applies to the half-overlaps between temporary dexels. But if a temporary dexel is "fully-overlaps" by a temporary dexel, then the temporary dexel is invalid. By using this output information, we can separate dexels into different groups and delete the invalid dexels. The temporary dexel is valid only when all the temporary dexels in the same group are valid.

3.4. Traveling Process

After the grouping process, the dexels with the same group index are put together. To connect dexels in the same group to have the contour boundary, we developed a traveling process to store the points on each contour in a counter clockwise sequence. For each group of dexels, the algorithm first selects the starting dexel which is the dexel that has the smallest x coordinate (in other words, the dexel which has a left most end point among other dexels). For exmaple, the dexel 1 is the starting dexel in Fig. 4. Then, the algorithm selects a moving direction (in this case, anti-clocokwise direction has been selected. This means that while you are moving you must increase the x value. Related to this direction determine the overlapping dexels. Next point will be the one which is the end point of this overlapping dexel. If there is no such a dexel, the algorithm will go to the other end of the same dexel and continue this way until reaching the starting point. In Fig. 4, two contours are formed by the contour reconstruction algorithm in the counterclockwise sequence.

4. Implementation and Analysis

4.1. Implementation

The code for the contour generation and surface reconstruction is written in C++. It runs on a Microsoft Windows XP workstation with a 2.8G Hz CPU, 512 MB RAM, and a GeForce4 MX 420 graphics card with 64MB memory. The graphic-rendering component is built on the OpenGL with the GLUT library.

The system diagram is shown in Fig. 6. The input is the triangular surface represented solid model, ray casting process discretizes the model into dexel data. Then, dexels on every two adjacent rays are grouped into different groups using grouping process. At the same time, temporary dexels are identified. Dexels which are in the same group are searched in the traveling process to extract the boundaries. Triangular surface model is finally reconstructed from slices of planar contours using the algorithm developed by Meyers and Skinner [4].

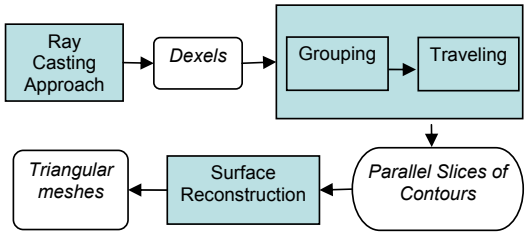

Figure 6 Contour generation diagram

To demonstrate the contour reconstruction process, solid objects as seen in Fig. 7(a) are discretized into the dexel data as shown in (b). The reconstructed contour data are shown in (c). Some reconstructed triangular mesh models from planar slices are shown in Fig. 8.

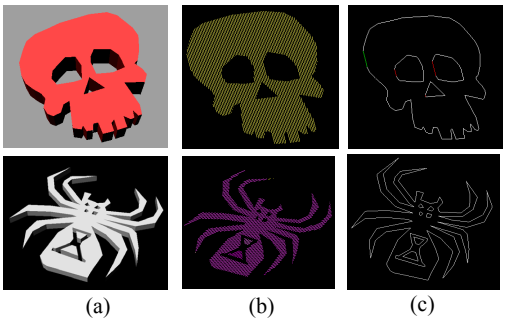

Figure 7 (a) Original models, (b) dexel models, and (c) contour models

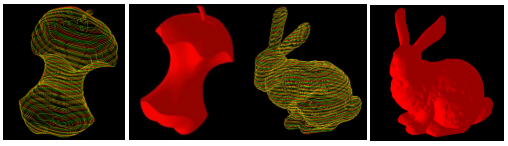

Figure 8 Reconstructed 3D mesh models from planar contours

4.2. Complexity Analysis

The curve reconstruction algorithm consists of two processes. The grouping process is very similar to the classical connected component labeling algorithm. By slightly modifying a connected component labeling algorithm, one can perform the grouping process. Chang *et. al* proposed a linear time algorithm for the connected component labeling problem [23]. Therefore, the grouping process can be implemented in a linear-time complexity using such a labeling approach. In the traveling process, finding the starting dexel takes a linear-time (O(n), where n is the number of dexels). After determining the starting dexel, we visit each dexel only once and we make constant amount of comparisons for each dexel as we move. Therefore, traveling process also has a linear time complexity (O(n)). As a result, the overall process of the curve reconstruction algorithm has a linear time complexity.

4.3. Precision analysis

Precision is an important issue for the dexel based method because model information is lost between rays. This loss of information may cause failures of the curve reconstruction algorithm. For example, as seen in Fig. 9 where (b) and (c) are the zoom-in pictures of (a), dexel D_1 and D_2 in (b) should be connected because they belong to the same contour. However, they are not in the same group because the contour information between

them is lost such that they are not overlapped with each other. In our current implementation, they are individually connected as lines. The same problem happens to the dexel D_3 and D_4 in (c). The solution to this problem is to increase the number of rays until such problem disappears.

