UNIVERSITY OF WEST BOHEMIA

Faculty of Applied Sciences

DOCTORAL THESIS

Plzeň, 2004 MSc. Alexandr Jeměljanov

UNIVERSITY OF WEST BOHEMIA

Faculty of Applied Sciences

DOCTORAL THESIS

in partial fulfillment of the requirement for
the degree of
Doctor of Philosophy
in specialization

Computer Science and Engineering

MSc. Alexandr Jeměljanov

Surface Reconstruction from Clouds of Points

Supervisor: Prof. Ing. Václav Skala, CSc.
Date of State doctoral exam: 25.06.2003
Date of Thesis consignation: 19.05.2004

V Plzni, 2004

ZÁPADOČESKÁ
UNIVERZITA
V PLZNI

Fakulta Aplikovaných Věd

DISERTAČNÍ PRÁCE

k získání akademického titulu doktor
v oboru

Informatika a Výpočetní Technika

MSc. Alexandr Jeměljanov

Surface Reconstruction from Clouds of Points

Školitel: Prof. Ing. Václav Skala, CSc.
Datum státní doktorské zkoušky: 25.06.2003
Datum odevzdání práce: 19.05.2004

V Plzni, 2004

Surface Reconstruction from Clouds of Points

Alexandr Jeměljanov

Abstrakt

Při rekonstrukci povrchu vznikají v praxi často dva následující problémy. První problém obyčejně vzniká v případech, kdy vstupní množina bodů je příliš velká, aby mohla být zpracována dokonce i na výkonných moderních počítačích. To je typické pro množiny bodů, který reprezentují velký sochy, domy, krajiny, kosmické objekty atd. Druhý problém vzniká v případě, kdy použitý algoritmus není schopen úplně zrekonstruovat CAD-model. Obvykle k tomu dochází při zpracovaní množin bodů, které byly získány skenováním objektů vně laboratoří. Oba výše popsané problémy mohou vznikat jak současně tak i odděleně. V této práci jsou navrženy metody, které tyto problémy řeší. Protože uvažované problémy mohou vzniknout při použití jakéhokoliv z existujících algoritmů, navržené metody nepředpokládají využití nějakého určitého algoritmu rekonstrukce povrchu. Řešení prvního problému je založeno na minimalizaci nákladů rekonstrukce povrchu. Práce obsahuje popis obecné koncepce a také popis dvou jednoduchých a rychlých algoritmů, které jsou určeny pro využití v rámci téhle koncepce. V obecném případě tyto algoritmy mohou být využity jako krok pre-processingu pro libovolný algoritmus rekonstrukce povrchu, který umožňuje zpracovávat množinu bodů spolu s danými hranicemi již částečně rekonstruovaného povrchu. V práci je dále navržena a popsána metoda pro opravu defektů CAD-modelu. Tato metoda pro rekonstrukci chybějícího povrchu využívá siločáry speciálně vytvořeného tenzorového pole a může být použita jako krok post-processingu pro libovolný algoritmus rekonstrukce povrchu. V této práci popsané metody se vzájemně doplňují a mohou být spolu využity jako samostatný systém pro rekonstrukci povrchu.

Táto páce byla podporována Ministerstvem Školství České Republiky – projekt MSM235200005 a Microsoft Research Ltd. – projekt 2003-178.

Západočeská univerzita v Plzni
Katedra informatiky a výpočetní techniky
Univerzitní 8
30614 Plzeň
Česká republika

Surface Reconstruction from Clouds of Points

Alexandr Jeměljanov

Abstract

In practice of surface reconstruction the following two problems often arise. The first one is that we often deal with very large clouds of points (for example, clouds representing buildings, big sculptures, landscape areas, etc.). Processing such clouds of points often leads to the problem of lack of machine resources even for modern powerful computers. In general this problem is a particular case of the problem of minimization of the cost of surface reconstruction. The second problem can be formulated as “What to do, if an algorithm is not capable to reconstruct a CAD-model completely?”. This problem usually arises when we deal with clouds of points obtained outside of a laboratory. The problems are found both: separately and together. In the given work methods for solution of these problems are proposed. Since these problems can be found in using any of surface reconstruction algorithms, the proposed solutions don’t suppose using a concrete algorithm of surface reconstruction. The solution, which is meant to reduce the cost of surface reconstruction, includes description of a general concept and description of two simple and fast algorithms, which are intended to be used within the framework of this concept. In general, these algorithms can be used as a pre-processing step for a surface reconstruction algorithm that can process a set of scattered points with given surface boundaries. The method offered to repair defects of a CAD-model can be considered as a post-processing step for any surface reconstruction algorithm. For surface reconstruction inside damaged regions this method uses force lines of a specially constructed tensor field. The described in the given work solutions can make a self-dependent surface reconstruction system as well.

This work was supported by the Ministry of Education of the Czech Republic – project MSM235200005 and by Microsoft Research Ltd. - project 2003-178

University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Univerzitni 8
30614 Pilsen
Czech Republic

Contents

1. INTRODUCTION

1.1 General motivations of the work

The problem of creating a CAD model for an existing physical object from a given set of points of the object surface is important in many fields of science and industry. There are many methods available for solution of this problem. These methods are based on a great variety of principles, and have various properties, that in many cases allows choosing the most suitable algorithm for a given task. At the same time many authors don't consider two problems, which often arise in practice. The first one is that we often deal with very large clouds of points (for example, clouds representing buildings, big sculptures, landscape areas, etc.). Processing such clouds of points often leads to the problem of lack of machine resources even for modern powerful computers. In general this problem is a particular case of the problem of minimization of the cost of surface reconstruction. The second problem can be formulated as "What to do, if an algorithm is not capable to reconstruct a CAD-model completely?" This problem usually arises when we deal with clouds of points obtained outside of a laboratory. The problems are found both: separately and together.

1.2 Used system of denotations

In the given work the following system of denotations is used. The six groups of denoted objects are defined: definitions, conditions, lemmas, mathematical expressions, figures, and tables. Each object is denoted by the index that is composed in the following way. As its basic part the index of corresponding enumerated block of text is used. Before the basic part there is a letter denoting the type of given object: "D" for definitions, "C" for conditions, "L" for lemmas, "E" for expressions, "F" for figures, and "T" for tables. At the end of basic part the ordinal number of the object is added. Enumeration is made from the beginning in each enumerated block of text. For each group of objects enumeration is made separately. If a figure or an expression has concern only to a given definition, condition, or lemma, then the index of such figure or expression is assembled by adding the letter "F" or "E" respectively before the index of given definition, condition, or lemma. For new introduced or locally defined terms the *italic* type of font is used.

1.3 Overview of the work

In most cases surface reconstruction implies that a given algorithm takes a cloud of sampled points as an input data and produces a CAD-model as a result. This scheme (let's denote it *S1*) is shown in figure F1.3-1.

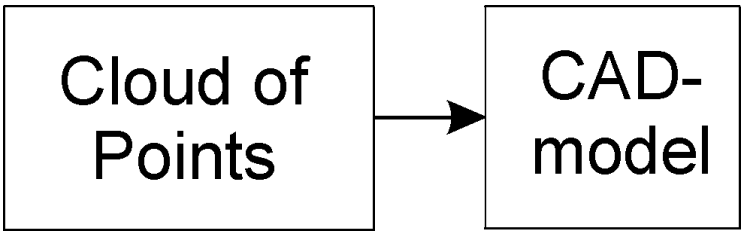


Figure F1.3-1

Definition D1.3-1. Let's consider an unsuccessful result of work of some surface reconstruction algorithm. Let we have a particularly reconstructed surface, and all regions of this surface (let's denote the aggregate of them A) are topologically equivalent to the corresponding regions of the original surface. Let's assume there are no invalid edges and triangles in the considered result. It means that each point of the input cloud of points is included in A or doesn't have any triangles and edges (let's call such point a *free point*). Let's call such result an *incomplete normalized CAD-model* (*ICADM*). Let's denote the aggregate of regions supplementing A up to a topologically correct CAD-model $\overline{A}$.

In the given work surface reconstruction is considered as a process of the two following steps: firstly from a given point cloud we obtain an ICADM and then from the ICADM we obtain a CAD-model. This scheme (let's denote it *S2*) is shown in figure F1.3-2.

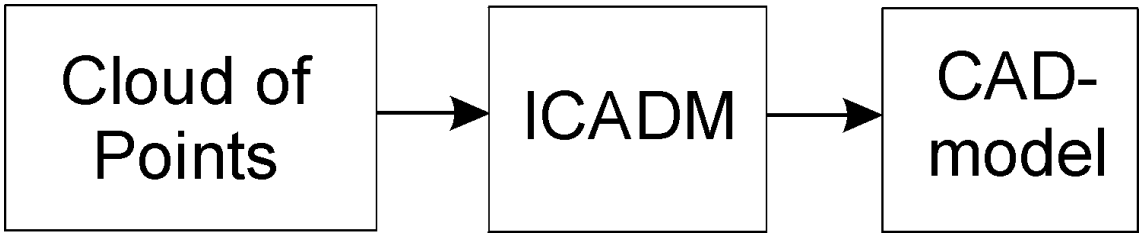


Figure F1.3-2

In chapter 3 at first on the base of scheme S2 definition of a general surface reconstruction concept is made. Then the case of application of scheme S1, when surface reconstruction from a given point cloud can be made by application of a given algorithm with the corresponding cost, is considered. It is shown, that using scheme S2, when the given algorithm is used only to realize step ICADM->CAD-model, the cost of the surface reconstruction can be reduced (with the assumption that the given algorithm can process a set of scattered points with given surface boundaries). In the given chapter two simple and fast algorithms to realize step PointCloud->ICADM are proposed without detailed considering step ICADM->CAD-model. The proposed solution can be used as a pre-processing step for any surface reconstruction algorithm that corresponds to the abovementioned assumption.

In chapter 4 the case, when application of scheme S2 is the only possible way, is considered. It occurs when a used surface reconstruction algorithm can't reconstruct a CAD-model completely. In this chapter a method to realize step ICADM->CAD-model is offered. For surface reconstruction inside damaged regions this method uses force lines of a specially constructed tensor field. The method can be used as a post-processing step for any surface reconstruction algorithm. Step PointCloud->ICADM isn't considered in the given chapter.

In general, the offered in the chapters 3 and 4 methods supplement each other, since they are intended to implement the corresponding steps of scheme S2. Because the same fact these methods (and the corresponding problems) can be considered independent from each other. Therefore each abovementioned chapter has its own section of results and conclusions. The total generalized conclusion concerning the work done and designation of the future work direction are made in chapter 5.

2. STATE OF THE ART

In this chapter a general description of basic groups of existing surface reconstruction methods is adduced. One of the most complete survey [MM98] is used for this purpose. All the information in this chapter is taken from the abovementioned survey and is not a result of my own work. A necessary discussion and criticism about some methods adduced in this chapter are made in corresponding parts of the next chapters.

2.1 Spatial subdivision

Common to the approaches that can be characterized by “Spatial Subdivision” is that some bounding box of the set *P* of sampling points is subdivided into disjoint cells. There is a variety of spatial decomposition techniques which were developed for different applications [LC87]. Typical examples are regular grids, adaptive schemes like octrees, or irregular schemes like tetrahedral meshes. Many of them can also be applied to surface construction. The goal of construction algorithms based on spatial subdivision is to find cells related to the shape described by *P*. The selection of the cells can be done in roughly two ways: surface-oriented and volume-oriented.

2.11 Surface-oriented cell selection

The surface-oriented approach consists of the following basic steps.

Surface-oriented cell selection:
1. Decompose the space in cells.
2. Find those cells that are traversed by the surface.
3. Calculate a surface from the selected cells.

2.111 The approach of Algorri and Schmitt

An example for surface-oriented cell selection is the algorithm of Algorri and Schmitt [AS96] (figure F2.111-1). For the first step, the rectangular bounding box of the given data set is subdivided by a regular voxel grid. In the second step, the algorithm extracts those voxels which are occupied by at least one point of the sampling set *P*. In the third step, the outer quadrilaterals of the selected voxels are taken as a first approximation of the surface. This resembles the cuberille approach of volume visualization [HL79].

In order to get a more pleasent representation, the surface is transferred into a triangular mesh by diagonally splitting each quadrilateral into two triangles. The cuberille artifacts are smoothed using a depth-pass filter that assigns a new

position to each vertex of a triangle. This position is computed as the weighted average of its old position and the position of its neighbors. The approximation of the resulting surface is improved by warping it towards the data points. For more on that we refer to section 2.32.

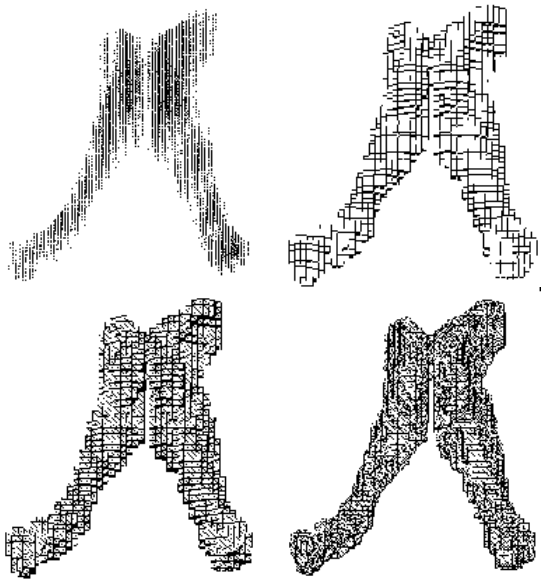


Figure F2.111-1

2.112 The approach of Hoppe et al.

Another possibility of surface-oriented cell selection is based on the distance function approach of Hoppe [HDD92, HDD93, HH94] (figure F2.112-1). The distance function of the surface of a closed object tells for each point in space its minimum signed distance to the surface. Points on the surface of course have distance 0, whereas points outside the surface have positive, and points inside the surface have negative distance. The calculation of the distance function is outlined in section 2.21.

The first step of the algorithm again is implemented by a regular voxel grid. The voxel cells selected in the second step are those which have vertices of opposite sign. Evidently, the surface has to traverse these cells. In the third step, the surface is obtained by the marching cubes algorithm of volume visualization [LC87]. The marching cubes algorithm defines templates of separating surface patches for each possible configuration of the signs of the distance values at the vertices of a voxel cell. The voxels are replaced with these triangulated patches. The resulting triangular mesh separates the positive and negative distance values on the grid.

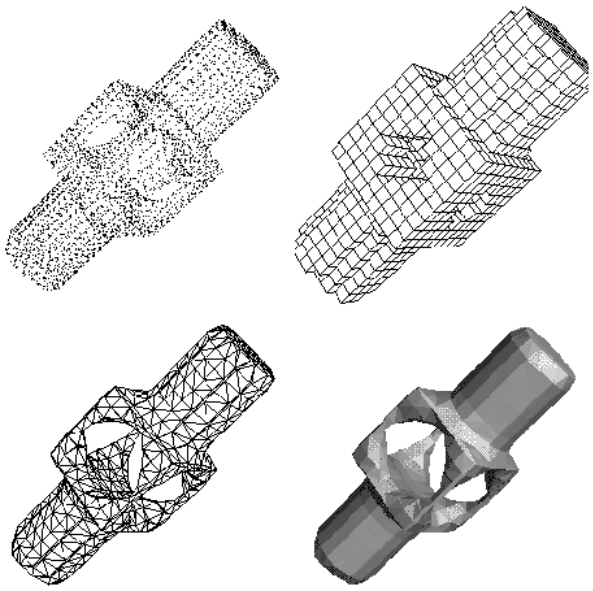


Figure F2.112-1

A similar algorithm was suggested by Roth and Wibowoo [RGW97]. It differs from the approach of Hoppe et al. in the way the distance function is calculated, cf. section 2.21. Furthermore, the special cases of profile lines and multiple view range data are considered besides scattered point data.

A difficulty with these approaches is the choice of the resolution of the voxel grid. One effect is that gaps may occur in the surface because of troubles of the heuristics of distance function calculation.

2.113 The approach of Bajaj, Bernardini et al.

The approach of Bajaj, Bernardini et al. [BBX95] differs from the previous ones in that spatial decomposition is now irregular and adaptive.

The algorithm also requires a signed distance function. For this purpose, a first approximate surface is calculated in a preprocessing phase. The distance to this surface is used as distance function. The approximate surface is calculated using α-solids which will be explained in section 2.12.

Having the distance function in hand, the space is incrementally decomposed into tetrahedra starting with an initial tetrahedron surrounding the whole data set. By inspecting the signs of the distance function at the vertices, the tetrahedra traversed by the surface are found out. For each of them, an approximation of the traversing surface is calculated. For this purpose, a Bernstein-Bezier trivariate implicit approximant is used. The approximation error to the given data points is calculated. A bad approximation induces a further refinement of the tetrahedrization. The refinement is performed by incrementally inserting the centers of tetrahedra with high approximation error into the

tetrahedrization. The process is iterated until a sufficient approximation is achieved.

In order to keep the shape of the tetrahedra balanced, an incremental tetrahedrization algorithm is used so that the resulting tetrahedrizations always have the Delaunay property. A tetrahedrization is said to have the *Delaunay property* if none of its vertices lies inside the circumscribed sphere of a tetrahedron [PS85].

The resulting surface is composed of trivariate implicit Bernstein-Bezier patches. A C^1 smoothing of the constructed surfaces is obtained by applying a Clough-Tocher subdivision scheme.

