# Netbased Modelling

Lars Linsen*

Universität Karlsruhe, Germany†

## Abstract

Given objects in boundary representation, i.e. piecewise polynomial surfaces, modelling operations such as union, intersection, and difference, as well as trimming and rounding off for objects of arbitrary shape are usually be solved approximately. In this paper a new, simpler approach is introduced. Rather than working with the polynomial surfaces we use their control nets or an interpolating net. Thus the modelling operations can be realized on nets. Due to the widespread field of possible applications various methods will be discussed.

**Keywords:** Discrete Modelling, Interpolating Nets, Control Nets, Union, Cut, Difference, Intersection, Trimming, Fairing.

## 1 Introduction

Intersecting and joining objects is a topic often required and considered in geometric modelling and computer graphics, since it is an elementary tool for creating new objects. A well-known example is the *Constructive Solid Geometry (CSG)*, where one uses simple basic solids and some modelling operations like union, intersection, and difference to construct more complex ones.

With free-form surfaces modelling operations are difficult to compute and in general can be done only approximately. In this paper we suggest therefore to perform modelling operations with nets representing the surfaces rather then with the surfaces themselves.

**Related work:**
CSG is a standard technique in computer graphics and well described in the literature. In the eighties first attempts were made to combine CSG with free-form surfaces. Either free-form solid primitives were allowed (cf. [Guo/Menon '96]) or the modelling operations were extended, e.g. by surface-solid operations. A well-structured survey is given in [Várady/Pratt '84].

However, all these approaches apply the modelling operations to the free-form surfaces themselves, which requires sophisticated computations of the curves of intersection. The approach of this paper uses net-representations of free-form surfaces, which in particular simplifies the computation of the intersection.

The major remaining problem is the merging of two nets along their intersection. A similar problem occurs, building triangular nets from three-dimensional contour lines. In this field some work is presented in [Choi/Park '94], [Christiansen/Sederberg '78], [Ekoule et al. '91], [Klingert '94], and [Meyers et al. '92], who consider also branching techniques, but always focus on the aesthetics of the resulting triangles.

Hardly any work was done, on how to fulfill certain topological constraints. [Talbert/Parkinson '90] try to convert a triangular net into quadrilateral elements, whereas [Joe '95] tries to create quadrilateral nets within polygonal regions, and [Schreiber '95] shows a way to improve the topological regularity of quadrilateral nets.

**Definition:**
To avoid misunderstandings, a brief explanation should be given about what is meant by *nets*. A net consists of points $\in \mathbb{R}^3$, called vertices, connected by edges. Three or more connected vertices, each of which has exactly two edges to the others, build a mesh of the net. We assume that every edge contributes to at most two meshes.

Therefore every net is a two-dimensional manifold and corresponds to a continuous surface.

In particular if each edge contributes to exactly two meshes, the net has no boundaries. Surfaces of solids can be represented by such closed nets.

In a regular triangular net each mesh has three edges and each interior vertex[1] the valence six, whereas in a regular quadrangular net each mesh has four edges and each interior vertex the valence four. Control nets fulfill such *topological constraints* with some extraordinary vertices or meshes.

**Overview:**
In the following section we discuss how to find the intersection of two nets, in Section 3 we describe how to connect the desired parts of the nets along the intersection, in Section 4 we show another possibility for joining two nets by creating connecting meshes, and demonstrate the results in Section 5. The sixth

*llinsen@ira.uka.de

†http://i33www.ira.uka.de/~llinsen

[1] An interior vertex is a vertex, which does not contribute to a boundary of the net.

section is dedicated to further modelling operations, i.e. the trimming of nets and the construction of fair blendings.

## 2 Intersection

The first step for constructing a combined net out of two intersecting nets, is to cut off the jutting parts.

This is not straight-forward because we need to compute the intersections of edges of a net with meshes of a second net.

In the simplest case the second net consists only of triangular meshes, the problem reduces to line-triangle intersections.

In case the second net has non-triangular meshes we need to compute the intersection of a line and a non-triangular mesh. This is not well-defined, because the vertices of a non-triangular mesh do not lie in a plane in general. Approximating meshes by planes does not eliminate the ambiguity and causes further problems due to gaps between the approximating planes of adjacent meshes. Thus we divide each $n$-lateral mesh into triangular meshes.

