# Fast Distance Field Interpolation for Reconstruction of Surfaces from Contoursure

**Andreas G. Schilling, Reinhard Klein**

**ISSN 0946-3852**
**WSI-99-15**

Wilhelm-Schickard-Institut für Informatik
Graphisch-Interaktive Systeme
Auf der Morgenstelle 10/C9
D-72076 Tübingen
Tel.: +49 7071 29-75462
Fax: +49 7071 29-5466

email: {andreas|reinhard}@gris.uni-tuebingen.de
URL: http://www.gris.uni-tuebingen.de/

Nearly identical version submitted as short paper to Eurographics 99, authors swapped.
04/30/1999

# Fast Distance Field Interpolation for Reconstruction of Surfaces from Contours

Andreas Schilling and Reinhard Klein

## Abstract

One simple and robust way to get a reconstruction of surfaces from a given contour stack dealing well with branching and other problems which are generally difficult to solve is based on the well known MC-algorithm. To overcome the staircase artefacts produced by the MC-algorithm Jones et. al. [3] proposed to use a distance field interpolation between the slices and to run the MC-algorithm on this distance field. The main problem of this approach is the distance field computation as it is very time consuming especially if high resolution grids (e.g. $1024 \times 1024$ are used. Therefore, in the original algorithm the resolution of the chosen grid is much less than the resolution of the given contour sacrificing accuracy of the resulting surface. Especially in medical applications this is not accepted by the doctors.

In this paper we introduce a new method for the computation of the discrete distance field, which is a breaktrough in terms of speed and accuracy. This new method allows us to reconstruct surfaces from contour stacks with guaranteed accuracy in reasonable time. Several examples show the power of this approach.

## 1 Introduction and previous work

The approach considered in this paper is to generate in a first step an accurate voxel representation from the contours and then in a second step to use a Marching Cubes (MC) algorithm [5] to extract the resulting surface. The major advantage of this approach is that a unique reconstruction is achieved automatically without solving the complicated correspondence and branching problems.

A simple implementation of this approach has the problem of staircase artifacts. To deal with this problem a discrete distance field can be used [3]. A standard distance transform as used in image processing is not applicable in this context, since it works on pixelized contours. Here, the contours are given in higher resolution than the pixel grid and we want to measure the exact distances to these contours. Unfortunetely, the exact distance field computation is very time consuming. To overcome this problem, Jones et. al. [3] reduced the resolution of the underlying grid sacrificing accuracy. One reason, that this approach is not commonly used in medical applications may be, that doctors do not accept solutions without guaranteed accuracy.

## 2 The distance field interpolation

Since our approach is also based on the principle of the distance field interpolation we briefly review this method. One possibility to overcome the problem of missing data between the slices is to interpolate the so called *distance field* between the contours [4, 7, 2, 3, 1].

Let $\Omega$ be the 3D object and

$$\Omega_i = \{(x, y) | (x, y, z_i) \in \Omega\}$$

a finite set of cross sections. Then the distance fields at the levels $z_0, \ldots, z_n$ is defined by

$$D_i(x, y) = \begin{cases} -dist((x, y)\partial\Omega_i) & \text{if} (x, y) \in \Omega_i \\ dist((x, y)\partial\Omega_i) & \text{otherwise} \end{cases}$$

where $\partial\Omega_i$ denotes the boundary of $\Omega_i$ which is described by the the contours and $dist$ denotes the Euclidean distance within the slices. Now an interpolation of the distance values in $z$-direction is used to find intermediate contours (where the interpolated distance is zero). A practical approach to this iso-surface is to apply a simple MC-algorithm [3]. But to get really good results it is necessary

to use accurate distance fields, that means that in every pixel the exact floating point distance to the contour must be available. Otherwise, staircase artifacts remain in the resulting mesh, see Figure 2.

# 3 The algorithm

In the voxalization step a field function is computed for each voxel (i.e. grid point) in an $n_x, n_y, n_z$ volume. The field function is defined as

$$f(x,y) = \begin{cases} -dist(x,y) & \text{if (x,y) is outside all contours} \\ 0 & \text{if (x,y) is on a contour} \\ dist(x,y) & \text{if (x,y) is inside a contour,} \end{cases} \quad (1)$$

where $dist(x,y)$ is the minimal distance from $(x,y)$ to the set of contours in the slice.

