Optimal Surface Reconstruction from Digitized Point Data using CI Methods

Prof. Dr.-Ing. K. Weinert
Dipl.-Inform. J. Mehnen, Dipl.-Inform. G. Prestifilippo
ISF, Department for Machining Technology,
University of Dortmund, Germany

January 13th, 1997

Keywords: Triangulations, digitized point data, surface reconstruction, quality characterization of triangulations, polyhedral approximation, optimization, α-look-ahead method, shortest-path method, simulated annealing, computational intelligence.

Abstract

In many scientific and technological endeavors, a three-dimensional solid must be reconstructed from digitized point data. This paper presents three solutions to the problem of reconstructing smooth surfaces using triangular tiles. The presented algorithms differ in their strategic approach. Here, two deterministic algorithms and one non-deterministic CI (computational intelligence) strategy will be described. In order to compare triangulations, two quality criteria will be introduced.

1.1 Motivation

Practical applications of tactile or optical scanning methods generate huge sets of many thousands of weakly structured three-dimensional point co-ordinates. These are usually very hard to handle using common CAD systems. CAD systems are expecting smooth Bézier surfaces, NURBS (*uniform rational B-splines*) or B-splines rather than a large set of discrete points. Due to approximation theoretical and practical reasons, it is not useful to span a complete surface with only one spline. In order to use tensor product surfaces, discrete regions in the three-dimensional data set have to be determined. The main problem in surface reconstruction is the partitioning of the set of digitized point data into subsets that resemble the *construction logic* employed by a CAD-designer.
One way towards that admittedly lofty mission is the approximation of digitized three-dimensional data via triangulations. The algorithms introduced here determine *smooth triangulations* from digitized point sets.
The first two algorithms are deterministic and make use of the line structure of the digitized point data sets. They result in rather good surface qualities and need only a short processing time. Due to the fact that triangulation is a difficult multimodal optimization problem, i.e. there exist several suboptimal solutions, the *α-look-ahead algorithm* compares the quality of different solutions to avoid getting

caught in local minima. The *shortest path strategy* finds better but usually still suboptimal solutions in $O(n \, log(n))$-time.
Although many deterministic algorithms show high performance, an optimal triangulation usually cannot be found using only local information. CI (*computational intelligence*) methods present a new approach towards global optimization problems. CI covers the fields of Fuzzy Logic (FL), Neural Networks (NN) and Evolutionary Algorithms (EA).
An *EA* approach towards the global optimization problem of triangulation will be introduced here. The potential of the algorithm to avoid getting caught in local optima can be shown. This strategy does not make use of local information like the line structure of a given point set.
In order to compare the quality of triangulations objectively, smoothness criteria have to be defined. Here, the *max-min criterion* and the *total absolute curvature criterion* will be introduced.

1.2 Characterizing the Quality of Triangulations

In the literature, several triangulation quality criteria can be found [1, 2, 3]. Looking at a shaded triangulation, it is very easy to distinguish between smooth and jagged surfaces. Typically, in computer graphics, two objective quality definitions for triangulations are used: triangle-based criteria and edge-based criteria.
Triangle-based criteria follow the rule of maximization or minimization, respectively, the angles of each triangle. The so-called *max-min angle criterion* prefers short triangles with obtuse angles.

Definition: *(max-min angle criterion)* For each triangle T in a triangulation Δ, let $m(T)$ be the smallest angle in T, and let $m(\Delta) = min_{T \in \Delta} \; m(T)$. We define $C(\Delta, \tilde{\Delta}) = m(\Delta) - m(\tilde{\Delta})$ to be a metric for comparing triangulations. The triangulation Δ is *better* than triangulation $\tilde{\Delta}$ with respect to C provided that $C(\Delta, \tilde{\Delta}) \; < \; 0$.
Under this criterion, an optimum triangulation is one whose smallest angle is maximal. It is known that a triangulation is optimal with respect to the max-min angle criterion if and only if it is a Delaunay triangulation [3].

Edge-based criteria describe curvature-relations of triangles that share one common edge. Making use of the classic differential geometry, v. Damme and Alboul [4] propose the *total absolute curvature (tac) criterion.* The main idea of this criterion is to measure the curvature of a triangulated surface in a discrete point by calculating the sum of the 3D-angles of the triangles sharing this point.