To do so we split an $n$-lateral mesh with vertices $\mathbf{p}_1, \ldots, \mathbf{p}_n$ into $n$ triangular meshes by connecting all vertices with the centroid $\frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_i$.

Now the intersections of the edges of the first net with the meshes of the second net can be calculated, and vice versa. In the next section we show, how to connect the required parts of the two nets using these intersection points.

## 3 Connecting the points of intersection

Before two nets can be cut off at the points of intersection and be stuck together to a new net, it has to be decided which parts of the original nets contribute to the new net and which parts of the original nets can be omitted.

Figure 1 demonstrates the various possibilities. The pear and the cylinder shown in the figure intersect twice. Thus for the pear as well as for the cylinder the decision has to be made whether the outer or the inner parts should be left out leading to the four different solutions illustrated in the figure.

*Remark:* If the nets of the pear and of the cylinder are regarded as the surface representations of two solids[2], the four possibilities realize different modelling operations like in CSG-systems, namely

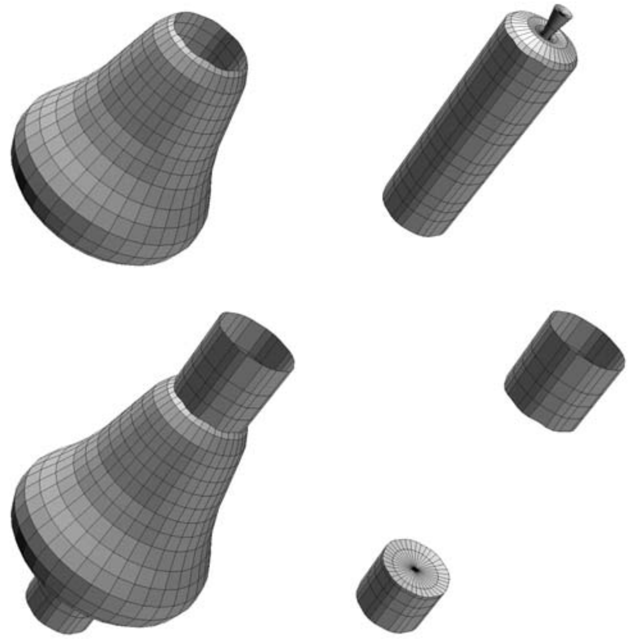


Figure 1: Decision of removableness

- pear $\setminus$ cylinder
- pear $\cap$ cylinder
- pear $\cup$ cylinder
- cylinder $\setminus$ pear

Now after the decision of removableness has been made, an *intersection polygon* is built by connecting each point of intersection with its (generally two) "neighbours". Then cutting off the original net is realized first by adding this polygon to the net as new vertices and new edges, second by connecting the points of intersection with the not removed vertices of the intersected edges of the original net, and finally by throwing away everything that lies beyond the polygon of intersection. Thus the polygon is part of the boundary of the remaining part of the net.

This has to be done for both nets symmetrically, leading to a situation of two new nets, which are parts of the original ones and which should join each other in an area around the calculated intersection polygons.

Note that the intersection may consist of several components. For example, the pear and the cylinder shown in Figure 1 intersect twice leading to two intersection polygons for each net. The intersection polygons of both nets can be paired.

Although both polygons of such a pair represent similar curves they are not equal in general. Therefore we merge both polygons to a single polygon, adapt both nets to that polygon and finally join both nets along this new polygon.

To merge two polygons we determine a correspon-

[2]The cylinder needs to be closed by lids.

dence between their vertices. The merging polygon we construct is given by the midpoints of corresponding vertices. Further we insert the vertices of the original polygons, that have no correspondence, into the merging polygon.

Four different types of correspondences have been constructed, which are illustrated in Figure 2 for two planar nets intersecting in a straight line.