In Bernardini et al. [BBC97] an extension and modification of this algorithm is formulated [BBX97, BB97]. The algorithm consists of an additional mesh simplification step to reduce the complexity of the mesh represented by the α-solid [BS96]. The reduced mesh is used in the last step of the algorithm for polynomial-patch data fitting using Bernstein-Bezier patches for each triangle (by interpolating the vertices and normals and by approximating data points in its neighborhood). Additionally, the representation of sharp features can be achieved in the resulting surface.

2.114 Edelsbrunner's and Mucke's alpha-shapes

Edelsbrunner and Mucke [EPM92, EPM93, EH92] also use an irregular spatial decomposition. In contrast to the previous ones, the given sample points are part of the subdivision. The decomposition chosen for that purpose is the Delaunay tetrahedrization of the given set *P* of sampling points. A tetrahedrization of a set *P* of spatial points is a decomposition of the convex hull of *P* into tetrahedra so that all vertices of the tetrahedra are points in *P*. A tetrahedrization is a *Delaunay tetrahedrization* if none of the points of *P* lies inside the circumsphere of a tetrahedron. It is well known that each finite point set has a Delaunay tetrahedrization which can be calculated efficiently [PS85]. This is the first step of the algorithm.

The second step is to erase tetrahedrons, triangles, and edges of the Delaunay tetrahedrization using so-called α*-balls* as eraser sphere with radius α. Each tetrahedron, triangle, or edge of the tetrahedrization whose corresponding minimum surrounding sphere does not fit into the eraser sphere is eliminated. The resulting so-called α*-shape* is a collection of points, edges, faces, and tetrahedra.

In the third step, the triangles are extracted out of the α-shape which belong to the desired surface, using the following rule. Consider the two possible spheres of radius α through all three points of a triangle of the α-shape. If at least one of these does not contain any other point of the point set, the triangle belongs to the surface.

A problem of this approach is the choice of a suitable α. Since α is a global parameter the user is not swamped with many open parameters, but the drawback is that a variable point density is not possible without loss of detail in the reconstruction.

An example for a reconstruction of a body is shown in figure F2.114-1. If α is too small, gaps in the surface can occur, or the surface may become fragmented.

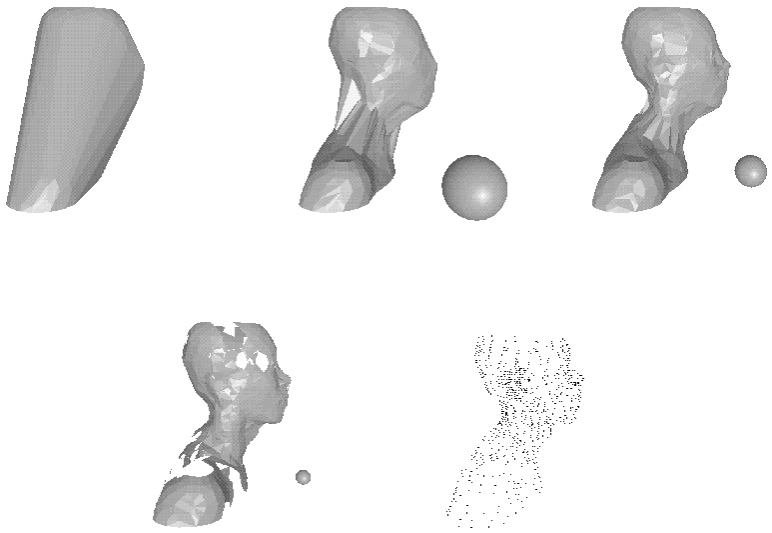


Figure F2.114-1

Guo et al. [GMW97] also make use of α-shapes for surface reconstruction but they propose a so-called *visibility* algorithm for extracting those triangles out of the α-shape which represent the simplicial surface.

2.115 Attali's normalized meshes

In the approach of Attali [AD97], the Delaunay tetrahedrization is also used as a basic spatial decomposition. Attali introduces so-called *normalized meshes* which are contained in the Delaunay graph. It is formed by the edges, faces and tetrahedra whose dual Voronoi element intersects the surface of the object.

In two dimensions, the normalized mesh of a curve *c* consists of all edges between pairs of points of the given set *P* of sampling points on *c* which induce an edge of the Voronoi diagram of *P* that intersects *c*. The nice property of normalized meshes is that for a wide class of curves of bounded curvature, the so-called *r-regular* shapes, a bound on the sampling density can be given within which the normalized mesh retains all the topological properties of the original curve.

For reconstruction of *c*, the edges belonging to the reconstructed mesh are obtained by considering the angle between the intersections of the two possible circles around a Delaunay edge. The angle between the circles is defined to be the smaller of the two angles between the two tangent planes at one intersection point of the two circles. This characterization is useful because Delaunay discs tend to become tangent to the boundary of the object. The reconstructed mesh consists of all edges whose associated Delaunay discs have an angle smaller than $\frac{\pi}{2}$. If the sampling density is sufficiently high, the reconstructed mesh is equal to the normalized mesh.

While in two dimensions the normalized mesh is a correct reconstruction of shapes having the property of r-regularity, the immediate extension to three dimensions is not possible. The reason for that is that some Delaunay spheres can intersect the surface without being approximately tangent to it. Therefore, the normalized mesh in three dimensions does not contain all faces of the surface.

To overcome this problem, two different heuristics for filling the gaps in the surface structure were introduced.

The first heuristic is to triangulate the border of a gap in the triangular mesh by considering only triangles contained in the Delaunay tetrahedrization.

The second heuristic is volume-based. It merges Delaunay tetrahedra to build up the possibly different solids represented in the point set. The set of mergeable solids is initialized with the Delaunay tetrahedra and the complement of the convex hull. The merging step is performed by processing the Delaunay triangles according to decreasing diameters. If the current triangle separates two different solids in the set of mergable solids, they are merged if the following holds:

- no triangle from the normalized mesh disappears;
- merging will not isolate sample points inside the union of these objects, i.e. the sample points have to remain on the boundary of at least one object.

The surface finally yielded by the algorithm is formed by the boundary of the resulting solids.

2.116 Weller's approach of stable Voronoi edges

Let *P* be a finite set of points in the plane. *P'* is an *ε-perturbation* of *P* if $d(p_i p_i^{'}) \le \varepsilon$ holds for all $p_i \in P$, $p_i^{'} \in P'$, $i = 1,...,n$. An edge $p_i^{'}, p_j^{'}$ of the Delaunay triangulation is called *stable* if the perturbed endpoints $p_i^{'}, p_j^{'}$ are also connected by an edge of the Delaunay triangulation of the perturbed point set *P'*.

It turns out that for intuitively reasonably sampled curves in the plane, the stable edges usually are the edges connecting two consecutive sampling points on the curve, whereas the edges connecting non-neighboring sampling points are instable. The stability of an edge can be checked in time *O*(Voronoi neighbors) [WF97].

The extension of this approach to 3D-surfaces shows that large areas of a surface can usually be reconstructed correctly, but still not sufficiently approximated regions do exist. This resembles the experience reported by Attali [AD97], cf. section 2.11. Further research is necessary in order to make stability useful for surface construction.

2.12 Volume-oriented cell selection

Volume-oriented cell selection also consists of three steps which at a first glance are quite similar to those of surface-oriented selection:

Volume-oriented cell selection:
1. Decompose the space in cells.
2. Remove those cells that do not belong to the volume bounded by the sampled surface.
3. Calculate a surface from the selected cells.

The difference is that a volume representation, in contrast to a surface representation, is obtained.

Most implementations of volume-oriented cell selection are based on the Delaunay tetrahedrization of the given set *P* of sampling points. The algorithms presented in the following differ in how volume-based selection is performed. Some algorithms eliminate tetrahedrons expected outside the desired solid, until a description of the solid is achieved [BJ84, IBS97, VR94]. Another methodology is the use of the Voronoi diagram to describe the constructed solid by a “skeleton” [SB97, AD97].

2.121 Boissonnat's volume-oriented approach

Boissonnat's volume-oriented approach starts with the Delaunay triangulation of the given set *P* of sampling points. From this triangulation of the convex hull, tetrahedra having particular properties are successively removed. First of all, only tetrahedra with *two faces, five edges and four points* or *one face, three edges and three points* on the boundary of the current polyhedron are eliminated. Because of this elimination rule only objects without holes can be reconstructed, cf. figure F2.121-1.

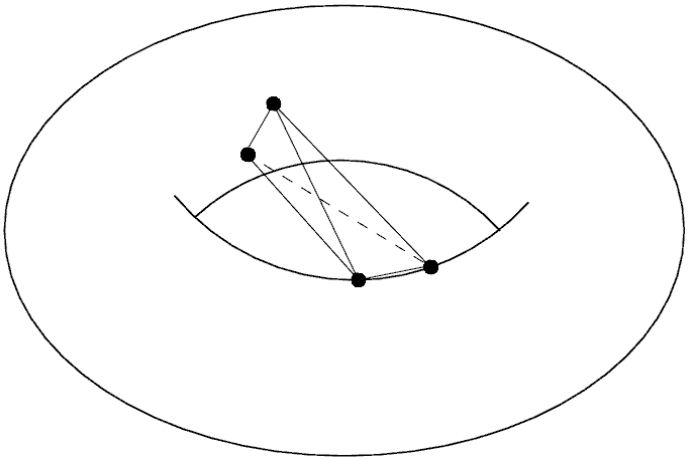


Figure F2.121-1

Tetrahedra of this type are iteratively removed according to decreasing *decision values*. The decision value is the maximum distance of a face of the tetrahedron to its circumsphere. This decision value is useful because flat tetrahedra of the Delaunay tetrahedrization usually tend to be outside the object and cover areas of higher detail. The algorithm stops if all points lie on the surface, or if the deletion of the tetrahedron with highest decision value does not improve the sum taken over the decision values of all tetrahedra incident to the boundary of the polyhedron.

2.122 The approach of Isselhard, Brunnett, and Schreiber

The approach of [IBS97] is an improvement of the volume-oriented algorithm of Boissonnat [BJ84]. While Boissonnat cannot handle objects with holes, the deletion procedure of this approach is modified so that construction of holes becomes possible.

As before, the algorithm starts with the Delaunay triangulation of the point set. An incremental tetrahedron deletion procedure is then performed on tetrahedra at the boundary of the polyhedron, as in Boissonnat's algorithm. The difference is, that more types of tetrahedron can be removed in order to being able to reconstruct even object with holes. The additionally allowed types of tetrahedra are those with *one face and four vertices* or *three faces* or *all four faces* or on the current surface provided that no point would become isolated through their elimination.

The elimination process is controlled by observing an *elimination function*. The elimination function is defined as the maximum decision value (in the sense of Boissonnat) of the remaining removable tetrahedra. In this function, several significant jumps can be noticed. One of these jumps is expected to indicate that the desired shape is reached. In practice, the jump before the stabilization of the function on a higher level is the one which is taken. This stopping point

helps handling different point densities in the point set which would lead to undesired holes through the extended type set of removable tetrahedra in comparison to Boissonnat's algorithm [BJ84].

If all data points are already on the surface, the algorithm stops. If not, more tetrahedra are eliminated to recover sharp edges (reflex edges) of the object. For that purpose the elimination rules are restricted to those of Boissonnat, assuming that all holes present in the data set have been recovered at this stage. Additionally, the decision value of the tetrahedra is scaled by the radius of the circumscribed sphere as a measure for the size of the tetrahedron. In this way, the cost of small tetrahedra is increased which are more likely to be in regions of reflex edges than big ones. The elimination continues until all data points are on the surface and the elimination function does not decrease anymore.

An example point set and the deletion process is depicted in figure F2.122-1.

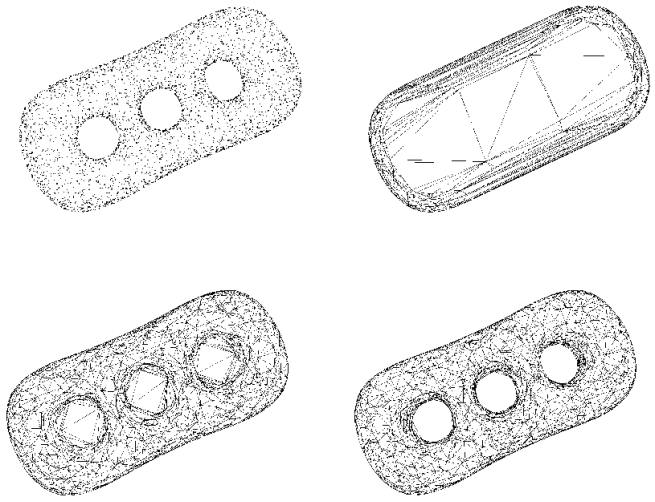


Figure F2.122-1

2.123 The γ-indicator approach of Veltkamp

To describe the method of Veltkamp [VR94, VR95] some terminology is required. A *γ-indicator* is a value associated to a sphere through three boundary points of a polyhedron which is positive or negative, cf. figure F2.123-1 for an illustration of the 2D-case. Its absolute value is computed as $1-\frac{r}{R}$, where *r* is the circle for the boundary triangle and *R* the radius of the boundary tetrahedron. It is taken to be negative if the center of the sphere is on the inner side and positive if the center is on the outer side of the polyhedron. Note, that the γ-indicator is independent of the size of the boundary triangle (tetrahedron,

respectively). Therefore, it adapts to areas of changing point density. A removable face is a face with positive γ-indicator value.

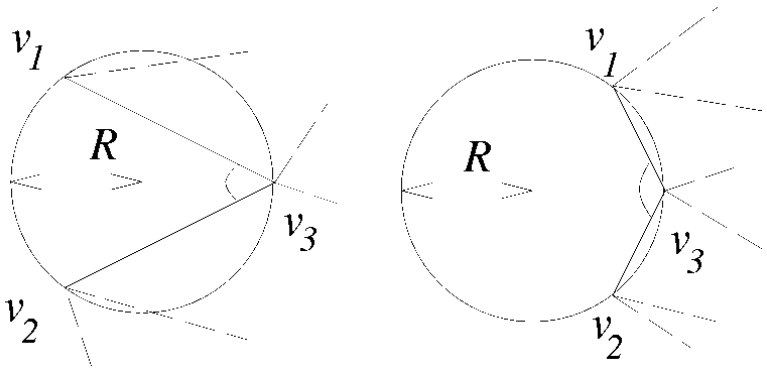


Figure F2.123-1

The first step of the algorithm is to calculate the Delaunay tetrahedrization.

In the second step, a heap is filled with removable tetrahedra which are sorted according to their γ-indicator value. The removable tetrahedra are of the same boundary types as in Boissonnat's volume-oriented approach [BJ84]. The tetrahedron with the largest γ-indicator value is removed and the boundary is updated. This process continues until all points lie on the boundary, or there are no further removable tetrahedra.

The main advantage of this algorithm is the adaption of the γ-indicator value to variable point density. Like Boissonnat's approach, the algorithm is restricted to objects without holes.

Some intermediate stages during the construction of a surface are displayed in figure F2.123-2.

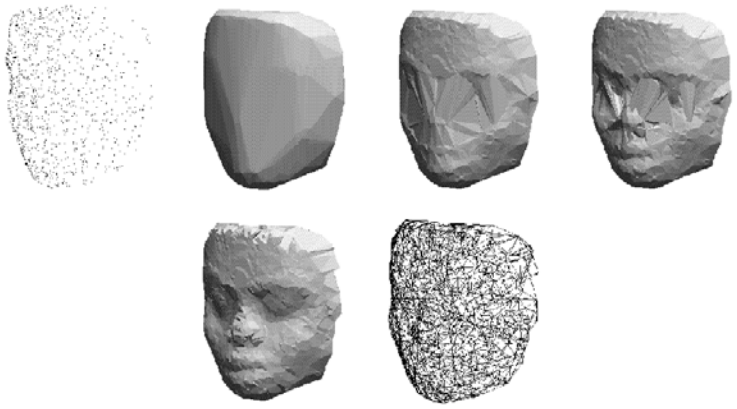


Figure F2.123-2

2.124 The approach of Schreiber and Brunnett

The approach of Schreiber and Brunnett [ST97, SB97] uses properties of the Voronoi diagram of the given point set for tetrahedra removal. The *Voronoi diagram* of a point set *P* is a partition of the space in regions of nearest neighborhood. For each point *p* in *P*, it contains the region of all points in space that are closer to *p* than to any other point of *P*. It is interesting to note that the Voronoi diagram is dual to the Delaunay tetrahedrization of *P*. For example, each vertex of the Voronoi diagram corresponds to the center of a tetrahedron of the tetrahedrization. Edges of the Voronoi diagram correspond to neighboring faces of the tetrahedra dual to its vertices. The same observation holds for Voronoi diagrams in the plane that are used in the following for the explanation of the 2D-version of the algorithm.

In the first step, the Delaunay triangulation and the dual Voronoi diagram of *P* is determined. The second step, the selection of tetrahedra, uses a minimum spanning tree of the Voronoi graph, cf. figure F2.124-1. The *Voronoi graph* is the graph induced by the vertices and edges of the Voronoi diagram. A *minimum spanning tree* (*MST*) of a graph is a subtree of the graph which connects all vertices and has minimum summed edge length. Edge length in our case is the Euclidean distance of the two endpoints of the edge.