## 3.1 Computation of the distance field

We exploit the graphics hardware to compute the distance field. This approach is similar to the approach of Müller in in the context of collision detection [6]. Let $p_1, \ldots, p_n$ be the polygon describing the contour. Note that the $p_i$ are given in floating point precision and normally do not necessarily lie on pixel centers of our image. For the line-segments of the polygon we define two quadrilaterals in such a way, that the z-value on the plane quadrilaterals define the Euclidean distance to the line-segment. The corresponding surface for points, where again the z-value represents the Euclidean distance from the point is a cone with its apex in the point itself. We approximate this cone with several triangles. Note, that dependent on the angle of successive line-segments only a small part of the cone, approximated by one or two triangles is sufficient. The whole process is shown in Figure 1. Now, all objects defined in this a way are rendered into the z-buffer. After rendering the z-buffer contains the correct distance values and is read out.

One problem that must be solved in this context is that the exact cones are approximated by triangles. In this way the z-values generated by this triangles are not the exact Euclidean distance. Therefore, the distances generated in such a way are not consistent. That means, that the z-buffer may establish correspondences to wrong contours.

| resolution | drawing | z-buffer read |
|---|---|---|
| $300 \times 300$ | 128 | 960 |
| $500 \times 500$ | 136 | 3300 |
| $1000 \times 1000$ | 146 | 12560 |

Table 1: Time for generating the distance fields for the Pelvis in Figure 2. All times are in milliseconds, measured on a PC with a ELSA Office 2000 graphics card. The time is dominated by the z-buffer read.

Inconsisttencies occur especially between the approximation of the cones (pyramids) and the rectangles representing the distance to the edges. Therefore, we sweep the pyramids along the polygon edge and use the resulting surface as the distance function. Fortunately, this sweep surface is simply a transformed rectangle, see Figure 1. Note, that the approximation of the Euclidean distance generated in this way delivers a consistent distance function.

## 3.2 Running the MC-algorithm

After the voxalization step a standard MC-algorithm is used to extract the isosurface for the salar value zero from the voxel data set containing the distance field.

# 4 Examples

We run the algorithm on different data sets. The times to generate the distance fields depend primarily on the time needed to read out the z-buffer. Independent of the complexity of the contours, the drawing time is neglectable, see Table 1.

# 5 Conclusion and future work

For the computation of the distance fields we have proposed an accurate algorithm that exploits widespread standard graphics hardware to be efficient and that can advantageously be applied for other problems where discrete distance fields are needed.

Our algorithm allows to reconstruct surfaces in an accurate but nevertheless efficient way.

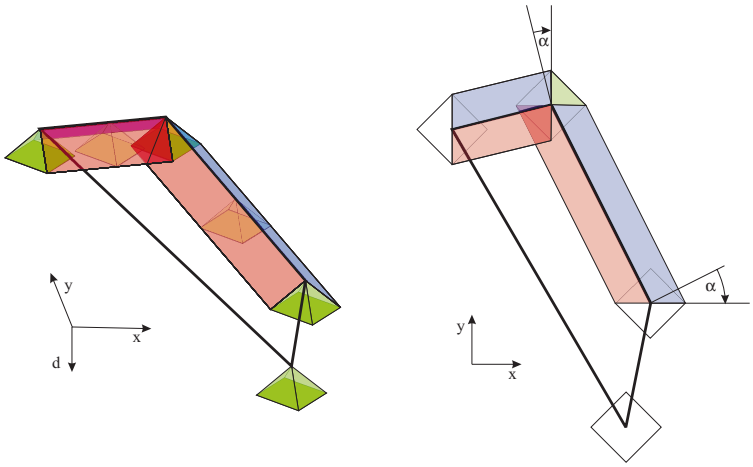


Figure 1: A crude approximation of a cone by a pyramid is swept along a line segment. On the right the top view is shown. The sweep surface between the pyramids on the edges consists of two deformed rectangles. The deformation is determined by the angle $\alpha$. The rectangles inside the polygon are drawn in red, the ones outside in blue. For demonstration the pyramids are colored green. In the algorithm the triangles of the pyramid must also be drawn in red or blue dependend on their location in the inner or outer of the contour. The z-values of the drawing primitives at vertices on the contour are set to zero, at all other vertices to $d$. Inaccuracies caused by the approximation of the cone by triangles are smaller than $1\%$ if the inner angle of the triangles approximating the cones is less than $16.2°$. This means that a cone must be approximated by at least 24 triangles (of which only a few must actually be drawn). To avoid confusion, in the figure the drawing primitives are clipped at a small distance.