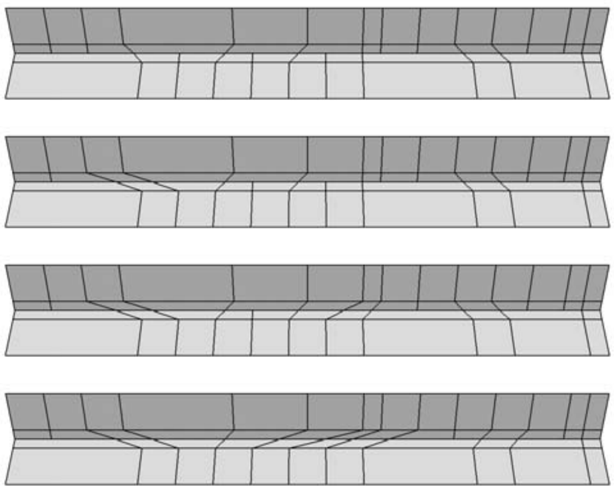


Figure 2: The different correspondences lead to different results, favouring aesthetic or topological aspects.

For the first correspondence we try to construct only short edges and roughly equal angles in every mesh to obtain visually pleasing nets. Every vertex corresponds to the closest vertex of the paired polygon if possible.

For the last correspondence we try to preserve the topology of the original nets accepting also strange-looking stretched meshes. It is guaranteed, that every vertex of the polygon, which contains a lower number of vertices, corresponds to a vertex of the paired polygon.

For the two other correspondences we try to find a compromise between the optimization of aesthetics and the optimization of topological regularity. Topological considerations are restricted to a local area. The two methods differ in the size of the area.

For details see [Linsen '97].

Note that due to the granularity of the nets, irregularities can not be avoided in general. Hence optimizing the regularity means to minimize the quantity of extraordinary vertices and meshes.

# 4 Connecting by meshes

In this section we introduce another possibility for net connection. Figure 3 sketches, why it may be useful: If one wants to maintain the topology, inserting the points of intersection as new vertices might lead to an unpleasant solution as demonstrated on the left side of the figure.

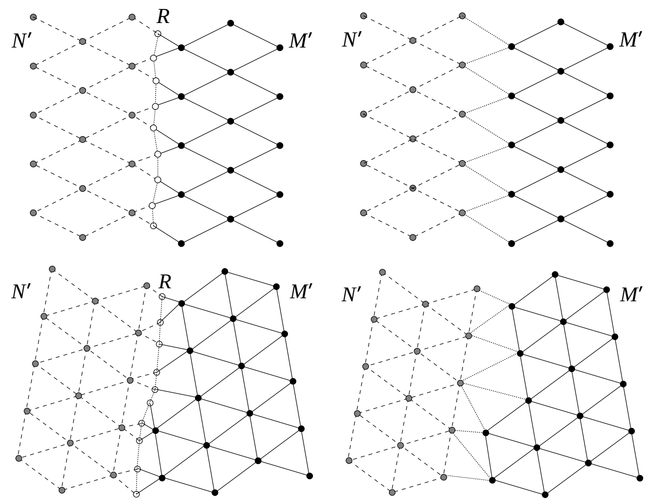



Figure 3: Reason for introducing a new method:
left: Connecting the points of intersection
right: Connecting by meshes

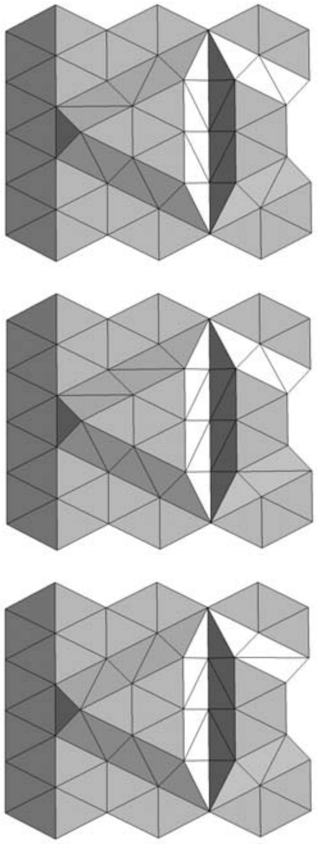
Connecting the nets by constructing new meshes in the way shown on the right avoids such topological irregularities.

*Remark:* Especially when we apply this method to triangular nets the resulting nets are triangular, too, whereas by the method of Section 3 non-triangular meshes are produced frequently.