Definition: *(total absolute curvature (tac) criterion)* Let P be a point with d neighborhood points $N_1, \ldots, N_d$ ($N_{d+1} = N_1$). A point belongs to the neighborhood of P, if it is a vertex of a triangle that shares the common point P. Let α_i,

$i = 1, \ldots, d$ be the 3D-angles of the triangles around P. Then the total absolute curvature is defined to be:

$$tac(P) = |2\,\pi - \sum_{i=1}^{d} \alpha_i|.$$

In order to compare triangulations, a total order depending on different metrics has to be defined. For triangle-based methods, a quality vector $M(T) := (m(T_1), \ldots, m(T_N)) \in R^N$, where $m(T_i)$ is a triangle-based criterion (like the max-min-criterion) applied to the triangle T_i, has to be calculated. For edge-based criteria, a quality vector may look like: $M(T) := (m(e_1), \ldots, m(e_M)) \in R^M$, where e_i are the edges of a triangulation Δ. A quality vector using the *tac*-criterion can be written like: $M(T) := (tac(P_1), \ldots, tac(P_v)) \in R^v$, where P_i are the interior points of a digitized surface.
A total relation for comparing two triangulations T and T' can be defined via the p-norm:

$$M(T) \leq_p M(T') :\Leftrightarrow (\sum_{i=1}^{d} |m(t_i)|^p)^{1/p} \leq (\sum_{i=1}^{d} |m(t_i')|^p)^{1/p}$$

where $m(t_i)$ is the $i-$th element of the vector $M(T)$, d the number of elements in T.

1.3 Triangulation Strategies

After the definition of two quality criteria and a total relation on triangulations, we have objective tools to compare triangulation strategies with each other. Typically, there exist two ways of solving optimization problems. Depending on complexity and knowledge about the structure of a problem, either deterministic or non-deterministic optimization algorithms can be used. Deterministic strategies are very efficient when the solution can be given explicitly and the complexity of the problem is polynomial. If there exists no explicit knowledge about the "path" towards an optimum but optima can be characterized efficiently, non-deterministic algorithms are a good choice.

Concerning the typical line structure of digitized point data, Keppel [5] calculates the number of possible triangulations between two lines with each m and n points, respectively, to be

$$\frac{(n+m)!}{n!\;m!}.$$

Thus, the algorithmical complexity of the triangulation problem is exponential.

Both deterministic algorithms described here make use of the line structure of the digitized point data. This structure can be transformed into a corresponding matrix, which was first mentioned by Fuchs et al. [6]. An entry $m_{i,j} = w_{i,j}$ in this matrix corresponds to an edge between to points a_i and b_j taken from the

digitalizing lines $(a_1, \ldots, a_n) \in (R^3)^n$ and $(b_1, \ldots, b_m) \in (R^3)^m$. The weight $w_{i,j}$ characterizes the quality of an edge by any arbitrary quality criterion (e.g. see 1.2). Any other entry in the matrix should be chosen appropriately.
Making use of this definition, Friedhoff has proposed the so called *α-look-ahead-strategy* [7]. The main idea of this method is to find an optimal path in the "Fuchs matrix". This path corresponds to an optimal triangulation between each two lines of digitized point data. The difficulty in finding an optimum is that only *local* quality information at each point is evaluated in order to gain a *globally* optimal triangulation. The α-look-ahead-strategy chooses the "cheapest" path in the matrix by selecting the minimum sum from two weighted paths with length α. The algorithm has $O(n)$-algorithmic complexity. Therefore, the α-look-ahead-strategy produces triangulations in a very short time. Choosing $\alpha = n$, the algorithm shows $O(n^2)$ complexity. Unfortunately, "well-shaped" triangulations can only be expected when the lines run approximately parallel in the 3D-space.
The *shortest-path algorithm* proposed first in this article finds a minimal path through the "Fuchs matrix" using Dijkstra's $O(n\ log(n))$-time algorithm [8]. The algorithmical complexity of the shortest-path algorithm is $O(m^2\ log(m^2))$, where m is the maximum number of points in all lines. This strategy is guaranteed to calculate the shortest path between two points of a weighted graph using an adjacency list and a priority queue. The adjacency list contains the corresponding edges of each vertex. The priority queue is realized via a heap structure containing the weights of each edge. The solution of this strategy is always better or equal to the solution discovered by the α-look-ahead-strategy.
The problems arising from evaluating local information is common to both strategies.