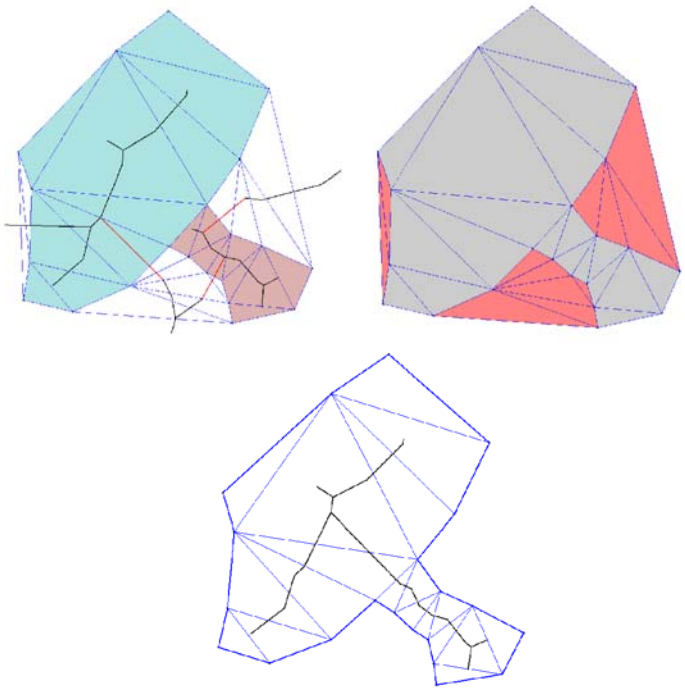


Figure F2.124-1

In the second step, a pruning strategy is applied to it which possibly decomposes it into several disjoint subtrees. Each subtree represents a region defined by the union of the triangles dual to its vertices.

Two pruning rules have been developed for that purpose:

1. All those edges will be pruned for which no end point is contained in the circumcircle of the dual Delaunay triangle of the other end point.
2. An edge will be pruned if its length is shorter than the mean value of the radii of both circumcircles of the dual Delaunay triangles of its voronoi end points.

The number of edges to be pruned can be controlled by using the edge length as a parameter.

The resulting regions are then distinguished into inside and outside. In order to find the inside regions, we add the complement of the convex hull as further region to the set of subtree regions. The algorithm starts with a point on the convex hull which is incident to exactly two regions. The region different from the complement of the convex hull is classified “inside”. Then the label “inside” is propagated to neighboring regions by again considering points that are incident to exactly two regions.

After all regions have been classified correctly, the boundary of the constructed shape is obtained as the boundary of the union of the region labeled “inside”.

An adaption of this method to three dimensions is possible.

2.125 The α-solids of Bajaj, Bernardini et al.

Bajaj, Bernardini et al. [BBX95, BBX97, BB97, BBC97] calculate so-called *α-solids*. While α-shapes are computed by using eraser spheres at every point in space, the eraser spheres are now applied from outside the convex hull, like in Boissonnat's approach [BJ84]. To overcome the approximation problems inherent to α-shapes a re-sculpturing scheme has been developed. Re-sculpturing roughly follows the volumetric approach of Boissonnat in that further tetrahedra are removed. This goal is to generate finer structures of the object provided the α-shape approach has correctly recognized the larger structures of the object.

2.2 Surface construction with distance functions

The distance function of a surface gives the shortest distance of any point in space to the surface. For closed surface the distances can be negative or positive, dependent on whether a point lies inside or outside of the volume bounded by the surface. In section 2.1 we have already described an algorithm which uses the distance function for the purpose of surface construction. There the question remained open how a distance function can be calculated from the given set *P* of sample points. Solutions are presented in the next subsection.

Another possibility of calculating a distance function is to construct a surface to the given set P of data points and take the distance to this surface. The idea behind that is that this distance function may be used to get a better surface, for instance a smooth surface as in [BBX95].

Besides marching cubes construction of surfaces as explained in section 2.11, distance plays a major role in construction of surfaces using the medial axis of a volume. The medial axis consists of all points inside the volume for which the maximal sphere inside the volume and centered at this point does not contain the maximal sphere of any other point. Having the medial axis and the radius of the maximum sphere at each of its points, the given object can be represented by the union taken over all spheres centered at the skeleton points with the respective radius. An algorithm for surface construction based on medial axes is described in section 2.22.

2.21 Calculation of distance functions

2.211 The approach of Hoppe et al.

Hoppe et al. [HDD92, HH94] suggest the following approach. At the beginning, for each point p_i an estimated tangent plane is computed. The tangent plane is obtained by fitting the best approximating plane in the least square sense [DRH73] into a certain number k of points in the neighborhood of p_i. In order to get the sign of the distance in the case of close surfaces, a consistent orientation of neighboring tangent planes is determined by computing the *Riemannian graph* (figure F2.211-1). The vertices of the Riemannian graph are the centers of the tangent planes which are defined as the centroids of the k points used to calculate the tangent plane. Two tangent plane centers o_i, o_j are connected with an edge (i, j) if one center is in the k-neighborhood of the other center. By this construction, the edges of the Riemannian graph can be expected to lie close to the sampled surface.

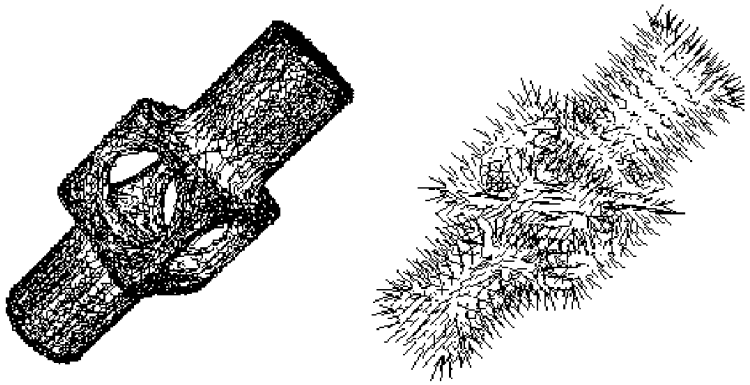


Figure F2.211-1

Each edge is weighted by *1* minus the absolute value of the scalar product between normals of the two tangent plane centers defining the edge. The orientation of the tangent planes is determined by propagating the orientation at a starting point, by traversing the minimum spanning tree of the resulting weighted Riemannian graph.

Using the tangent plane description of the surface and their correct orientations, the signed distance is computed by first determining the tangent plane center nearest to the query point. The distance between the query point and its projection on the nearest tangent plane. The sign is obtained form the orientation of the tangent plane.

2.212 The approach of Roth and Wibowoo to distance functions

The goal of the algorithm of Roth and Wibowoo [RGW97] is to calculate distance values at the vertices of a given voxel grid surrounding the data points. The data points are assigned to the voxel cells into whcih they fall. An “outer” normal vector is calculated for each data point by finding the closest two neighboring points in the voxel grid, and then using these points along with the original point to compute the normal.

The normal orientation which is required for signed distance calculation is determined as follows. Consider the voxel grid and the six axis directions ($\pm x$, $\pm y$, $\pm z$). If we look from infinity down each axis into the voxel grid, then those voxels that are visible must have their normals point towards the viewing direction. The normal direction is fixed for these visible points. Then the normal direction is propagated to those neighboring voxels whose normals are not fixed by this procedure. This heuristic only works if the nonempty voxel defines a closed boundary without holes.

The value of the signed distance function at a vertex of the voxel grid is computed by taking the weighted average of the signed distances of every point in the eight neighboring voxels. The signed distance to a point with normal is the Euclidean distance to this point, with positive sign if the angle between the normal and the vector towards the voxel vertex exceeds 90°.

2.22 Bittar's et al. surface construction by medial axes

The approach of Bittar et al. [BTG95] consists of two steps, the calculation of the medial axis and the calculation of an implicit surface from the medial axis.

The medial axis is calculated from a voxelization of a bounding box of the given set of points. The voxels containing points of the given point set *P* are assumed to be boundary voxels of the solid to be constructed. Starting at the boundary of the bounding box, voxels are successively eliminated until all boundary voxels are on the surface of the remaining voxel volume. A distance

function is propagated from the boundary voxels to the inner voxels of the volume, starting wiht distance 0 on the boundary voxels. The voxels with locally maximal distance value are included to the medial axis.

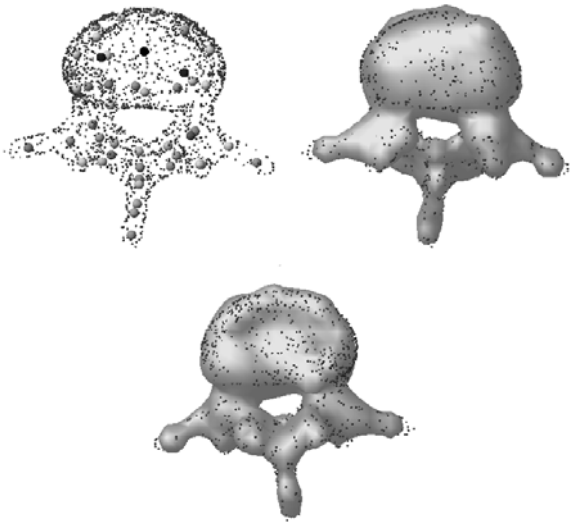


Figure F2.22-1

The desired surface is calculated by distributing centers of spheres on the medial, cf. figure F2.22-1. The radius of a sphere is equal to the distance assigned to its center on the medial axis. For each sphere, a field function is defined which allows to calculate a scalar field value for arbitrary point in space. A field function of the whole set of spheres is obtained as sum of the field functions of all spheres. The implicit surface is defined as an iso-surface of the field function, that is, it consists off all points in space for which the field function has a given constant value.

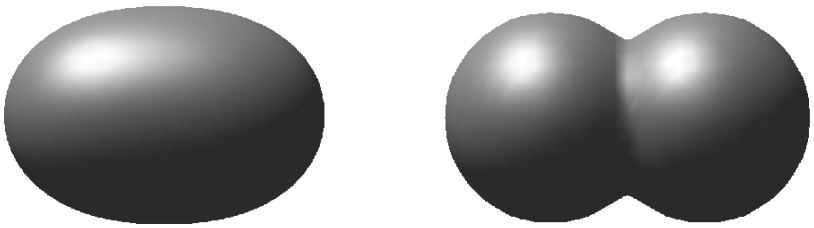


Figure F2.22-2

In order to save computation time, a search strategy is introduced which restricts the calculation of the sum to points with suitable positions.

The shape of the resulting surface is strongly influenced by the type of field function. For example, a *sharp* field function preserves details while a *soft*

function smoothes out the details, cf. figure F2.22-2. Also the connectness of the resulting solid can be influenced by the shape function cf. figure F2.22-3.

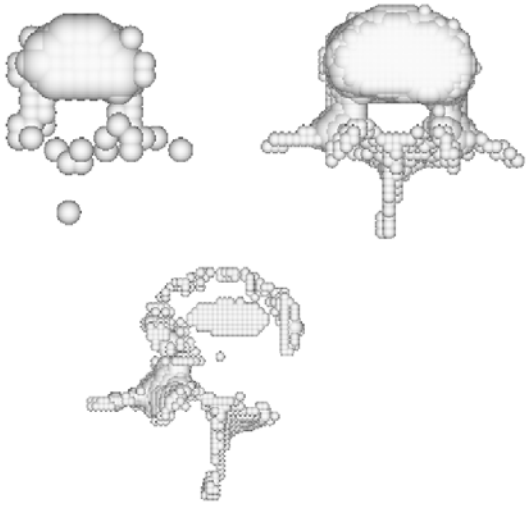


Figure F2.22-3

Because of the voxelization, a crucial point is tuning the resolution of the medial axis. If the resolution of the axis is low, finer details are not represented very accurately. The display of the surface detail is improved if the resolution is increased but can also tend to disconnect parts of the surface if the resolution is higher than the sample density at certain regions.

A result of this algorithm is shown in figure F2.22-1.

2.3 Surface construction by warping

Warping-based surface construction means to deform an initial surface so that it gives a good approximation of the given point set *P*. For example, let the initial shape be a triangular surface to some or all of its vertices corresponding points in *P* are determined to which the vertices have to be moved in the warping process. When moving the vertices of the mesh to their new locations, the rest of the mesh is also deformed and yields a surface approximation of the points in *P*.

Surface construction by warping is particularly suited if a rough approximation of the desired shape is already known. This simplifies detection of corresponding points.

Several methods of describing deformable surfaces were developed in the past. Muraki suggested a "*blobby model*" to approximate 2.5 D range images [MS91]. Terzopoulos, Witkin and Kass [TM91, TWK88] made use of *deformable superquadrics* which have to fit the input data points.

Miller et al. [MBL91] extract a topologically closed geometric model from a volume data set. The algorithm starts with a simple model that is already topologically closed and deforms the model on a set of constraints, so that the model grows or shrinks to fit the object within the volume while maintaining it closed and a locally simple non-self-intersecting polyhedron that is either embedded in the object or surrounds the object in the volume data representation. A function is associated with every vertex of the polyhedron that associates costs with local deformation adherent to properties of simple polyhedra, and the relationship between noise and feature. By minimizing these constraints, one achieves an effect similar to inflating a balloon within a container or collapsing a piece of shrink wrap around the object.

A completely different approach to warping is modeling with *oriented particles* suggested by Szeliski and Tonnesen [SRT92]. Each particle owns several parameters which are updated during the modeling simulation. By modeling the interaction between the particles themselves the surface is being modeled using forces and repulsion. As an extension Szeliski and Tonnesen describe how their algorithm can be extended for automatic 3D reconstruction. At each sample location one particle with appropriate parameters is generated. The gaps between the sample points (particles, respectively) are filled by growing particles away from isolated points and edges. After having a rough approximation of the current surface the other particles are rejusted to smooth the surface.

In the following three subsections three approaches are outlined which stand for basically different methodologies, a purely geometric approach, a physical approach, and a computational intelligence approach.

2.31 Spatial free form warping

The idea of spatial free-form warping is to deform the whole space in which an object to be warped is embedded in, with the effect that the object is warped at the same time. Space deformation is defined by a finite set of displacement vectors consisting of pairs of initial and target point, from which a spatial displacement vector field is interpolated using a scattered data interpolation method. There is a huge number of scattered data interpolation methods known in literature, cf. e.g. [HJL93]. Among them that one can be chosen that yields the most reasonable shape for the particular field of application.

The resulting displacement vector field tells for each point in space its target point. In particular, if the displacement vector field is applied to all vertices of the initial mesh, or of a possibly refined one, the mesh is warped towards the given data points [RM95].

The advantage of spatial free form warping is that usually only a small number of control displacement vectors located at points with particular features like corners or edges is necessary. A still open question is how to find good control displacement vectors automatically.

2.32 The approach of Algorri and Schmitt

The idea of Algorri and Schmitt [AS96] is to translate given approximate triangular mesh into a physical model, cf. figure F2.32-1. The vertices of the mesh are interpreted as mass points. The edges are replaced with springs. Each nodal mass of the resulting mesh of springs is attached to its closest point in given set P of sampling points by a further spring. The masses and springs are chosen so that the triangular mesh is deformed towards the data points.

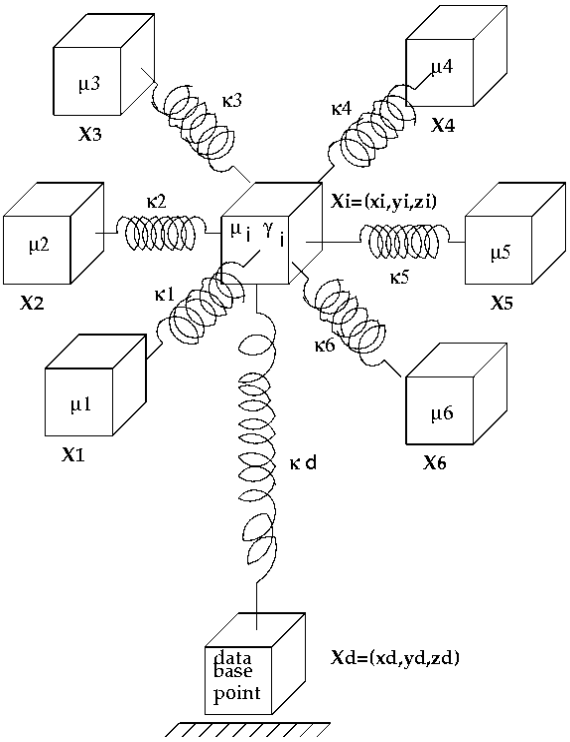


Figure F2.32-1

The model can be expressed as a linear differential equation of degree 2. This equation is solved iteratively. Efficiency is gained by embedding the data points and the approximate triangular mesh into a regular grid of voxels, like that one already yielded by the surface construction algorithm of the same authors, cf. section 2.11.

2.33 Kohonen feature map approach of Baader and Hirzinger

The Kohonen feature map approach of Baader and Hirzinger [BH93, BH94] can be seen as another implementation of the idea of surface construction by warping. Kohonen's feature map is a two-dimensional array of units (neurons), cf. figure F2.33-1.