So for the method of connecting by meshes there is no need for splitting up meshes into removable and remaining submeshes anymore. Instead, all meshes with any points of intersection are left out completely. Certainly the meshes, that contribute to the removable parts of the net, are omitted again. This produces two new nets with narrow gaps between them. These gaps characterize the regions, where the original nets intersect, and can be described by pairs of polygons, that contain the not removed, adjacent boundary vertices in correct order. These polygons need to be related, whereby new problems such as *branching* occur. Branching means, that one boundary polygon of a net is related with two or more boundary polygons of the other net. This has to be solved by splitting up the single polygon.

After having defined the gaps, they must be closed by new meshes. Figure 4 shows three possibilities for triangular meshes.

The triangular topology can always be maintained. In addition, in method (a) the lengths of the new edges are minimized, whereas in methods (b) and (c) we try to construct vertices with valence 6. In method (b) we realize a unilateral topological optimization still regarding the length of the edges for not creating very unpleasant solutions. In method (c) we realize a bilateral topological optimization, where aesthetical aspects are only considered, when the valences of the vertices do not lead to a unique solution. The result of method (c) applied to the figured example has only one unavoidable irregular vertex.



(a) (b) (c)

Figure 4: Connecting by triangular meshes:
(a) Minimization of the length of the edges
(b) Unilateral topological optimization
(c) Bilateral topological optimization

If arbitrary triangular nets are used, no further topological optimization are necessary and method (a) gives the best results, i.e. the most pleasant.

Also for quadrilateral nets three different methods for constructing connecting meshes have been established. Figure 5 illustrates a method similar to method (c) from above with a bilateral topological optimization of the valences of the vertices. However, it is more difficult: Neither non-quadrilateral meshes nor vertices of valence $\neq 4$ can be avoided. Thus, also the number of edges per mesh has to be considered during the construction.

For certain applications it is useful to avoid irregular meshes or irregular vertices. Therefore a method for minimizing the number of irregular vertices is presented as well as a method for minimizing the number of irregular meshes. The results are shown in Figure 6 and Figure 7, respectively.

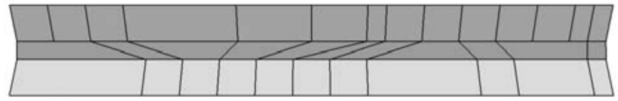


Figure 5: Connecting by quadrilateral meshes and considering the valences of the vertices

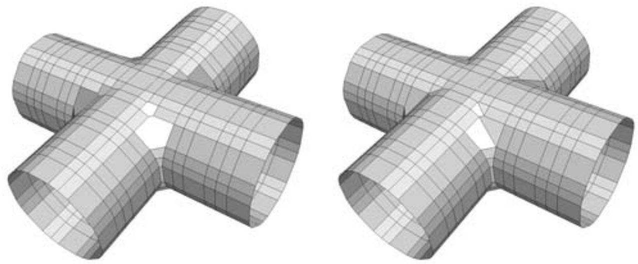


Figure 6: Connecting by quadrilateral meshes and minimizing the number of irregular vertices with (right) and without (left) post-process

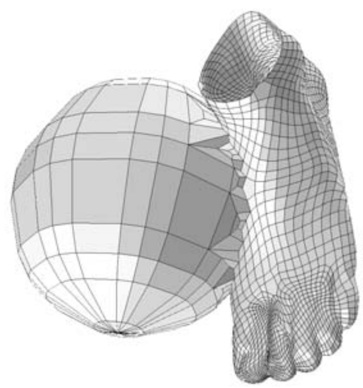


Figure 7: Connecting by quadrilateral meshes and minimizing the number of irregular meshes

By the method of minimizing the number of irregular vertices meshes with many edges may arise. They can be split up into smaller meshes in a post-process.

By the method of minimizing the number of irregular meshes only quadrilateral meshes are constructed with at most one exception. It divides large meshes into four smaller meshes by inserting two new vertices. This method is especially useful when the two original nets have remarkable differences in the size of their meshes.

## 5 Results, discussion, and applications

Having introduced various methods for intersecting and connecting two nets, we discuss, what results could have been achieved due to some possible applications.

First we look at Constructive Solid Geometry. Figure 8 shows the Boolean combination of four objects, a cube and three cylinders.

This object can be described by $cube \cup cylinder1 \backslash cylinder2 \backslash cylinder3$.