1.4 CI-method for Optimization of Triangulations

Bezdek subsumed the fields of *Fuzzy Logic*, *Neural Networks* and *Evolutionary Algorithms* under the notion of *Computational Intelligence (CI)* [9]. The common nature of this class of universal adaptive algorithms is that they work subsymbolically. That is, in contrary to the methods of classic symbol-oriented Artificial Intelligence algorithms, they solve problems numerically. The foundation of Evolutionary Algorithms (EA) has been laid in the early Fifties. The notion of EA subsumes the class of inherently parallel and iterative algorithms which are software analogies of processes and structures of organic evolution. The Genetic Algorithms (GA), as proposed by Holland [10], Evolutionary Programming (EP) of Fogel et al. [11] and the Evolutionary Strategies (ES) of Schwefel and Rechenberg [12, 13] belong to this class. Simulated Annealing (SA) is another optimization strategy which is a software analogy of the annealing process of hot metals [14].
GAs are typically used for solving classification tasks. EPs can solve time series prediction tasks using finite automata. ES are typically used for solving vector-optimization tasks in the R^n. ES also serve very well in structural optimization.

At the Department of Machining Technology (ISF), first experiments have been made in order to optimize triangulations with this class of algorithms.
SA algorithms base on the concept of annealing, which is derived from materials science, where it is used to describe the process of eliminating lattice defects in crystals by a procedure of heating, followed by slow cooling. SA proceeds analogously. One considers an ensemble of arrangements weighted by the Bolzmann factor $exp(-\frac{E}{T})$, where E is the cost function (often called fitness or quality criterion) and T is a parameter that plays the role of the temperature. On lowering the value of T, the existence of unfavorable arrangements becomes less and less likely until, hopefully, the optimal solution remains at $T = 0$.
The advantage of EA methods compared with deterministic strategies is that the way towards the solution of a complex problem does not have to be described explicitely. Here, a quality criterion, adequate evolutionary operators (mutation, selection, recombination for instance) and an encoding of the cost function domain have to be defined.
Following Schumaker, the encoding of the function domain has been chosen to be a vector of edges that corresponds to an arbitrary triangulation [3]. When using an EA, an initial triangulation is needed. One of the algorithms described above can be used to yield this triangulation.
An edge-exchange operator in the 2D-space has been defined to serve as a mutation operator. It flips an edge within a convex polygon of four points. The polygon has to meet the condition that three of the four points are not collinear [1, 3]. The advantage of the application of a 2D-operator lies in the fact that special cases can be ignored which appear when edges are flipped in 3D-space. Hence, a simple projection of the digitized 3D points into 2D-space has to be performed. The mutation operator flips an edge by randomly choosing elements of the edge vectors in an equally distributed way.
Another advantage of EAs is the applicability of different fitness functions while the function domain remains the same. Here, the tac criterion (see 1.2) has been used. Following the selection scheme of a SA, the algorithm keeps solutions that show an improvement after each mutation step. A vector that shows no improvement survives selection due to the actual temperature. The idea of keeping vectors that do not show an improvement is to escape from local minima by random steps.
The non-deterministic algorithm presented here does not depend on any structural information within the given point data set. The only condition that has to be fulfilled is that the digitized surface must not show re-entrant angles, that is the surface can be projected to a 2D-plane uniquely.

1.5 Comparison of the Optimization Algorithms

Comparing deterministic with non-deterministic strategies it is noticeable that deterministic methods give regular triangulation structures. This can be seen in figure 1 (right-hand side) where the symmetric mesh structure of the triangulated surface of a segment of a connecting rod – generated by the shortest-path strategy – is shown. Smooth surfaces appear in figure 1 where the digitalizing lines run nearly parallel in 3D-space. A rough surface structure appears in steep regions of

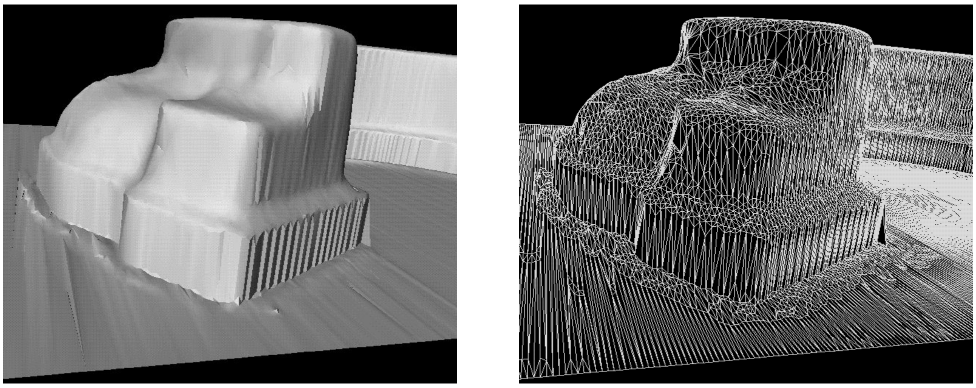

Figure 1: Shortest-path strategy

the surface. This is due to the fact that the deterministic algorithms described here optimize a triangulation only locally between two digitalization lines. This corresponds to the feature of deterministic strategies of finding only suboptimal solutions usually. The non-deterministic SA does not need a specific line representation. Thus, optimal triangles can be calculated by just considering the tac fitness