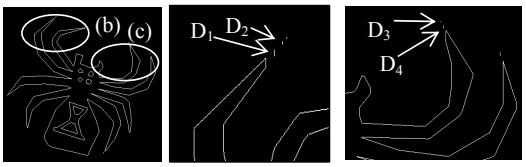

Figure 9 Examples of the precision problem

Conclusion

We have developed a novel method to extract 2D contour profile from dexel data to solve the view-dependent problem of dexel based applications: surface reconstruction and virtual sculpting. The dexel data are firstly separated into different groups and then dexels in the same group are connected according to their connectivity to get contours. A new approach have been suggested on the definition of the temporary dexel concept. Advantages of this technique have been explored by comparing to the traditional dexel representation for the contouring process. Implementation results and computational complexity analysis show the capability and effectiveness of the developed method.

Acknowledgements

This research is partly supported by the Intelligent Systems Center at the Missouri University of Science and Technology.

References

[1] M.C. Leu, B.Y. Maiteh, D. Blackmore and L. Fu. Creation of freeform solid models in virtual reality. *Annals of the CIRP*, 50(1): 73-76, 2001.

[2] M.C. Leu and X. Peng. Virtual sculpting with haptic interface. In *NSF Design, Service and Manufacturing Grantees and Research Conference 2003*, the University of Alabama, Birmingham, Alabama, 2003.

[3] X. Peng and M.C. Leu. Interactive virtual sculpting with force feedback. In *Japan-USA Symposium on Flexible Automation 2004*, Denver, Colorado, July 19-21, 2004.

[4] D. Meyers and S. Skinner. Surfaces from contours. *ACM Transactions on Graphics*, 11(3):228-258, 1992.

[5] J.L. Ellis, G. Kedem, T. C. Lyerly, D. G. Thielman, R. J. Marisa, J. Menon and H. B. Voelcker. The ray casting engine and ray representatives. In *Symposium on Solid Modeling and Applications 1991*, pages 255-267, June 05-07, 1991.

[6] J. P. Menon, R.J. Marisa and J. Zagajac. More powerful solid modeling through ray representations. *IEEE Computer Graphics and Applications*, 14(3):22-35, 1994.

[7] T. Van Hook. Real-time shaded NC milling display. In *ACM SIGGRAPH86*, pages 15-20, August, 1986.

[8] S. Stifter. Simulation of NC machining based on the dexel model: a critical analysis. *The International Journal of Advanced Manufacturing Technology*, 10: 149-157, 1995.

[9] A. H. Konig and E. Groller. Real time simulation and visualization of NC milling processes for inhomogeneous materials on low-end graphics hardware. In *Computer Graphics International 1998*, page 338, 1998.

[10] M. O. Benouamer and D. Michelucci. Bridging the gap between CSG and Brep via a triple ray representation. In *The Fourth ACM Symposium on Solid Modeling and Applications*, Atlanta, Georgia, pages 68-79, 1997.

[11] H. Muller, T. Surmann, M. Stautner, F. Albersmann and K. Weinert. Online sculpting and visualization of multi-dexel volumes. In *The Eighth ACM Symposium on Solid Modeling and Applications*, Seattle, Washington, pages 258–261, 2003.

[12] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 20:68–86, 1971.

[13] R.C. Veltkamp. The r-neighborhood graph. *Comput. Geom. Theory Appl.*, 1:227-246, 1992.

[14] L.H. Figueiredo and J.M Gomes. Computational morphology of curves. *Visual Computer*, 11:105-112, 1994.

[15] F. Bernardini, and C.L. Bajaj, Sampling and reconstructing manifolds using α-shapes. In *9th Canadian Conf. Comput. Geom.*, pages 193-198, 1997.

[16] D. G. Kirkpatrick and J. D. Radke. A framework for computational morphology. In *Computational Geometry, Edited by G. Toussaint. North Holland*, pages 217-248, 1985.

[17] D. Attali. R-regular shape reconstruction from unorganized points. In *13th Ann. Symposium Comput. Geom.*, pages 248-253, 1997.

[18] H. Edelsbrunner. Shape reconstruction with Delaunay complex. In *Third Latin American Symposium on Theoretical Informatics*, pages 119-132, 1998.

[19] T. K. Dey, K. Mehlhorn and E. Ramos. Curve reconstruction: Connecting dots with good reason. *Comput. Geom. Theo. & Appli.*, 15:229-244, 2000.

[20] Y. Huang and J. H. Oliver. Integrated simulation, error assessment and tool path correction for five-axis NC milling. *Journal of Manufacturing Systems*, 14(5):331-344, 1995.

[21] W. Zhu and Y-S. Lee. A visibility sphere marching algorithm of constructing polyhedral models for haptics sculpting and product prototyping. *International Journal of Robotics and Computer Integrated Manufacturing*, 21(1):19-36, 2005.

[22] W. Zhu. Virtual sculpting and polyhedral machining planning system with haptic interface. *Ph.D. thesis*, http://www.lib.ncsu.edu/theses/available/ etd-08172003-194602/, 2003.

[23] F. Chang, C-J. Chen and C-J. Lu. A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206-220, 2004.