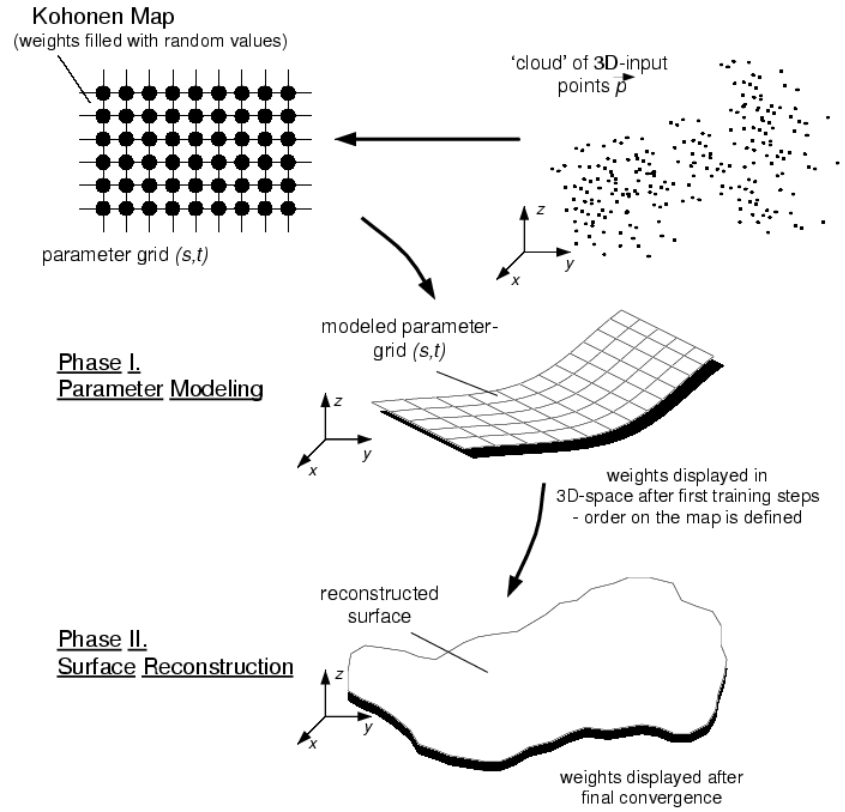


Figure F2.33-1

Each unit u_j has a corresponding weight vector $\overrightarrow{w_j}$. In the beginning these vectors are set to normalized random values (of length equal to 1). During the reconstruction or training process the neurons are fed with the input data which affects their weight vectors (which resemble their position in three-space). Each input vector $\vec{i}$ is presented to the units j which produce output o_j of the form $o_j = \overrightarrow{w_j}\vec{i}$. The unit generating the highest response o_j is the center of the excitation area. The weights of this unit and a defined neighborhood are updated by the formula $\overrightarrow{w_j}(t+1) = \overrightarrow{w_j}(t) + \varepsilon_i(\vec{i} - \overrightarrow{w_j}(t))$

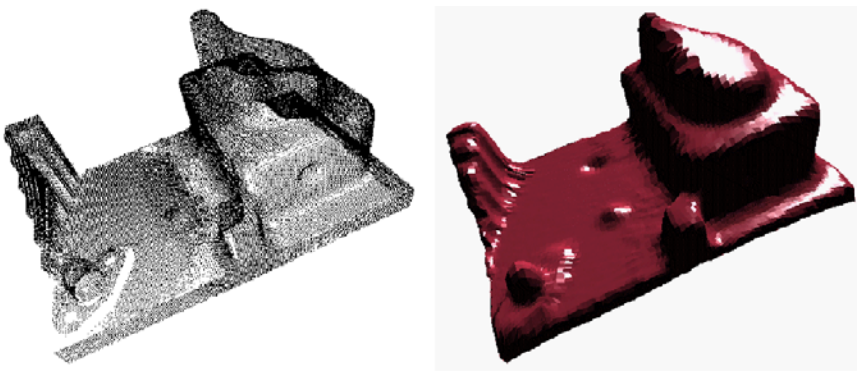


Figure F2.33-2

Note that after this update the weight vectors have to be normalized again. The value $\varepsilon_j = \eta h_j$ contains two values, the learning rate η and the neighborhood relationship h_j. Units far away from the center of excitation are only slightly changed.

The algorithm has one additional difficulty. If the input point data do not properly correspond with the neuron network it is possible, that neurons might remain which had not been in any center of excitation so far. Therefore they had been updated only by the neighborhood update which usually is not sufficient to place the units near the real surface. Having this in mind, Baader and Hirzinger have introduced a kind of *reverse training*. Unlike the *normal training* where for each input point a corresponding neural unit is determined and updated the procedure in the intermediate reverse training is reciprocal. For each unit u_j the part of the input data with the highest influence is determined and used for updating u_j.

The combination of this normal and reverse training completes the algorithm of Baader and Hirzinger and has to be used in the training of the network.

A result is depicted in figure F2.33-2.

2.4 Incremental surface-oriented construction

The idea of incremental surface-oriented construction is to build-up the interpolating or approximating surface directly on surface-oriented properties of the given data points. This can be done in quite different manner.

For example, surface construction may start with an initial surface edge at some location of the given point set *P*, connecting two of its points which are expected neighboring on the surface. The edge is successively extended to a larger surface by iteratively attaching further triangles at boundary edges of the emerging surface. The surface-oriented algorithm of Boissonnat explained in the first subsection may be assigned to this category.

Another possibility is to start with a global wire frame of the surface, in order to fill it iteratively to a complete surface. This is the idea of the approach of Mencl and Muller described in section 2.42.

2.41 Boissonat's surface-oriented approach

Boissonnat's surface oriented contouring algorithm [BJ84] usually starts at the shortest connection between two points of the given point set *P*. In order to attach a new triangle at this edge, and later on to other edges on the boundary, a locally estimated tangent plane is computed based on the points in the neighborhood of the boundary edge. The points in the neighbourhood of the boundary edge are then projected onto the tangent plane. The new triangle is

obtained by connecting one of these points to the boundary edge. That point is taken which maximizes the angle between at its edges in the new triangle, that is, the point sees edge boundary edge under the maximum angle, cf. figure F2.41-1.

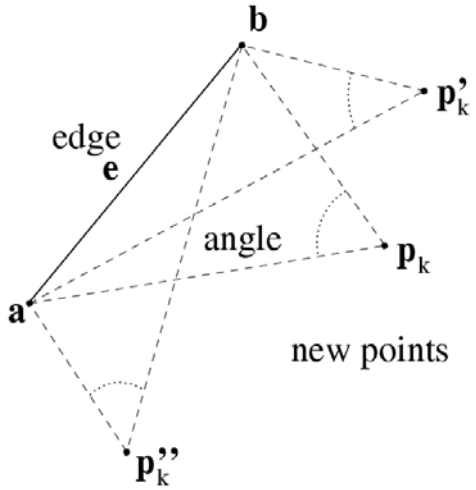


Figure F2.41-1

The algorithm terminates if there is no free edge available any more. The behavior of this algorithm can be seen in figure F2.41-2.

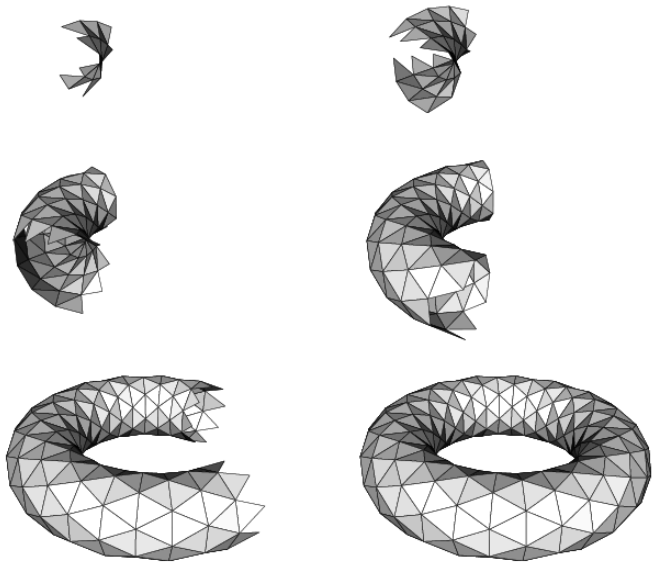


Figure F2.41-2

2.42 Approach of Mencl and Muller

The solution of Mencl and Muller consists of seven main steps [RM95, MM97]:

1. The computation of the *EMST* (*Euclidean minimum spanning tree*) of the point set.
2. Extension of the graph at leaf points of the EMST.
3. Recognition of features.
4. Extraction of different objects out of the graph.
5. Connection of features of the same kind.
6. Connection of associated edges in the graph.
7. Filling the wire frame with triangles.

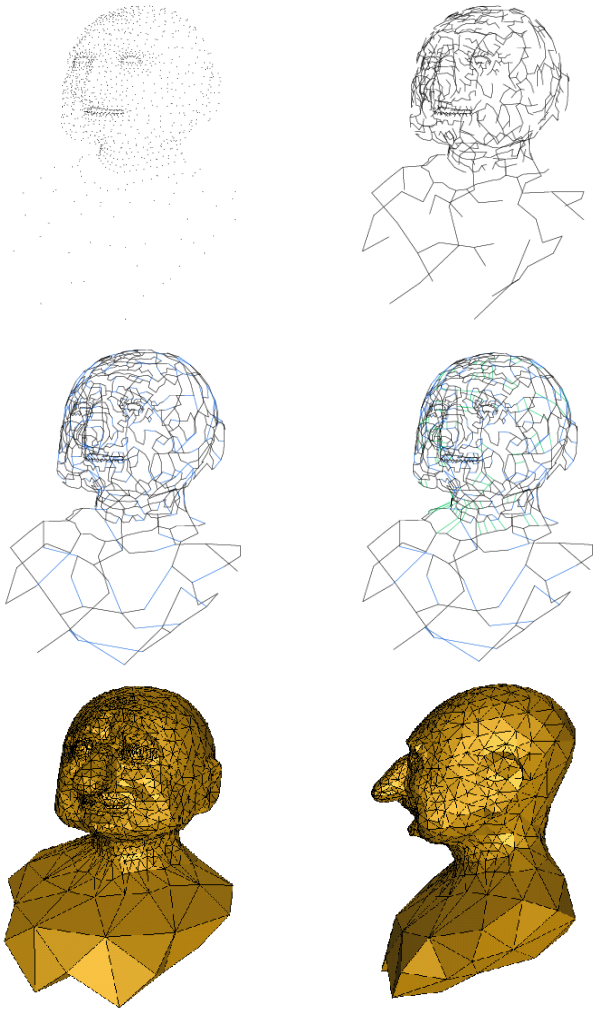


Figure F2.42-1

The first two steps are designed to build up an initial *surface description graph* (*SDG*). This is performed by computing the EMST (Euclidean minimum spanning tree) and an graph extension step afterwards, cf. figure F2.42-1. Next,

a feature recognition is performed to gain necessary information considering the possible structure of the surface in the third step. As in object recognition of raster images Mencl and Muller consider features to be regions with special information about the objects structure like paths, edges, point rings and so on. After that, these feature areas are disconnected and/or connected according to certain rules to have a proper description of the objects in the point set (step 4 and 5). In the last step before the triangle filling procedure, the so far computed graph is extended more by connecting associated edges in the graph under consideration of certain constraints. Finally, the triangles are filled into this surface description graph by using a rule system to assure a resulting surface with high accuracy.

As a main concept, Mencl and Muller introduce the concept of feature recognition and clustering to improve the accuracy of the surface description graph according to the assumed surface of the object [MM97]. The idea is the possibility to integrate different kind of recognition algorithms in the main algorithm while maintaining the structural consistency of the SDG.

In contrast to many other methods this approach returns a piecewise linear surface which interpolates exactly the input point set. The algorithm can handle point sets with high changes in point density. This makes it possible to describe objects with only the least necessary amount of points since it is not necessary to oversample areas with low local curvature. The reconstruction of sharp edges in artificial or synthetic objects can be done properly as well as the reconstruction of non-orientable surfaces like Mobius strips, for example.

3. MINIMIZATION OF THE COST OF SURFACE RECONSTRUCTION

3.1 Concept of consecutive application of algorithms

Let's consider a way to decrease the total cost of surface reconstruction. Let we have a cloud of N points and two surface reconstruction algorithms (let's denote them A and B respectively). Let algorithm A is able to reconstruct a correct CAD-model with the specific cost c_A (seconds per a point). Let algorithm B can produce a result in that kN $(0 < k < 1)$ sampled points belong to a correct triangulated surface and the remaining $(1-k)N$ sampled points don't belong to it (such points can have edges and triangles or remain free). Let the specific cost of work of algorithm B is c_B (seconds per a point). Let we have also an algorithm of filtering, that having an unsuccessful result of work of a surface reconstruction algorithm at the input can remove all wrong edges and triangles (i.e. produce an ICADM) with the specific cost c_F (seconds per a point). The usually way of processing the given cloud of points is using only algorithm A. In this case the total cost of surface reconstruction is:

$$C_A = c_A N \quad \text{(E3.1-1)}$$

At the same time surface reconstruction can be made in that way: first we apply algorithm B, then we apply the filtering algorithm, and then for the set of remaining free points we use algorithm A (it is supposed that A can process a set of scattered points with given surface boundaries). The cost of such surface reconstruction is:

$$C_{BA} = c_B N + c_F N + c_A(1-k)N \quad \text{(E3.1-2)}$$

Definition D3.1-1. Let's call the described above using scheme S2 the *concept of consecutive application of surface reconstruction algorithms* (*CCAA*). Let's call an algorithm applied first for processing an input cloud of points the *start algorithm*, an algorithm applied to remove wrong edges and triangles (to obtain an ICADM) the *filtering algorithm*, and an algorithm applied to obtain a CAD-model from the given ICADM the *base algorithm*. This concept can be inductively extended to an arbitrary number of surface reconstruction algorithms having right properties.

Condition C3.1-1. Application of CCAA is more advantageous than application of only the base algorithm if the following condition is fulfilled:

$$kc_A > c_B + c_F \quad \text{(EC3.1-1)}$$

where all the denotations correspond to the described above example.

3.2 Development of a new solution motivation

For realization of CCAA we need to find for a given algorithm chosen as the base algorithm an algorithm to use as the start algorithm and an algorithm to use as the filtering algorithm with the properties, which meet condition C3.1-1. Since nowadays very many such algorithms are designed, we, of course, in many cases can find suitable existing algorithms. However, let's take into consideration, that the absolute majority of existing surface reconstruction algorithms is designed for self-dependent complete surface reconstruction. But, in CCAA the start algorithm has "a right of mistake". Thus, using this fact we can try to design a surface reconstruction algorithm with the specific cost smaller than existing algorithms. Of course, such algorithm in general case can't be used as a self-dependent surface reconstruction algorithm, but it can be successfully applied as the start algorithm within the framework of CCAA. An algorithm designed in accordance with these means and an algorithm to use as the filtering algorithm are described below.

3.3 An algorithm to use as the start algorithm

In general, this algorithm uses the ideas of *greedy triangulation* (*GT*) and belongs to the group of interpolating methods. The GT of a point set in the plane is the triangulation obtained by starting with the empty set and at each step adding the shortest compatible edge between two of the points [BE95]. In 2D a compatible edge is defined to be an edge that crosses none of the previously added edges. This algorithm is an extension of 2D GT-strategy for 3D.

For using GT-strategy in 3D a special very simple and fast test was designed. This test analyzes a topology of a created mesh at the place of prospective inclusion of an edge, and can be formulated as follows: if insertion of the current tested edge leads to at least one of the following situations:

- appearance of edges having more than two adjacent triangles;
- appearance of a tetrahedron;
- appearance of angles lesser than a given threshold value (let's denote it A_T)

then the edge is considered as incompatible and compatible otherwise. The given test uses floating-point operations in a minimal degree, and is performed fast.

To achieve the linear complexity, the step of sorting created set of all the possible edges is realized in that way: we use a one-dimensional regular grid and during the sorting just put a current considered edge in the corresponding cell of the grid. Such grid having about 1000 cells provide a satisfactory quality of sorting.

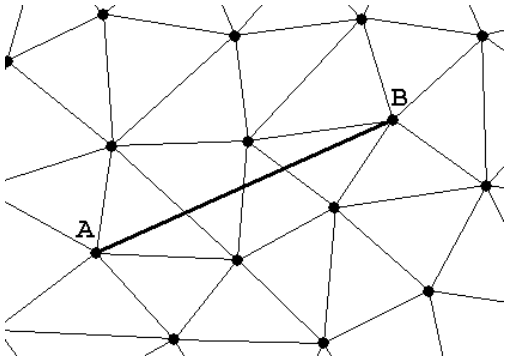


Figure F3.3-1

Naturally, an algorithm based on such simple principles can't provide 100% reliability. For example, the applied test can't prevent appearance of edges having less than 2 adjacent triangles (edge AB in figure F3.3-1). During experiments we have about 85-98% correctly connected points. But it is enough index for the start algorithm.

3.4 An algorithm to use as the filtering algorithm

At the input of the described algorithm we have an unsuccessful result of work of a surface reconstruction algorithm. At the output we have a set of correctly reconstructed regions and a set of free points. In general, the algorithm detects and marks points of correctly triangulated regions. Simultaneously the algorithm removes wrong edges and triangles. The algorithm can improve some errors of the input triangulation, which are caused by generation of redundant edges and triangles, but it doesn't create new edges and triangles.

Condition C3.4-1. On an input set of edges is imposed the following condition: each edge can have only one or two adjacent triangles.

As the basis of this algorithm a variant of the well-known "umbrella" filtering [AGJ02] is used. Each point at the input can have one or several chains of adjacent triangles. These chains can be closed or open. Because of condition C3.4-1 such chains can't have any intersections (shared edges). The possible cases are shown in figure F3.4-1.

Condition C3.4-2. Can be easily proved, that a point of a correctly triangulated region can have only one chain of triangles. This chain is closed for an internal point, and is open for a boundary point.

Condition C3.4-3. Condition C3.4-2 is necessary, but is sufficient only if a given triangulated region is enough extensive. For example, vertices of a single isolated wrong triangle meet this condition. Let we have a triangulated region consist of points, which meet condition C3.4-2. Let's consider such region

correct, if it has at least N_i internal points, where N_i is an empirically chosen parameter.