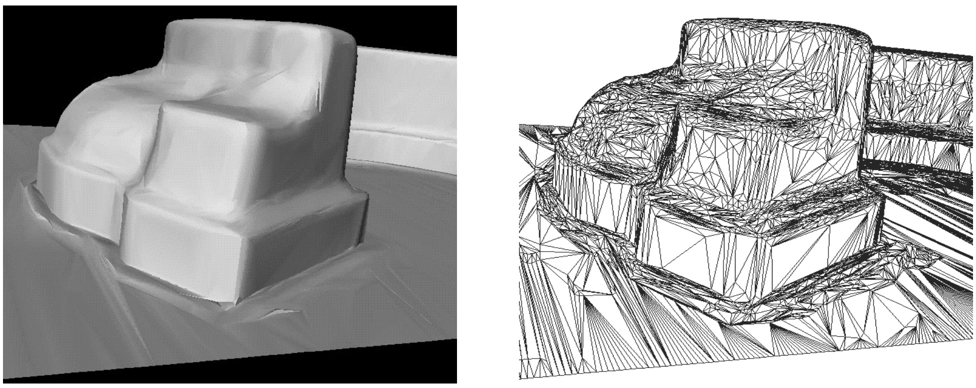

Figure 2: Simulated annealing

function. One can see from the mesh structure in figure 2 (right-hand side) that it seems more appropriate to choose obtuse-angle triangles for approximating plane surfaces. The SA selects long triangles with acute angles to fit regions with high curvature values. This CI method allows to find smooth surfaces which even satisfy optical demands of smoothness. For application tasks, the trade-off between quality and time has to be taken into account. The SA triangulation of the connecting

rod (14,339 points, 3,000 iterations) takes 156 minutes. The shortest-path method takes only 15 seconds for the same problem.
Non-deterministic methods represent new ways of optimizing triangulations. The structures built from triangles by self-organization effects in the SA context enable the formulation of new approximation-theoretical statements that may help to develop even more efficient algorithms.

References

[1] N. Dyn, D. Levin, and S. Rippa. Data Dependent Triangulations for Piecewise Linear Interpolation. *In: IMA Journal of Numerical Analysis*, 10:137–154, 1990.

[2] L. Schumaker and E. Quak. Cubic spline fitting using data dependent triangulations. *In: Computer Aided Geometry Design*, 7:293–301, 1990.

[3] L. Schumaker. Computational triangulation using simulated annealing. *In: Computer Aided Geometry Design*, 10:329–345, 1993.

[4] R. van Damme and L. Alboul. Polyhedral metrics in surface reconstruction. *Tight Triangulations, technical report*, 1994.

[5] E. Keppel. Approximation Complex Surfaces by Triangulations of Contour Lines. *In: IBM Journal of Research Development*, 19:2–11, 1975.

[6] H. Fuchs, Z.M. Kedem, and S.P.Uselton. Optimal Surface Reconstruction from Planar Contours. *In: Communications of the ACM*, 20(10):693–702, 1977.

[7] J. Friedhoff. *Aufbereitung von 3D-Digitalisierdaten für den Werkzeug-, Formen- und Modellbau.* University of Dortmund, Germany, 1997.

[8] R. Sedgewick. *Algorithmen*, volume 1. Addison Wesley, Bonn, 1991.

[9] J.C. Bezdek. What is computational intelligence ? *In: J.M. Zurada (eds.): Computational Intelligence: Imitating Life. IEEE Press*, 4:1–12, 1994.

[10] J.H. Holland. *Adaption in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, MI, 1975.

[11] L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial Intelligence through Simulated Evolution.* Wiley, New York, 1966.

[12] H.-P. Schwefel. *Evolutionsstrategie und numerische Optimierung, Dissertation.* Technische Universität Berlin, 1975.

[13] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Dissertation.* Technische Universität Berlin, 1971.

[14] S. Kirpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization of Simulated Annealing. *Science*, 220:671, 1983.