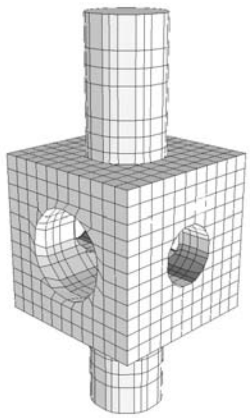


Figure 8: Constructive Solid Geometry: $cube \cup cylinder1 \backslash cylinder2 \backslash cylinder3$

For this application the method of connecting the points of intersection is the most practicable because adding the points of intersection to the net gives sharp contours instead of rounded edges.

As already mentioned, the triangular net topology can only be maintained by the method of Section 4. Figure 9 shows another example, where two non-regular triangular nets are joined.

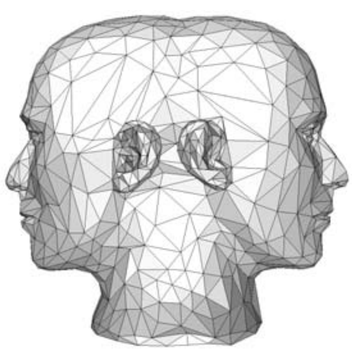


Figure 9: "Roman God Janus" constructed out of two heads with the method of connecting by triangular meshes

The new constructed meshes fit well into the resulting net and fill the gap without strange-looking artifacts, thus forming a nice blending.

Another application is shown in Figure 10. It is a shuttle constructed out of five parts, namely the trunk and the four wings.

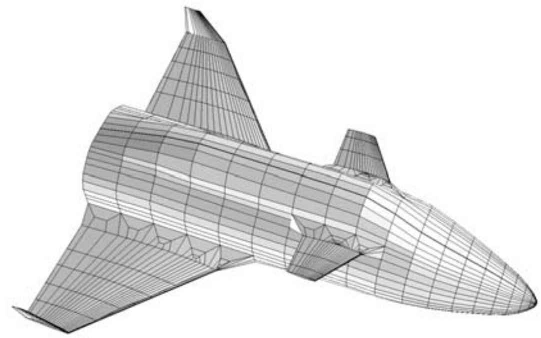


Figure 10: A possible application: Reverse Engineering

Finally we give an example where it is useful to maintain certain net characteristics. The nets shown in Figure 11 and Figure 12 are control nets of the surfaces also shown in these figures. More precisely, the surfaces are free form splines as described in [Prautzsch '97].

*Example:* If two cylinders or pipes are to be blended smoothly, their control nets can be united using the method of connecting by quadrilateral meshes as shown on the upper part of Figure 11.

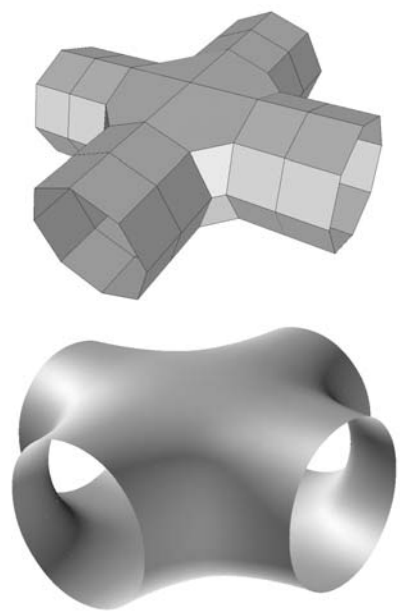


Figure 11: Modelling of tensor product spline surfaces

In the same manner we computed the union of a goblet and a chessman, see Figure 12.

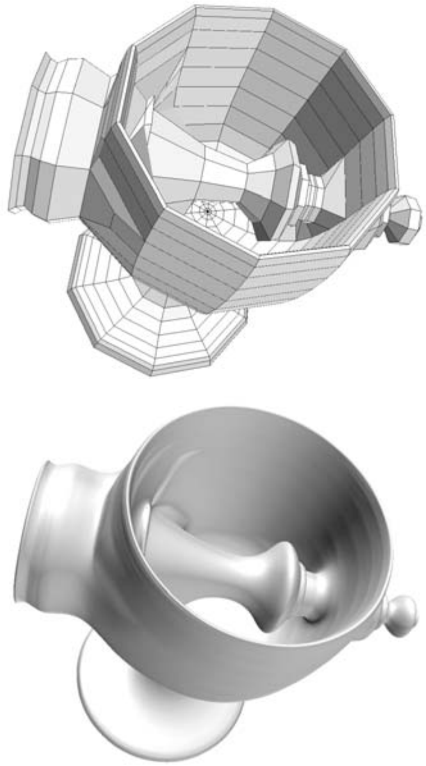


Figure 12: *goblet* ∪ *king* as $C^2$-spline surfaces

# 6 Enhanced modelling operations

**Trimming:**
Trimming is another application, which can easily be done by the described methods. Given are a net and a curve, which lies on the surface represented by the net. The curve can be discretized and extended in a direction (locally) orthogonal to the net, which leads to a second net.