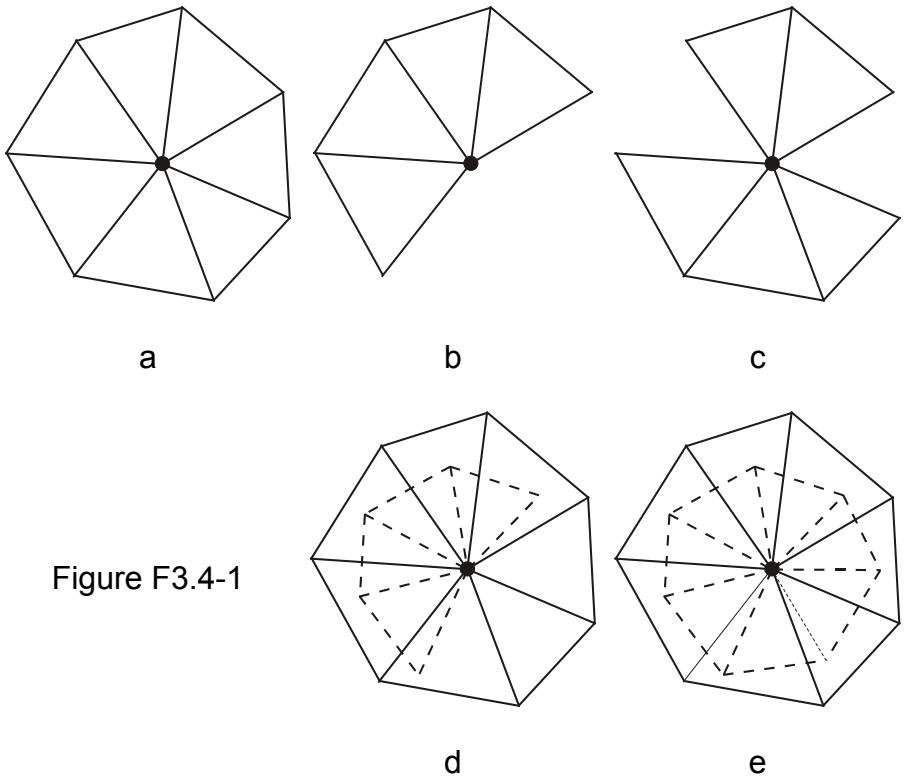


Figure F3.4-1

At the filtering all the input points are sequentially considered. For a current considered point first determination of all chains of triangles is made. The work of the algorithm depending on the obtained configuration of the chains is described in table T3.4-1:

No	Case	Action
1	Only one closed chain (figure F3.4-1a)	The point is considered an internal point;
2	Only one open chain (figure F3.4-1b)	If the point meets condition C3.4-3 then this point is considered a boundary point, otherwise all the triangles of the point are deleted and the point is considered a free point;
3	Several open chains (figure F3.4-1c)	Elimination of all the open chains is made with exception of the chain having the greatest number of triangles, then the point is considered like in case 2;
4	One closed and one or several open chains (figure F3.4-1d)	Elimination of all the open chains is made, then the point is considered an internal point;
5	More than one closed chains (figure F3.4-1e)	Elimination of all the chains is made, and then the point is considered a free point.

Table T3.4-1

In all the cases of elimination of triangles the described algorithm is recursive applied for all the other points, which were vertices of the deleted triangles.

3.5 Results and conclusions

The proposed CCAA has been tested in the following configuration: the algorithm described in paragraph 3.3 with $A_T = 11^\circ$ as the start algorithm and the algorithm described in paragraph 3.4 with $N_i = 10$ (introduced in C3.4-3) as the filtering algorithm have been used. As the base algorithm a projection-based algorithm that is analogous to the algorithm described in [GK02] has been used. This algorithm has a good processing speed in comparison with other existing nowadays algorithms. All tests have been done on a computer with 500 MHz Pentium-3 CPU.

The used algorithms have the following specific costs:
$c_B = 1.7 \cdot 10^{-4}$ (s / pnt) for the start algorithm;
$c_F = 2.8 \cdot 10^{-5}$ (s / pnt) for the filtering algorithm;
$c_A = 2.9 \cdot 10^{-4}$ (s / pnt) for the base algorithm.

Here the results for two well-known models are adduced. They are “Bunny” (figure F3.5-1) and “Bone” (figure F3.5-2).

Figure F3.5-1

Figure F3.5-2

The numerical results of the tests are shown in table T3.5-1. In this table the total number of input points is denoted "N", the rate of correctly triangulated points after application of the start algorithm is denoted "k", the time of surface reconstruction only by the base algorithm is denoted "t_A", the time of surface reconstruction by the given CCAA is denoted "t_{BA}", the achieved speedup is denoted "S".

Model	N	k	t_A, s	t_{BA}, s	S
"Bunny"	35947	0.96	10.4	7.9	1.32
"Bone"	68537	0.88	19.9	16.1	1.24

Table T3.5-1

Thus, formulated CCAA with using the designed algorithms allows improving processing speed of even such fast algorithm like [GK02]. The effect of CCAA is proportional to the value of k.

4. REPAIRING RESULTS OF INCOMPLETE SURFACE RECONSTRUCTION

4.1 Formalization of the problem

In the given chapter a method for obtaining a CAD-model of a given object from an ICADM is described. This method is designed with the assumption, that information contained in the given ICADM is sufficient for obtaining a result with practical value. In general, A (introduced in D1.3-1) may consist of one coherent surface region or several ones isolated from each other (let's call such isolated regions *islands*). As a rule, there are holes on A. Thus A has a set of closed boundaries. This set consists at least of one element. An element of this set can be the boundary of a hole or the boundary of an island. Sometimes an answer to the question, what exactly a given boundary bounds (a hole or an island) is obvious (figure F4.1-1a), and sometimes is not (figure F4.1-1b). Thus, we need to have a strict criterion to answer this question in any case. For the generality in the further statement let's suppose (if an other supposition isn't explicitly given), that a considered boundary is a smooth curve and the surface region bounded by this boundary is a smooth surface region.

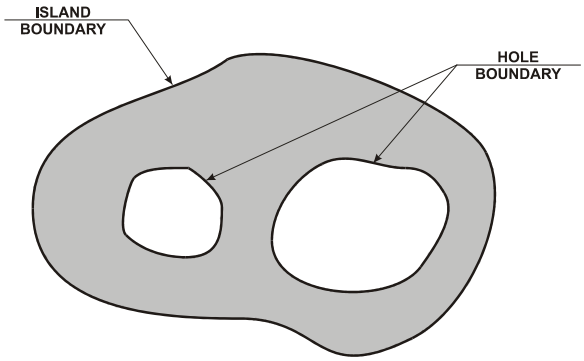


Figure F4.1-1a

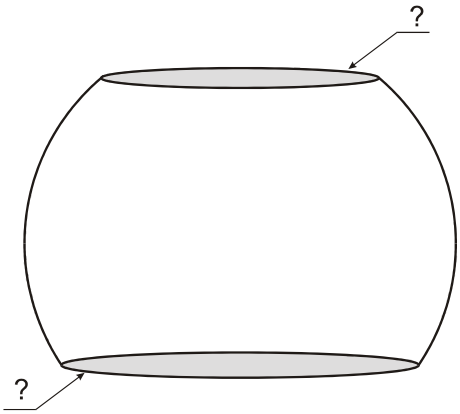


Figure F4.1-1b

Definition D4.1-1. Let's consider a boundary (let's denote it B). Let's define a *main chord vector* as the vector connecting two points on the boundary with condition, that the second point is the most distant from the first point. Let we have a main chord vector (vector $\overrightarrow{XY}$ in figure FD4.1-1). In a close neighborhood of point Y the boundary can be approximated by the tangent line at this point. In the same neighborhood the surface can be approximated by the tangent plane at the point. The tangent line lies in the tangent plane and splits it in two half-planes. Let's call the half-plane corresponding to the surface the *internal half-plane* at Y. At Y let's define the angle ϕ as the angle between $\overrightarrow{XY}$ and the internal half-plane. For the given boundary let's determine the *boundary integral value* (D):

$$D = \oint_B \cos\phi(b)\ db \qquad \text{(ED4.1-1)}$$

where b is the arc length.

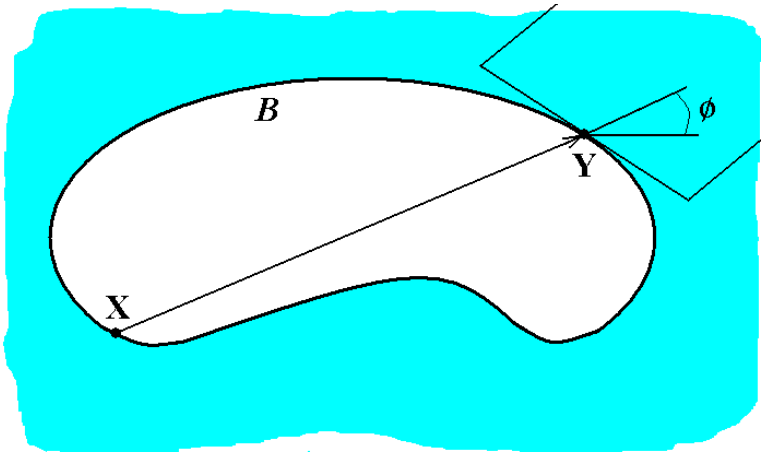


Figure FD4.1-1

A boundary is classified as the boundary of a hole, if $D > 0$ for this boundary, otherwise as the boundary of an island.

Definition D4.1-2. Since each island is a part of a reconstructed surface of an object, we can determine the external and the internal sides of the given island. Let's call an island with determined sides an *oriented* island. A method of determination of the orientation of an island is described in section 4.48. For the island let's define the *positive direction* of movement along its boundary the counter-clockwise direction, if we look to the external side. For the boundary of a hole inside an island the positive direction is reverse.

Definition D4.1-3. Let there be a hole. Let's call this hole a *simple hole* if a proper surface inside it can be reconstructed by an existing fast and simple surface reconstruction method (let's call such method a "*darning*" *method*), and a *complex hole* otherwise.

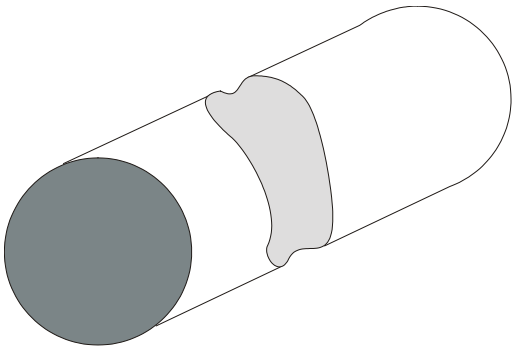


Figure F4.1-2

Naturally, that the same hole can be simple concerning a given “darning” method and can be complex concerning another one. For example the hole in figure F4.1-2 is most likely simple concerning a spline-based method and is complex concerning a projection-based method that uses single-stage projection onto a plane. For the further statement only the fact, that reconstruction of a proper surface inside a simple hole is not a problem, is important.

When considering the problem of obtaining a correct CAD-model from a given ICADM it is necessary to take into account, that reconstruction of $\overline{A}$ (introduced in D1.3-1) can be made only on the basis of analysis of behavior of A and distribution of free points. These points typically have the following properties:

- density and uniformity of distribution are sufficiently smaller than the corresponding indices of points of A;
- inaccuracy of determination of the coordinates is greater than this index of points of A.

Thus, usually we don't have necessary information to provide a complicated behavior of reconstructed $\overline{A}$. And usually we can’t reconstruct $\overline{A}$ with the same precision like A. In general, adequacy of reconstruction of $\overline{A}$ has a likelihood character. Taking into account these considerations let’s work out the problem with the assumption that behavior of $\overline{A}$ isn’t too complicated. In general, probability of adequate reconstruction of $\overline{A}$ is higher in regions where A has a simple behavior.

Definition D4.1-4. Each ICADM can be related to one of the following classes:

ICM1: A is represented by one coherent region, that has only simple holes (figure FD4.1-4a, the holes are simple concerning the algorithm used in paragraph 3.5 as the base algorithm);

ICM2: A is represented by one coherent region too, but there are both simple and complex holes inside (figure FD4.1-4b);

ICM3: A is represented by the aggregate of several islands, which have possible holes of the both types (figure FD4.1-4c).

a b c

Figure FD4.1-4

In accordance with definition D4.1-3 obtaining a correct CAD-model from a given ICADM of class ICM1 is not a problem. Thus, our problem is to find a method of correct processing ICADMs of classes ICM2 and ICM3.

4.2 Development of a new solution motivation

Among the groups of methods considered in chapter 2 warping-based methods [AS95, AS96, BH93, BH94, DH94, HG92, HJL93, MBL91, RA00, RM95, MS91, SRT92, TM91, TWK88, VT92] have necessary features to solve the above-formulated problem. Of these methods, the ones based on physical modeling of the warping are [AS95, AS96, DH94, HG92, RA00, VT92] the most suitable. This technique means, that there is an initial membrane (mesh) that is topologically equivalent to the original surface of a given object (in most cases it is the topology of a closed sphere). Elements of the membrane interact both with each other and with sampled points by rules defined by a given physical model. As a result of this interaction the membrane takes deformation to approximately the original surface of the object. A system of material points connected by springs is usually used as the interaction model. Material points corresponding to sampled points are considered fixed (figure F2.32-1). Such model can be expressed as a system of linear differential equations of degree 2. This system is solved iteratively. In general, application of the warping-based methods to solve the considered problem has the following advantages and disadvantages:

Advantages:

- If a modeled membrane at the beginning of the process is topologically equivalent to the original object surface, then topologically correctness of reconstructed $\overline{A}$ is provided.
- $\overline{A}$ is represented by a region (regions) of the surface of minimal potential energy of the modeled membrane. It is the most probable approximation of the original surface, and in many cases the condition of smoothness is satisfied as well.
- ICADMs of all the three classes can be processed by the same method.

Disadvantages:

- Topological correctness of $\overline{A}$ is provided by the fact that during the modeling the membrane is considered entirely. Of course, at the final step only regions of the membrane corresponding to $\overline{A}$ can be modeled, but the cost of elimination form the modeling regions of the membrane corresponding to A masks optimization effect. Therefore, the cost of obtaining a CAD-model from a given ICADM is not essentially smaller than the cost of obtaining the same result from the corresponding cloud of points.
- In general, the process of modeling is costly enough. To reduce this cost many existing methods use the warping technique only at the final step to obtain a total shape of the membrane. An initial shape of the membrane approximating the original object surface is obtained by using

a different algorithm that is not so costly. But for ICADMs of classes ICM2 and ICM3 (especially) such algorithm often can't make an initial shape of the membrane with the correct topology.

Thus, for solution of our problem it would be good to find a method with the following properties:

- the cost depending only on the number of points included in $\overline{A}$ (they are points on boundaries of A and free points);
- a capability to determine the correct topology of $\overline{A}$ with the cost smaller than the cost of the modeling membrane and with robustness greater than the robustness of existing fast surface reconstruction algorithms;
- a capability to provide behavior of $\overline{A}$ in accordance with the surface of minimal potential energy of a membrane that is correctly connected to boundaries of A.

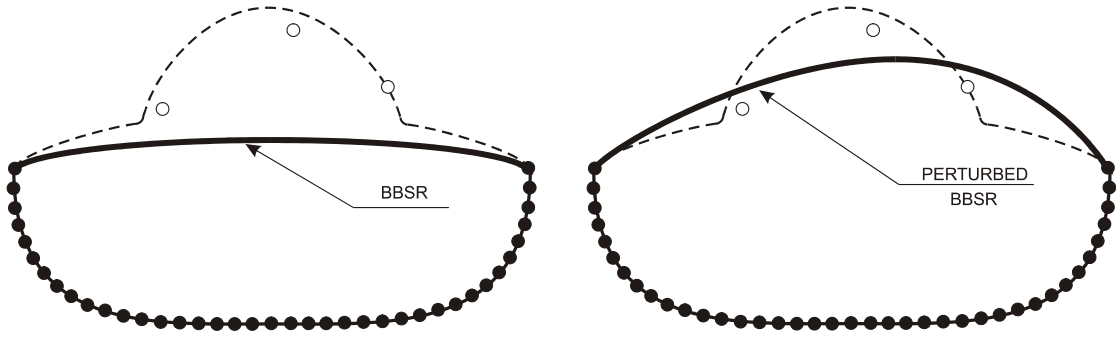
4.3 Basic concepts

4.31 Concept of perturbed boundary-based surface

Definition D4.31-1. Let's consider $\overline{A}$ of a given ICADM. In general case it is the aggregate of one or several coherent surface regions bounded by boundaries of A. Let each of these regions meet the following conditions:

- it is completely defined only by the corresponding sections of the boundaries of A and the normals of A along these sections;
- it is topologically equivalent to the corresponding region of the original object surface;
- it is some compromise between minimization of its area and interpolation of the corresponding region of A.

Let's call such region of $\overline{A}$ a *boundary-based surface region* (*BBSR*).



a b

Figure F4.31-1

Taking into account the mentioned in paragraph 4.1 properties of free points, let's use for reconstruction of $\overline{A}$ the following approach. At the first step determination of BBSRs is made (figure F4.31-1a). In accordance with definition D4.31-1 free points are no taken into consideration for determination of their behavior. At the second step we use the free points as a perturbing factor for the obtained BBSRs (figure F4.31-1b).