Figure 2: Reconstruction of a hip from 23 contours. The dataset is publicly available from Tel Aviv University. On the left side the contours are simply filled ($300 \times 300$ grid). On the right side a distance field was calculated using the new algorithm. Both surfaces were extracted with a standard MC-algorithm. The times for generating the distance-fields are reported in 1.

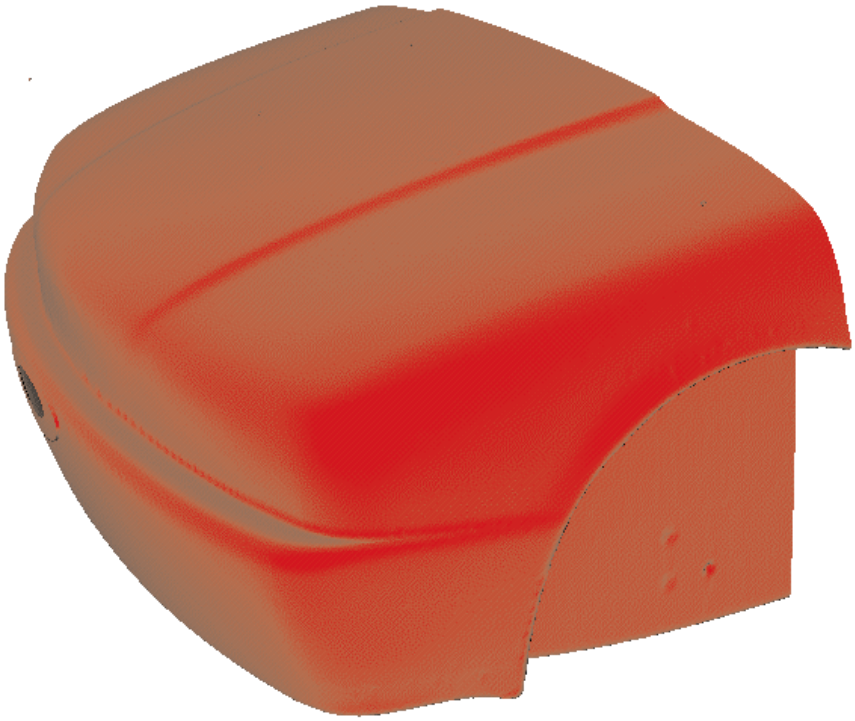

Figure 3: Reconstruction of the front part of a BMW. The slices were generated by a laser range scanner. In the first step the scanner data was converted into polygons. Then the distance field was computed. Original data by BMW AG München.

# 6 Acknowledgement

We would like to thank Holger Müller for all the programming efforts which were necessary to gain the described results.

# References

[1] Daniel Cohen-Or, David Levin, and Amira Solomovici. Contour blending using warp-guided distance field interpolation. In *IEEE Visualization '96*. IEEE, October 1996. ISBN 0-89791-864-9.

[2] Gabor T. Herman, Jingsheng Zheng, and Carolyn A. Bucholtz. Shape-based interpolation. *IEEE Computer Graphics and Applications*, 12(3):69–79, May 1992.

[3] M. W. Jones and Min Chen. A new approach to the construction of surfaces from contour data. *Computer Graphics Forum*, 13(3):C/75–C/84, ???? 1994.

[4] D. Levin. Multidimensional reconstruction by set-valued approximation. *IMA J.Numerical Analysis*, (6):173–184, 1986.

[5] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. In M. C. Stone, editor, *SIGGRAPH '87 Conference Proceedings (Anaheim, CA, July 27–31, 1987)*, pages 163–170. Computer Graphics, Volume 21, Number 4, July 1987.

[6] Heinrich Müller. Using graphics algorithms as subroutines in collision detection. In W. Straßer and F. Wahl, editors, *Proceedings Graphics and Robotics*, Springer Verlag, May 1994.

[7] Bradley A. Payne and Arthur W. Toga. Distance field manipulation of surface models. *IEEE Computer Graphics and Applications*, 12(1):65–71, January 1992.