Now the given net can be trimmed by intersecting and cutting it off at the second net as shown in Sections 2 and 3.

Figure 13 gives an example.

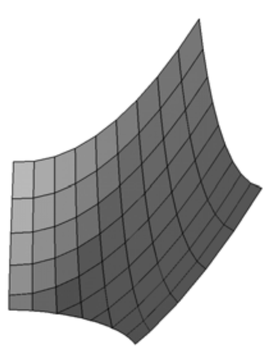


Figure 13: Trimmed net of a quadratic surface

**Rounding off:**
After having joined two nets, some designer may have the wish to round off the blendings. This can be done by optimizing the position of the vertices with respect to some energy.

*Variational methods* are used for this purpose. Only the positions of the vertices of the truncated meshes (in case of using the method "connecting the points of intersection") or of the connecting meshes (in case of using the method "connecting by meshes") are variable, while all the others are fixed.

Since we need a discrete analogue to a curvature definition, we use the *umbrella* $\Delta\mathbf{p}$ of a vertex $\mathbf{p}$ (cf. [Kobbelt '95]). Then the curvature at the point $\mathbf{p}$, which is connected with the vertices $\mathbf{q}_1, \ldots \mathbf{q}_n$, is given by

$$Curvature(\mathbf{p}) \ := \| \Delta\mathbf{p} \|_2 := \| \mathbf{p} - \frac{1}{n}\sum_{i=1}^{n} \mathbf{q}_i \|_2 \quad .$$

For higher moments the umbrella definition can be raised to a higher power, i.e. the $(2k)$-th moments are given by

$$\| \Delta^k \mathbf{p} \|_2 := \| \Delta^{k-1}\mathbf{p} - \frac{1}{n}\sum_{i=1}^{n} \Delta^{k-1}\mathbf{q}_i \|_2 \quad ,$$

while the $(2k+1)$-th moments can be expressed by

$$\frac{1}{n}\sum_{i=1}^{n} \| \Delta^k\mathbf{p} - \Delta^k\mathbf{q}_i \|_2 \quad .$$

From the latter the formula for the torsion can be derived for $k = 1$. This leads to the weighted energy of moments

$$E := \sum_{\mathbf{p}} ( \quad \begin{array}{l} \mu \cdot (Curvature(\mathbf{p}))^2 + \\ \nu \cdot (Torsion(\mathbf{p}))^2 + \ldots \ ) \end{array} \quad ,$$

where the higher moments are less important.

The results for choosing the new locations of the variable vertices by minimizing that energy functional are shown in Figure 14.

# 7 Conclusion

In this paper we have introduced modelling operations for discrete nets.
The considered modelling operations are:

- union
- cut
- difference
- trimming

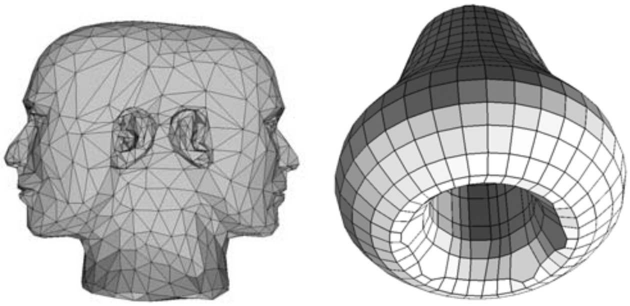


Figure 14: Fair blendings for triangular (left) and quadrangular (right) nets using the umbrella definition

- rounding off

In particular the nets can be control nets of free form splines. Our modelling operations therefore also yield approximate modelling operations for free form surfaces.

The modelling operations are based on the intersection and connection of nets. We presented different methods to compute an intersection and union of two nets depending on the net type and the application.