4.32 Concept of bridges

Let's note that to reduce of an ICADM of class ICM3 to an ICADM of class ICM2 and then to an ICADM of class ICM1 it is enough to make reconstruction of $\overline{A}$ only in properly chosen local regions (figure F4.32-1).

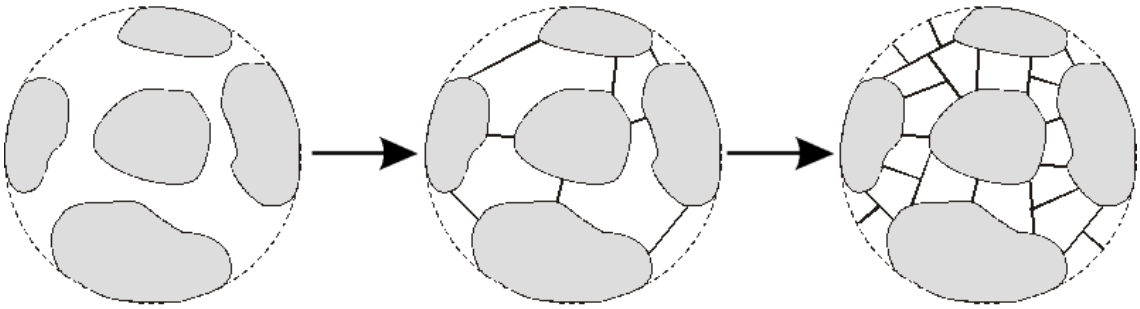


Figure F4.32-1

Definition D4.32-1. The simplest case of such local region is a curve connecting two selected boundary points of A with definition of the external normal vector along the curve (figure FD4.32-1). Let's call such curve a *bridge* and the boundary points (A and B in the figure) connected by this bridge its *supporting points.*

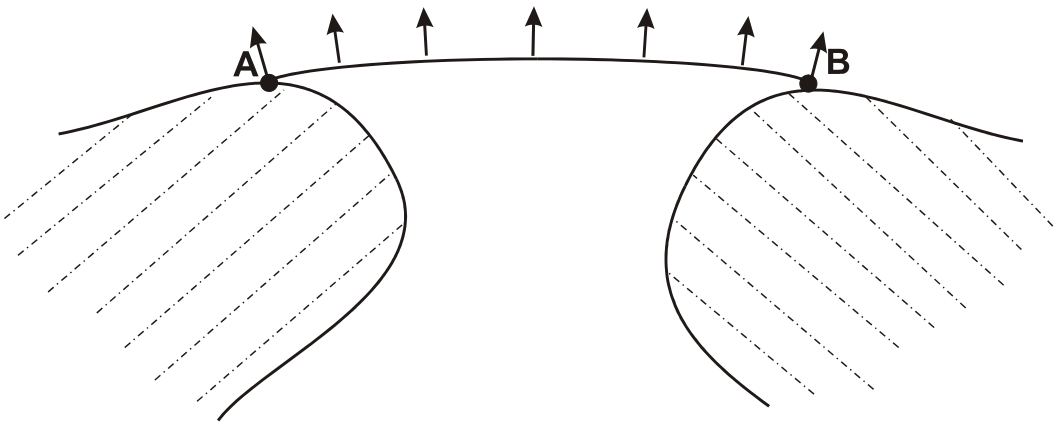


Figure FD4.32-1

For further description of properties of a bridge, let's use one-dimensional coordinate system where the coordinate of a bridge point is defined by the length of the corresponding arc. Let's denote a bridge by the pair of its supporting points (for example AB) where at the first position there is the point selected as the origin of the coordinate system.

Definition D4.32-2. Because of the fact that at each point of a bridge the corresponding normal vector is defined we can interpret a bridge as an infinite narrow surface strip. Let's use the term "*the surface of a bridge*" meaning this interpretation. Let's also consider, that *a bridge lays on a given surface* if the surface of the bridge coincides with the given surface at each point of the bridge.

Taking into account the formulated in section 4.31 concept of perturbed BBSR, let's make construction of a bridge in two steps. At the first step we obtain a bridge connecting given supporting points that lies on a BBSR. At the second step we make correction of the trajectory of the bridge with taking into account nearby free points.

4.33 General concept of reconstruction of $\overline{\mathrm{A}}$

Thus, let's make obtaining a correct CAD-model from a given ICADM (for generality let's assume that it belongs to class ICM3) in the following way:

1. we make reduction of the given ICADM to an ICADM of class ICM2 by construction bridges between the existing islands;
2. we chose a suitable surface reconstruction algorithm as a "darning" algorithm (therefore, we define the criterion of a simple hole);
3. we make reduction of the obtained ICADM of class ICM2 to an ICADM of class ICM1 by construction bridges inside existing complex holes;
4. we make surface reconstruction inside each simple hole by using the chosen "darning" algorithm.

The key problem of implementation of the formulated above concept is determination of properly sets of bridges for the first and second steps (let's denote them *ICM3→ICM2* and *ICM2→ICM1* respectively).

4.4 Determination of a set of bridges at step ICM3→ICM2

4.41 Concept of an indicator field

Definition D4.41-1. Let's define *quality* of a given bridge as estimated in some way probability of the bridge to be topologically correct (probability of belonging the bridge to a correct BBSR).

To implement step ICM3→ICM2 for a given set of islands we need to determine a proper set of bridges and these bridges should have the quality as high as possible. The most natural way to solve this problem is consecutive consideration of each of the given islands (let's call a currently considered island the *base island*) and determination for it a set of the best quality bridges with other islands. The concept of a method of determination of this set is described in the given paragraph. The total set of bridges is determined on the base of union of such sets bearing in mind the conditions described in section 4.47.

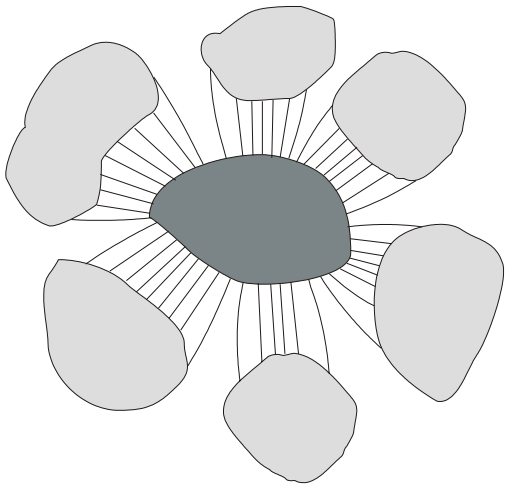


Figure F4.41-1

Let's assume that the boundaries of islands are sources of a special field (let's call it *indicator field*, H). For a given base island possible bridges connecting it with other islands are defined by force lines of this field (figure F4.41-1). These force lines come out of points on the boundary of the base island and go to the boundaries of other islands.

Let's assume that this field contains sufficient information to determine both: the trajectory of a force line as well as the external normal vector along this trajectory. Thus a force line of the field locally defines a BBSR in an infinitely small neighborhood of its trajectory.

Let's assume that the average scalar value of the field tension along a given force line is the value of the quality of the bridge represented by this force line.

Let's note, that in accordance with the last assumption the quality of a bridge with a high enough probability is proportional to the scalar value of the tension at its origin point. Taking it into consideration, to determine the required bridges we can track the force lines outgoing from points on the boundary of the base island with the highest scalar values of the tension until we obtain a sufficient number of bridges with proper quality. A natural way to track a force line is application of an iterative method, when at each iteration step on the base of a

current point of the force line we obtain the next point of this line by analyzing the field tension at the current point.

4.42 Formalization of the quality of a simple bridge

Let's solve the task of formalization of the quality of a bridge for a particular case that has direct relation to the task of construction of the indicator field.

Definition D4.42-1. In the further statement mathematical objects, being a set of certain number of scalar and vector values, are used. Let's call such object a *complex* and denote it by capital letter with one or more signs "∘" above. In declaration of a complex its components are listed in braces. For example: $\overset{\circ}{A}\{X, Y, \vec{a}, \vec{b}\}$ is a complex that consists of two points and two vectors; $\overset{\circ\circ}{A}\{\overset{\circ}{A}, \vec{c}\}$ is a complex that consists of a complex and a vector. Let's call a complex that consists of a point and a unit vector an *oriented point*. If the term "point" is used for a complex, it means that the given complex is an oriented point.

4.421 Base definitions and conditions

Condition C4.421-1. Let's consider an island and a bridge coming out of it in a little neighborhood defined by ε (a value close to zero) of the corresponding supporting point of the bridge (figure FC4.421-1). Let's formulate a necessary condition for topological correctness of joining the bridge and the island. First, let's require that the external normal vector of the bridge at the supporting point is perpendicular to the tangent line (τ) to the island boundary at the same point. It means that in the considered neighborhood the boundary lies on the bridge surface. Secondly, let's require, that if we go on the external side of the bridge to the island, then we should reach the external side of the island.

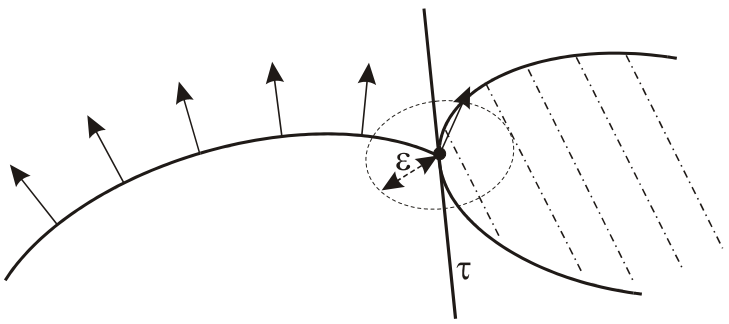


Figure FC4.421-1

Definition D4.421-1. Let's consider an island (figure FD4.421-1), a point O on its boundary and the tangent line (τ) to the boundary at this point. Let there be a point (X) outside the island. Let's consider the plane (α) containing τ and X. Let's arbitrarily define the external side of α and the corresponding external normal vector ($\vec{n}$). Let's call point $\overset{\circ}{X}\{X,\vec{n}\}$ a point *compatible* with point O, if during the movement along straight line XO from point X to point O on the external side of α we end up on the external side of the island, and *non-compatible* if we are on the internal side. In the case when XO coincides with τ let's call $\overset{\circ}{X}$ a *degenerated* point, because in this case we can't reach the island surface.

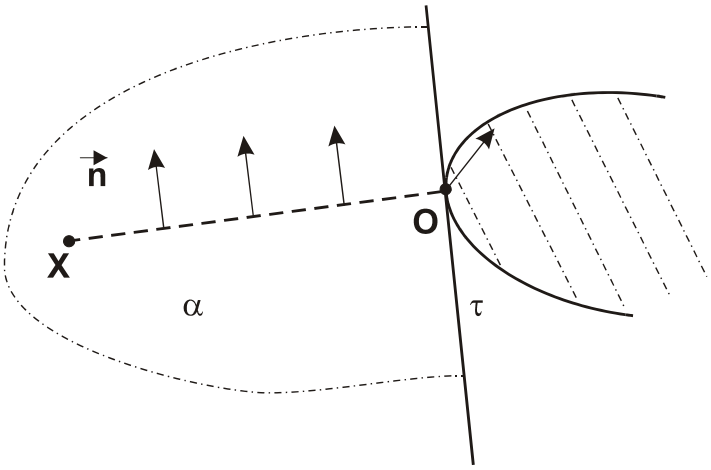


Figure FD4.421-1

Definition D4.421-2. Let's consider an island, a point O on its boundary, and the tangent line to the boundary at this point (figure FD4.421-2). Let's call the unit vector $\vec{\tau}$ on the tangent line with the origin at O directed in accordance with the positive direction of the boundary the *tangent vector* to the boundary at the given point.

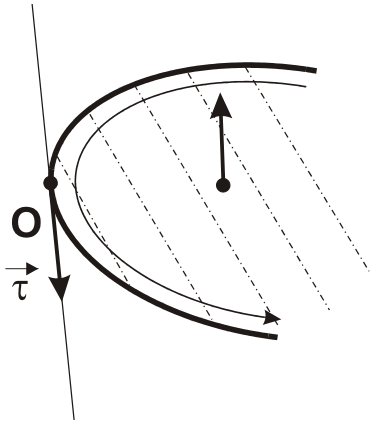


Figure FD4.421-2

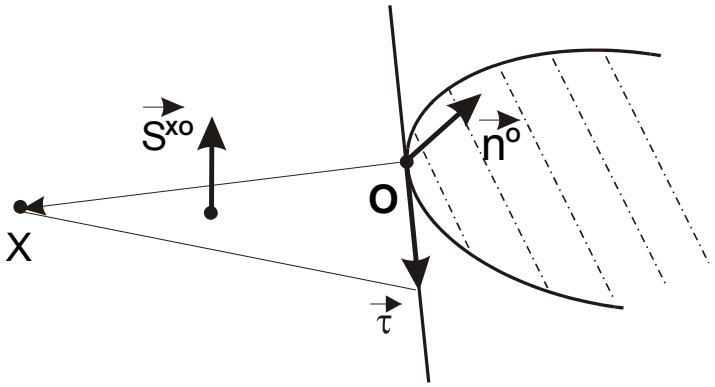


Figure FD4.421-3

Definition D4.421-3. Let's consider an island with a point $\overset{\circ}{O}\{O,\vec{\tau}\}$ on its boundary (where $\vec{\tau}$ is the tangent vector at O) and a point X outside the island (figure FD4.421-3). Let's call the vector $\vec{s}^{XO}$ making point $\overset{\circ}{X}\{X,\vec{s}^{XO}\}$ compatible with point O the *compatibility vector* for point X concerning point O. Using the properties of the vector product let's define this vector by the following formula:

$$\vec{s}^{XO} = \frac{\overrightarrow{OX}}{|\overrightarrow{OX}|} \times \vec{\tau} \qquad \text{(ED4.421-3)}$$

In that formulation the length of $\vec{s}^{XO}$ is equal to the absolute value of the sinus of the angle between the vectors $\overrightarrow{OX}$ and $\vec{\tau}$. It reveals the fact that the compatibility vector for a degenerated point doesn't exist. Let's note that in accordance with ED4.421-3 the compatibility vector is the same for all the points on closed half-line OX with exception of point O (because the compatibility vector for a point concerning itself makes no sense).

Condition C4.421-2. Let's consider a bridge with the trajectory represented by the segment between its supporting points (let's denote them A and B). From ED4.421-3 immediately follows that the necessary and sufficient condition for such bridge to fulfill condition C4.421-1 is that this bridge should have at the supporting points the external normal vectors (let's denote them $\vec{n}^{A}$ and $\vec{n}^{B}$ respectively) defined in the following way:

$$\vec{n}^{A} = \frac{\vec{s}^{BA}}{|\vec{s}^{BA}|} \qquad \text{(EC4.421-2a)}$$

$$\vec{n}^{B} = \frac{\vec{s}^{AB}}{|\vec{s}^{AB}|} \qquad \text{(EC4.421-2b)}$$

Condition C4.421-3. Let's define a necessary condition for the segment between two given points (let's denote them A and B) on the boundaries of given islands (let's denote them X and Y respectively) to be the topologically correct trajectory of a bridge by the following formula:

$$q_T = \vec{s}^{BA} \vec{s}^{AB} > 0 \qquad \text{(EC4.421-3)}$$

Index q_T can be interpreted in the following way. In accordance with C4.421-2 for any bridge with the given trajectory the external normal vectors at the supporting points are defined by EC4.421-2. If we go from a point of X on its external side to the segment, at point A we are on the segment and the external normal vector has the same direction as $\vec{s}^{BA}$ has. If we go along the segment to island Y keeping this external normal direction, we end up on: the external side of Y if $q_T > 0$ (and "easiness" of the transition is proportional to q_T); the internal side of Y if $q_T < 0$; the boundary of Y if $q_T = 0$. It is obvious, that only the first case corresponds to a topologically correct connection of the islands.

Definition D4.421-4. Let's consider a bridge with given supporting points and the trajectory represented by the segment between them. Let's assume that this bridge has distribution of the external normal vector that meets condition C4.421-2 and provides the simplest behavior of the bridge surface (in accordance with the assumption made in paragraph 4.1). Let's call such bridge a *simple bridge.*

Let's consider a simple bridge from the viewpoint of condition C4.421-3. Because of the fact that C4.421-3 is only necessary condition we can estimate by the value of index q_T only a probability of topological correctness of the bridge. It is easy to see, that if $q_T > 0$ then the bridge is probably topologically correct and the value of q_T increases with increasing this probability, otherwise it is wrong.

4.422 The basic task

Let we have a given topology of islands, one of them is chosen as the base island. Let' consider (figure F4.422-1): $\overset{\circ}{X}\{X, \vec{n}^X\}$ is a point in the space between given islands; $\overset{\circ}{O}\{O, \vec{n}^O, \vec{\tau}^O\}$ is a complex consisting of the following components: O is a point on the boundary of an island, that isn't the base island; $\vec{n}^O$ is the external normal vector to the island surface at point O; $\vec{\tau}^O$ is the tangent vector at this point.