## References


[Borgmeier '94] E. Borgmeier: *Rekonstruktion chaotischer Flächen mittels Triangulierung und deren Visualisierung.* Master's thesis, Universität Karlsruhe, 1994.

[Campagna/Seidel '97] S. Campagna, H.-P. Seidel: *Generating and displaying progressive meshes.* 3D-Image Analysis and Synthesis, Erlangen, 1997, 35 - 42.

[Choi/Park '94] Y.K. Choi, K.H. Park: *A heuristic triangulation algorithm for multiple planar contours using an extended double branching procedure.* The Visual Computer, Vol. 10 (7), 1994, 372 - 387.

[Christiansen/Sederberg '78] H.N. Christiansen, T. W. Sederberg: *Conversion of complex contour line definitions into polygonal element mosaics.* Computer Graphics, Vol. 12, 1978, 187 - 192.

[Ekoule et al. '91] A.B. Ekoule, F.C. Peyrin, C.L. Odet: *A triangulation algorithm from arbitrary shaped multiple planar contours.* ACM Transactions on Graphics, Vol. 10 (2), 1991, 182 - 199.

[Greiner '94] G. Greiner: *Variational design and fairing of spline surfaces.* Computer Graphics Forum, Vol. 13 (3), 1994, 143 - 154.

[Greiner et al. '96] G. Greiner, J. Loos, W. Wesselink: *Data dependent thin plate energy and its use in interactive surface modeling.* Computer Graphics Forum, 1996.

[Guo/Menon '96] B. Guo, J. Menon: *Local shape control for free-form solids in exact CSG representation.* Computer Aided Design, Vol. 28 (6/7), 1996, 483 - 493.

[Hamann '94] B. Hamann: *A data reduction scheme for triangulated surfaces.* Computer Aided Geometric Design, Vol. 11, 1994, 197 - 214.

[Hoppe et al. '93] H. Hoppe, T. DeRose, T. Duchamp, J. MacDonald, W. Stuetzle: *Mesh Optimization.* Computer Graphics Proceedings, Annual Conference Series, Vol. 7, 1993, 19 - 26.

[Joe '95] B. Joe: *Quadrilateral mesh generation in polygonal regions.* Computer Aided Design, Vol. 27 (3), 1995, 209 - 222.

[Klingert '94] A. Klingert: *Rekonstruktion von geometrischen 3D-Objekten durch Interpolation von Konturdaten.* Ph.D. thesis, Universität Karlsruhe, Verlag Shaker, Aachen, 1994.

[Kobbelt '95] L. Kobbelt: *Iterative Erzeugung glatter Interpolanten.* Ph.D. thesis, Universität Karlsruhe, Verlag Shaker, Aachen, 1995.

[Kobbelt '96] L. Kobbelt: *Discrete Fairing.* Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces, 1996.

[Linsen '97] L. Linsen: *Schnitt und Vereinigung von Kontrollnetzen.* Master's thesis, Universität Karlsruhe, 1997.

[Meyers et al. '92] D. Meyers, S. Skinner, K. Sloan: *Surfaces from contours.* ACM Transactions on Graphics, Vol. 11 (3), 1992, 228 - 258.

[Müller '97] Th. Müller: *Glatte Splineflächen aus Kontrollnetzen beliebiger Topologie.* Master's thesis, Universität Karlsruhe, 1997.

[Prautzsch '97] H. Prautzsch: *Freeform Splines.* Computer Aided Geometric Design, Vol. 14 (3), 1997, 201 - 206.

[Schreiber '95] R. Schreiber: *Kombinatorisch optimierte Konstruktion fast-regulärer Polygonnetze.* Ph.D. thesis, University of Stuttgart, Tectum Verlag, Marburg, 1995.

[Talbert/Parkinson '90] J.A. Talbert, A.R. Parkinson: *Development of an automatic, two-dimensional finite element mesh generator using quadrilateral elements and Bèzier curve* *boundary definition.* International Journal for Numerical Methods in Engineering, Vol. 29, 1990, 1551 - 1567.

[Várady/Pratt '84] T. Várady, M.J. Pratt: *Design techniques for the definition of solid objects with free-form geometry.* Computer Aided Geometric Design, Vol. 1, 1984, 207 - 225.