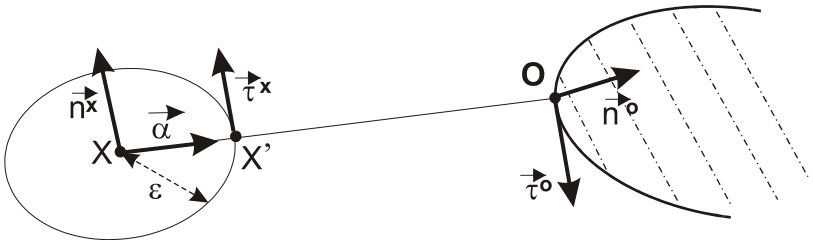


Figure F4.422-1

In the beginning let's assume that $\overset{\circ}{X}$ defines a round platform with an arbitrary small radius ε. This platform is perpendicular to $\vec{n}^{X}$ and has point X as the center. Let's consider this platform as an elementary island. Since ε is arbitrary small ($\lim_{\varepsilon \to 0} X' = X$) let's assume that any bridge connecting this elementary island with another island has as the supporting point on the elementary island point X.

Let's consider the simple bridge XO. We need to formalize the quality (let's denote it Q) of this bridge with assumption that the bridge is a part of a bridge connecting a point on the boundary of the base island and point O.

Firstly, let's note, that in accordance with C4.421-3 $Q \sim q_T$. Let's also take into account the following conditions for the trajectory of the bridge, which have influence on the quality:

Condition C4.422-1. The trajectory should be as short as possible.

Condition C4.422-2. The trajectory should lie in region that has as high concentration of free points as possible.

Condition C4.422-4. The trajectory should lie in the plane defined by $\overset{\circ}{X}$ as entirely as possible.

Condition C4.422-4. The trajectory should interpolate behavior of the surface of the island containing point O in some neighborhood of this point as entirely as possible.

Conditions C4.422-1 and C4.422-2 are a consequence of the fact that the probability of correctness of the bridge increases if its trajectory lies near to sampled points. At the same time condition C4.422-1 is dominating because free points have lesser density and lesser trustworthiness of the coordinates. Also, for a bridge in general case, condition C4.422-1 expresses the fact that the cost of construction of the bridge is proportional to its length. Condition C4.422-3 expresses a general assumption that behavior of a BBSR isn't too complicated (it follows from D4.31-1). Condition C4.422-4 expresses the fact that usually only A has a proper degree of trustworthiness.

Let's formalize the indices of fulfillment of these conditions by the considered bridge.

4.423 The index of fulfillment of condition C4.422-1

Let's consider that this index (let's denote it q_1) is defined by the function:

$$q_1 = q_1(L) \quad \text{(E4.423 -1)}$$

which has the following properties:

$$q_1(L) \geq 0 \quad \text{(E4.423-2a)}$$
$$\frac{dq_1}{dL} \leq 0 \quad \text{(E4.423-2b)}$$
$$q_1(0) = 1 \quad \text{(E4.423-2c)}$$
$$\lim_{L\to\infty} q_1(L) = 0 \quad \text{(E4.423-2d)}$$

4.424 The index of fulfillment of condition C4.422-2

To define this index let's assume that free points can compress the surrounding space (like sources of the gravitational field in the theory of relativity). As a result the length of a bridge passing in a neighborhood of free points shrinks. Let's assume the length of the bridge with taking into account such compression as the value of the index. To determine this length let's assume, that each free point is a source of an spherically symmetric scalar field $G = G(R)$, where R is the distance between the free point and a given space point. Let's assume that function $G(R)$ has the following properties:

$$G(R) \geq 0 \quad \text{(E4.424-1a)}$$
$$\frac{dG}{dR} \leq 0 \quad \text{(E4.424-1b)}$$
$$G(0) = \frac{k_t}{1-k_t},\ k_t \in (0,1) \quad \text{(E4.424-1c)}$$
$$\lim_{R\to\infty} G(R) = 0 \quad \text{(E4.424-1d)}$$

where k_t is the "factor of trustworthiness" for the coordinates of the given free point.

Let's assume that an elementary segment with length dl located at a point A where the value of the field tension is G^A has the effective length

$dl' = \frac{dl}{1+G^A}$. Therefore, in the case $R=0$, if $k_t \to 0$ then $dl' \to dl$, and if $k_t \to 1$ then $dl' \to 0$.

Thus, the index (let's denote it q_2) can be expressed by this formula:

$$q_2 = \int_0^L \frac{dl}{1+\sum_{i=1}^{m(\Omega)} G(|\vec{P_i} - \vec{Y}(l)|)} \qquad \text{(E4.424-2)}$$

where
Ω is the region around the considered bridge defined by the condition stating that only free points from this region have valuable influence on the length of the bridge (the criterion of this condition is considered in detail in section 4.47);
m is the total number of free points in Ω;
P_i is the i-th free point of these free points;
$\vec{Y} = \vec{Y}(l)$ is the parametric equation of the bridge;
L is the length of the bridge.

The limiting values of the index: for the maximal fulfillment of the condition $q_2 \to \frac{1-k_t}{1+k_t(m-1)} L$, for the minimal fulfillment $q_2 = L$.

4.425 The index of fulfillment of condition C4.422-3

We can formulate this index (let's denote it q_3) in the following way:

$$q_3 = |\sin(\vec{n}^X, \overrightarrow{XO})| = |\vec{n}^X \times \vec{d}| \qquad \text{(E4.425-1)}$$

where $\vec{d} = \frac{\overrightarrow{XO}}{|\overrightarrow{XO}|}$.

The limiting values of the index: for the maximal fulfillment of the condition $q_3 = 1$, for the minimal fulfillment $q_3 = 0$.

4.426 The index of fulfillment of condition C4.422-4

Let's define this index (let's denote it q_4) by the following formula:

$$q_4 = \frac{1 + \vec{s}^{\,\vec{XO}} \cdot \vec{n}^{\,\vec{O}}}{2} \qquad \text{(E4.426-1)}$$

The limiting values of the index: for the maximal fulfillment of the condition $q_4 = 1$, for the minimal fulfillment $q_4 \to 0$.

4.427 Derivation of the total quality index

To formalize the quality index of bridge XO we need to construct an expression based on the indices q_T, q_1, q_2, q_3, q_4 taking into account the logic of influence of each of these indices on the quality.

Firstly, it is naturally to unite the indices q_1 and q_2 (lets denote the united index q_{12}) by using q_2 as argument L of the function defining q_1 (introduced in subsection 4.423):

$$q_{12} = q_1(q_2) \qquad \text{(E4.427-1)}$$

Then, let's take into consideration that:

- the indices q_{12}, q_3, q_4 are proportional to degree of fulfillment of the corresponding conditions by the bridge;
- the possible values of the indices q_{12}, q_3, q_4 are non-negative;
- if any of the indices q_{12}, q_3 is equal to 0 then the quality is equal to 0 as well.

Thus, it is possible to write that $Q \sim q_T q_{12} q_3$. In this case if $Q > 0$ the bridge is probably correct, otherwise it is unconditionally wrong.

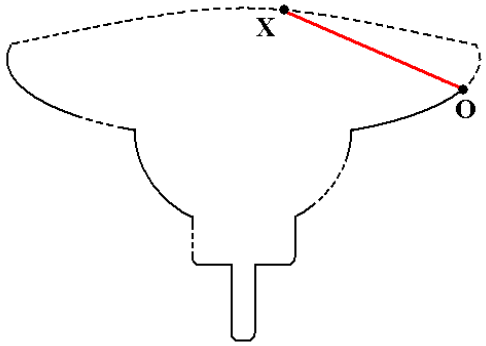


Figure F4.427-1

Then, let's take into account, that condition C4.422-4 often is not fulfilled even if the bridge is absolutely correct (figure F4.427-1). Nevertheless, fulfillment of this condition by the bridge increases its probability to be correct. Thus, let's express Q by the following formula:

$$Q = q_T q_{12} q_3 (1 + k_i q_4) \qquad \text{(E4.427-2)}$$

where k_i is the *factor of interpolation* that allows controlling influence of condition C4.422-4 on the value of Q.

Let's simplify formula E4.427-2. Using EC4.421-3 this formula can be rewritten in the following way:

$$Q = \vec{s}^{\,OX}\, \vec{s}^{\,XO} q_{12} q_3 (1 + k_i q_4) \qquad \text{(E4.427-3)}$$

In accordance with ED4.421-3:

$$\vec{s}^{\,XO} = -\vec{d} \times \vec{\tau}^{\,O} \qquad \text{(E4.427-4)}$$

Taking into consideration the assumption that $\overset{\circ}{X}$ defines a round platform we can write:

$$\vec{\tau}^{\,X} = \frac{\vec{n}^{\,X} \times \vec{d}}{|\vec{n}^{\,X} \times \vec{d}|} \qquad \text{(E4.427-5)}$$

then

$$\vec{s}^{\,OX} = \vec{d} \times \vec{\tau}^{\,X} \qquad \text{(E4.427-6)}$$

By substitution E4.427-6, E4.427-4, and E4.425-1 in E4.427-3 we obtain:

$$Q = (\vec{d} \times (\vec{n}^{\,X} \times \vec{d}))\vec{s}^{\,XO} q_{12} (1 + k_i q_4) \qquad \text{(E4.427-7)}$$

or

$$Q = (\vec{n}^{\,X} - \vec{d}(\vec{n}^{\,X} \vec{d}))\vec{s}^{\,XO} q_{12} (1 + k_i q_4) \qquad \text{(E4.427-8)}$$

4.43 Formalization of the indicator field

Let's make formalization of the indicator field on the basis of the considered in subsection 4.422 task (all the used conditions and conventional signs remain the same).

As an elementary source of the field let's define a complex of the type $\overset{\circ\circ}{O}\{\overset{\circ}{O}\{O,\vec{n}^{O},\vec{\tau}^{O}\},h^{O}\}$, and let's define a complex of the type $\overset{\circ}{X}\{X,\vec{n}^{X}\}$ as an object, that the field affects; where h^{O} is the "charge" associated with source $\overset{\circ\circ}{O}$, the remaining components of the complexes are the same like in subsection 4.422.

Let's assume that source $\overset{\circ\circ}{O}$ creates a field that makes influence on $\overset{\circ}{X}$ by force $\vec{f}$ acting at X and defined by the following equation:

$$\vec{f} = h^{O}\vec{d}Q \qquad \text{(E4.43-1)}$$

where

$\vec{d} = \dfrac{\overrightarrow{XO}}{|\overrightarrow{XO}|}$;

Q is the quality of bridge XO defined by (E4.427-8).

Thus, point O "attracts" point X, if a bridge between the base island and the elementary island defined by $\overset{\circ}{X}$ can be extended to point O by the probably topologically correct simple bridge and "repulses" it otherwise. The attraction force is proportional to the quality of this bridge.

Let's consider the case, when O belongs to the boundary of the base island. Because a bridge between two points of the base island boundary is topologically wrong, let's assume that sources of the field of the base island boundary have the negative "charge". It implies that in the considered case point O "repulses" point X if there is a possible correct smoothly joined simple bridge between these points.

Thus, force $\vec{f}$ is a function of one scalar and five vector arguments: $\vec{f} = \vec{f}(\overset{\circ\circ}{O},\overset{\circ}{X})$. During iterative tracking a force line of field H the components of $\overset{\circ\circ}{O}$ are considered as constants. The coordinates of X at a given iteration are defined at the previous iteration. Thus, at a given point X the acting force $\vec{f}$ can be considered as a function of $\vec{n}^{X}$.

Definition D4.43-1. Let's define the tension of field H created by a given source ($\overset{\circ\circ}{O}\{\overset{\circ}{O}\{O,\vec{n}^{O},\vec{\tau}^{O}\},h^{O}\}$) at a given point ($X$) as an operator H, whose result of action on a unit vector ($\vec{n}$) at this point is the force acting at the point:

$$\vec{f}(\vec{n}) = \mathrm{H}\vec{n} \qquad \text{(ED4.43-1)}$$

Let's also define the length of the force vector as the scalar value of the tension.

Like that, operator H is a function of one scalar and four vector arguments: $\mathrm{H} = \mathrm{H}(\overset{\circ\circ}{O}, X)$. This operator should provide the property of superposition of field H. Therefore, for this operator for any space point X, any unit vector $\vec{n}$, and any value of "charge" h^{O} the following equations would be true:

$$\mathrm{H}_1 + \mathrm{H}_2 = \mathrm{H}_2 + \mathrm{H}_1 \qquad \text{(E4.43-2a)}$$

$$(\mathrm{H}_1 + \mathrm{H}_2) + \mathrm{H}_3 = \mathrm{H}_1 + (\mathrm{H}_2 + \mathrm{H}_3) \qquad \text{(E4.43-2b)}$$

$$\mathrm{H}_1\vec{n} + \mathrm{H}_2\vec{n} = (\mathrm{H}_1 + \mathrm{H}_2)\vec{n} \qquad \text{(E4.43-2c)}$$

$$\mathrm{H}(\overset{\circ}{O}, kh^{O}, X) = k\mathrm{H}(\overset{\circ}{O}, h^{O}, X) \qquad \text{(E4.43-2d)}$$

where k is a constant.

Field H providing fulfillment of these conditions is a tensor field. The tension of this field created by the given elementary source $\overset{\circ\circ}{O}$ at the given point X is represented by the following matrix:

$$\mathrm{H}(\overset{\circ\circ}{O}, X) = h^{O}\mathrm{AB} \qquad \text{(E4.43-3)}$$

where

$$\mathrm{A} = m\begin{bmatrix} d_x s_x^{XO} & d_x s_y^{XO} & d_x s_z^{XO} \\ d_y s_x^{XO} & d_y s_y^{XO} & d_y s_z^{XO} \\ d_z s_x^{XO} & d_z s_y^{XO} & d_z s_z^{XO} \end{bmatrix} \qquad \text{(E4.43-4)}$$

where

$d_{x,y,z}$, $s_{x,y,z}^{XO}$ are components of the corresponding vectors $\vec{d}$ and $\vec{s}^{XO}$;

$m = q_{12}(1 + k_i q_4)$;

$$\mathrm{B} = \begin{bmatrix} 1-d_x^2 & -d_x d_y & -d_x d_z \\ -d_y d_x & 1-d_y^2 & -d_y d_z \\ -d_z d_x & -d_z d_y & 1-d_z^2 \end{bmatrix} \quad \text{(E4.43-5)}$$

Let's note that in spite of the fact that the field is defined on the base of analyzing of the quality of simple bridges, force lines of the field in many cases have behavior that is similar to behavior of spline curves. This behavior considerably depends on degree of influence of each of the indices q_1, q_2, q_4. These degrees are controlled by the functions $q_1(L)$, $G(R)$, and factor k_i respectively (they are introduced in the subsections 4.423, 4.424, 4.427). By proper definition of these functions and the factor we can manage the behavior of force lines of the field. The examples of such behavior are shown in figure F4.43-1. For each case the corresponding dominating index is adduced.

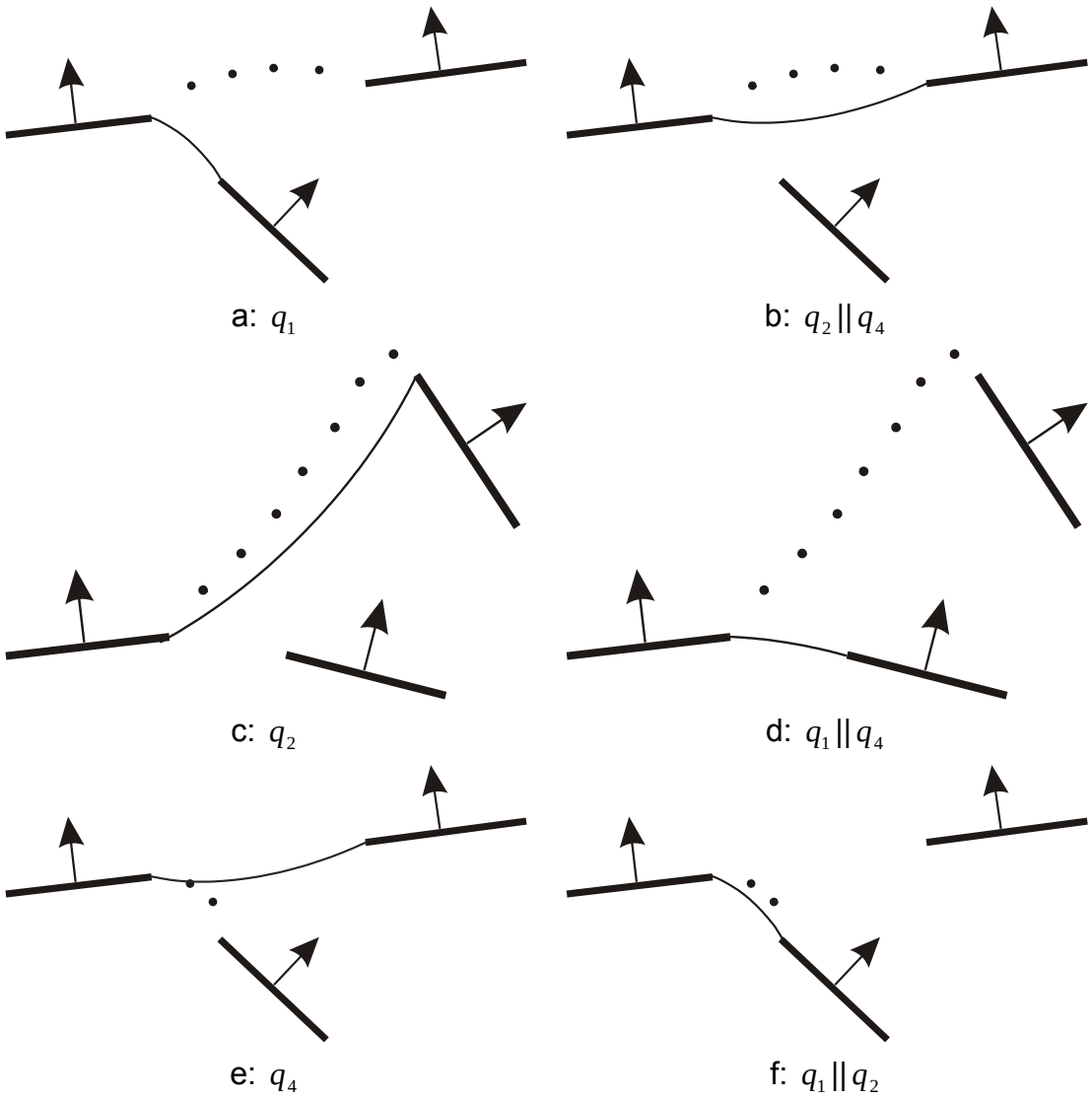


Figure F4.43-1

In conclusion, let's note that to increase robustness of the method, formula E4.43-1 can be supplemented by the following condition of shielding:

$$\vec{f} = \delta^{XO} h^{O} \vec{dQ} \quad \text{(E4.43-6)}$$

where $\delta^{XO} = 0$ if segment XO crosses A and $\delta^{XO} = 1$ otherwise.

4.44 Determination of the normal vectors along a force line

Force lines of defined above field H are defined only concerning some function $\vec{n} = \vec{n}(X)$ that associates a normal vector with each space point. During tracking a force line this function should be defined for each point of the iterative process. Let's take into account that at each of these points the plane allowed for the normal vector is known. For the origin point of the force line (it is a point on the boundary of a given base island) it is the plane that is perpendicular to the tangent vector at the point. For each of others points it is the plane that is perpendicular to the tangent line to the force line at this point (because found section of the force line ends at this point, to determine such tangent line the backward derivative is used). Let's define, that at a given point of the iterative process the normal vector is the solution of the following system:

for the origin point:

$$\begin{cases} |\mathrm{H}\vec{n}| = \max \\ |\vec{n}| = 1 \\ \vec{n}\vec{\tau} = 0 \end{cases} \quad \text{(4.44-1a)}$$

where $\vec{\tau}$ is the tangent vector;

for each of other points:

$$\begin{cases} |\mathrm{H}\vec{n}| = \max \\ |\vec{n}| = 1 \\ \vec{n}\vec{p} = 0 \end{cases} \quad \text{(4.44-1b)}$$

where $\vec{p}$ is the vector that is tangent to the trajectory of the force line at the given point.

The solution of each of these systems is found as the maximum of the function $|\vec{f}| = |H\vec{n}(\varphi)|$, where φ is the angular coordinate of vector $\vec{n}$ in a coordinate system in the allowed plane having the given point as the origin (figure F4.44-1).

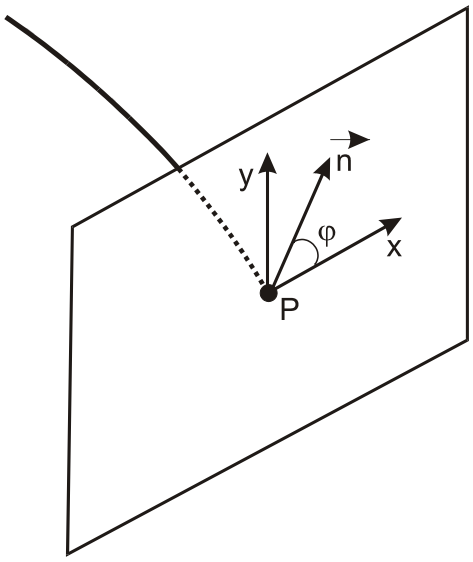


Figure F4.44-1

Thus, during the tracking we simultaneously find both: the function of the trajectory of the force line $X = X(l)$ and the function of distribution of the normal vector along it $\vec{n} = \vec{n}(l)$ (where l is the arc).

Because of the properties of the matrix of operator H the function $\vec{n} = \vec{n}(l)$ can't be continuous. In a point of discontinuity $|\vec{n}(\varphi)| = const$. Therefore, if a given force line is chosen as a bridge, we need to eliminate possible discontinuity points by using an interpolation.

4.45 Definition of the functions $q_1(L)$ and $G(R)$

In accordance with the logic of our task, function $q_1(L)$ should not have essential changes when $L \le L_1$ and $L \ge L_2$ ($L_2 > L_1$). The values of L_1 and L_2 are chosen for the following reasons. Let's consider case of a boundary of a real island. This boundary is represented by a polyline, vertices of which are represented by sampled points. Sources of the field are defined at points on this boundary (in the simplest case - at the vertices, in more details this problem is considered in section 4.46). Let's assume that L_1 should be approximately equal to the average distance between the sources of the field. As a result, the field created by such boundary approaches to the field created by a similar smooth boundary. L_2 defines the radius of the area that is considered at determination of the field tension. Thus, let's assume that:

$$q_1(L) = 1 \text{ in } L \in [0, L_1] \quad \text{(E4.45-1a)}$$

$$q_1(L) = (1 - \frac{L - L_1}{L_2 - L_1})^{\chi} \text{ in } L \in (L_1, L_2) \quad \text{(E4.45-1b)}$$

where χ is a positive factor;

$$q_1(L) = 0 \text{ in } L \in [L_2, \infty) \quad \text{(E4.45-1c)}$$

Analogously, function $G(R)$ should not have essential changes when $R \le R_1$ and $R \ge R_2$ ($R_2 > R_1$). The values R_1 and R_2 are chosen for the following reasons. R_1 should be approximately equal to the average distance between neighbor free points. It provides relatively equal effect of field G, created by a group of free points, on the trajectories of bridges passing trough the region of location of this group (figure F4.45-1).

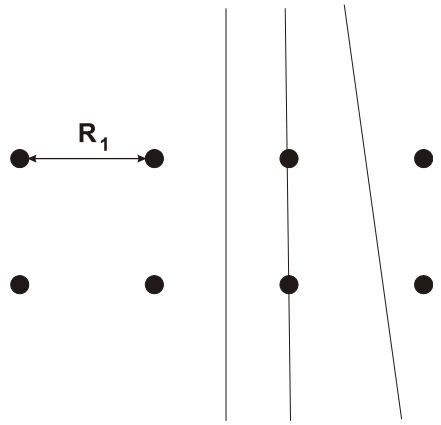


Figure F4.45-1

R_2 should not exceed a value that provides existence in the space region of a given ICADM corresponding to the inner part of the original object a region without action of field G. At the same time, R_2 should be approximately equal to one half of the average distance between islands in a considered area. Fulfillment of these conditions provides a positive effect concerning filtration of topologically incorrect bridges. By analogy with function $q_1(L)$ let's define function $G(R)$:

$$G(R) = G(0) \text{ in } R \in [0, R_1] \quad \text{(E4.45-2a)}$$

$$G(R) = G(0)(1 - \frac{R - R_1}{R_2 - R_1})^{\gamma} \text{ in } R \in (R_1, R_2) \quad \text{(E4.45-2b)}$$

where γ is a positive factor;

$$G(R) = 0 \text{ in } R \in [R_2, \infty) \quad \text{(E4.45-2c)}$$

4.46 Definition of sources of field H

In this section let's use the term "boundary of an island" for the boundary of a hole inside a given island as well. Thus, each island can have one or several boundaries. They are the external boundary and the boundaries of possible holes. At step ICM3→ICM2 as sources of the field we should consider both the external boundaries of given islands so the boundaries of holes which meet the following condition:

Condition C4.46-1. At step ICM3→ICM2 the boundary of a given hole should be considered as a source of the field, if its effective size is greater or equal to the effective size of the smallest island in a L_2-neighborhood of the boundary of the hole. Fulfillment of this condition allows to process situations like shown in (figure FC4.46-1) correctly.

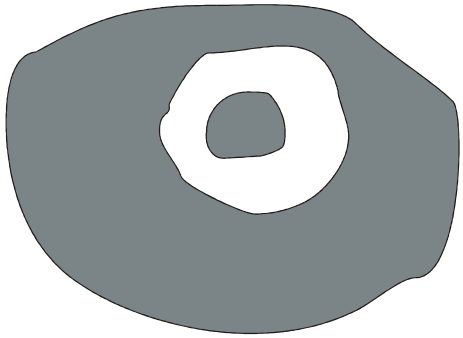


Figure FC4.46-1

It would be ideal, if all the boundaries considered as sources of the field have the same linear density of the "charge". But in practice we deal with a finite set of point sources of the field. For properly modeling uniform charged boundaries such sources should have the same absolute value of the "charge" and approximately the same linear density on the boundaries. Such density should be enough for properly sampling the boundaries.

The total tension of field H at a given space point created by a given (j-th) boundary of a given (i-th) island is defined by the following formula:

$$\mathrm{H}_j^i = h\sum_{k=1}^{n}\mathrm{H}(\overset{\circ}{O}(b), X) \qquad \text{(E4.46-1)}$$

where
n is the total number of sources of the field on the boundary;
h is the "charge" of each of these sources.

Thus, the tension of the field created at a given point by the aggregate of l islands, where each i-th island has m_i boundaries, which are sources of the field, is defined by the following formula:

$$\mathrm{H} = \sum_{i=1}^{l} \sum_{j=1}^{m^i} \mathrm{H}_j^i \qquad \text{(E4.46-2)}$$

where H_j^i is defined by formula E4.46-1.

4.47 Topological issues of a set of bridges

It is obvious, that a set of bridges created for step ICM3→ICM2 should meet the following condition:

Condition C4.47-1. Each section of the boundary of each island should be included in the contour of a created hole.

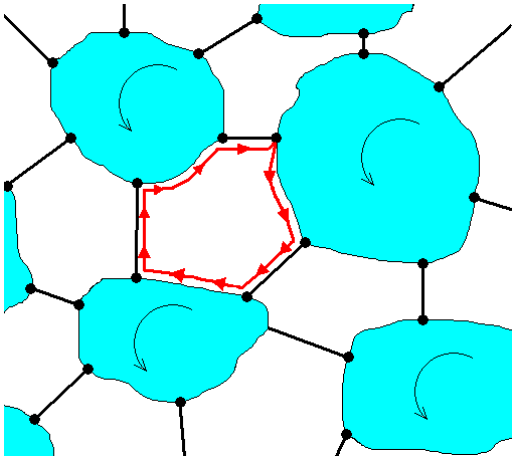


Figure F4.47-1

From a given topology of bridges (figure F4.47-1) we extract holes by the following algorithm:

1. We choose a supporting point of a bridge as the beginning point (P_0) and take the point as the current point (P_1).
2. From P_1 we move along the corresponding bridge to its other supporting point.
3. From this supporting point we move along the boundary of the corresponding island in the positive direction until the first supporting point (P_2) of any bridge is met.

4. If P_2 coincides with P_0, then the passed closed contour is considered as the found boundary of a hole, otherwise we take P_2 as P_1 and return to step 2.

Using this algorithm produces the following condition for topology of created bridges:

Condition C4.47-2. To avoid a topological ambiguity, the supporting points of any two bridges should not coincide with each other.

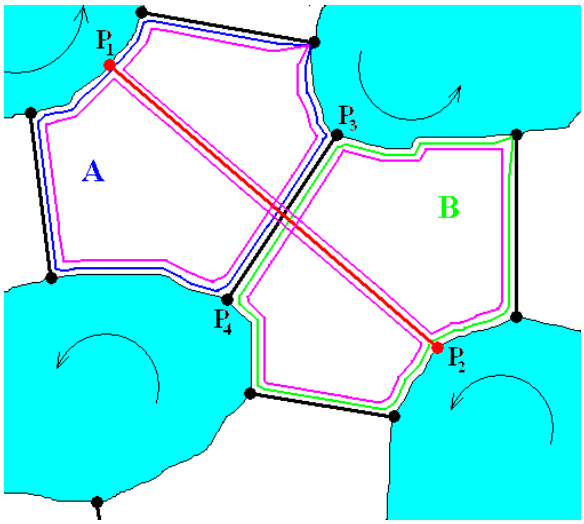


Figure F4.47-2

The proposed method of ICM3→ICM2 reduction is sensitive to appearing of wrong redundant bridges. As it is shown in figure F4.47-2, appearance of a wrong bridge (P_1P_2) leads to appearance of a contour having a complex topology. Such contour can't be interpreted as the boundary of a hole in a reconstructed surface.

Lemma L4.47-1. Let we have a topology created only by correct bridges and the corresponding set of correct contours. Let we add to the topology several wrong bridges with condition, that in each correct contour no more than one wrong bridge appears. Then, a closed contour containing a wrong bridge passes this bridge two times. Besides, if two correct contours connected by a wrong bridge have shared correct bridges, then these bridges are passed two times as well.

Proof. Let's consider a wrong bridge P_1P_2 (figure F4.47-2), which connects two correct contours A and B. The given wrong bridge takes away the trajectory of a round of contour A to contour B. It is obvious, that the trajectory can return to contour A only by the same wrong bridge. Also, to return from contour B the trajectory should return to the supporting point of the wrong bridge on contour B. The trajectory can make it only if it makes the round around all contour B. Similarly arguing, we make the conclusion, that for

closure of the round trajectory at the beginning point the trajectory should make the round around all contour A. Thus, if contours A and B have shared bridges (P_3P_4 in the figure), then these bridges are passed two times as well.

Therefore, if two correct contours don't have shared bridges, then we can simply detect a wrong bridge as a bridge passed two times. If these contours have shared bridges, the situation is more difficult, because we can't distinguish wrong and shared bridges. Taking into account that appearance of a wrong bridge is unacceptable, let's consider that all the bridges passed two (or more) times are removed.

Thus, let's formulate the last condition for topology of bridges:

Condition C4.47-3. The contour of each hole should pass through each bridge included in this contour only once.

Let's note, that the given condition also prevents appearance of islands connected with other islands only by one bridge. Presence of such islands leads to appearance of holes, which are too difficult to process at step ICM2→ICM1.

An algorithm of holes extraction is described below. This algorithm has stability concerning appearance of wrong bridges, if such bridges don't break the condition of lemma L4.47-1. During the holes extraction each segment of the boundary of an island made by supporting points of bridges can have two states - "free" and "used". Initially all the segments are marked "free". If the contour of a hole is extracted successfully, then the segments of the contour are marked "used". The algorithm consists of the following steps:

1. We chose a beginning point for extraction of the contour of a new hole. Initially we choose a "free" segment, then the "positive" (concerning the positive direction for this segment) endpoint of the segment is chosen as the beginning point. If there is no free segment, then the algorithm is ended.
2. We extract the contour of a hole from the chosen beginning point. For extraction the described above algorithm is used. If during the extraction of holes, bridges passed twice are found then these bridges are deleted and a jump to step 1 is made.
3. We mark the segments belonging to the found contour as "used" and then we jump to step 1.

4.48 Determination of the orientation of islands

As was mentioned above, one of the key problems of step ICM3→ICM2 is determination of the orientation of an island. Firstly let's adduce one of the possible approaches.

Let's consider the minimal sphere circumscribed around a given input cloud of points. Let's denote the center of the sphere O (figure F4.48-1). Let A is a point of the surface of an island and B is the point of intersection of the sphere

and half-line OA. Let's consider the direction of normal at A making a positive dot product with vector $\overrightarrow{OA}$ as the external direction if the following conditions are fulfilled:

- segment AB doesn't intersect the surface of any other island;
- there are no free points in the r neighborhood of segment AB; where $r = kd$, d is the average distance between neighbor free points in the corresponding given region, k is a constant.

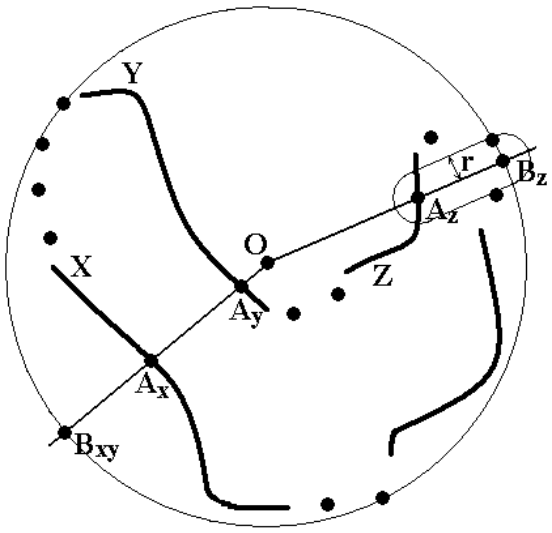


Figure F4.48-1

If a point, that meets these conditions, can be found for a given island, then its orientation can be determined by this method. However, this method (or any other simple method) isn't suitable for all cases. For example, in figure 4.48-1 the described method works for island X and doesn't work for islands Y and Z.

Let's have such "difficult" island in an environment of correctly oriented islands (let's call them *standard islands*). To determine the orientation of the given island the properties of field H can be used. Let's consider the boundaries of the standard islands as sources of field H and the given island as the base island. From the properties of field H follows that in case of correct orientation of the base island, points on its boundary are attracted to neighbor segments of the boundaries of the standard islands and are repulsed otherwise. Let's calculate the following integral for the both possible variants of the base island orientation:

$$\oint_B \vec{f}\vec{m}\, db \qquad \text{(E4.48-1)}$$

where
B is the contour of the boundary of the island;
$\vec{f}$ is the force vector at point (X) of the boundary;

$\vec{m}$ is the unit vector (figure 4.48-2) at point X, that is:

1) perpendicular to the normal (n) to the island surface at point X;
2) perpendicular to the tangent line (p) to the boundary at point X;
3) directed to the non-reconstructed area.

The greatest value of integral E4.48-1 corresponds to the correct orientation of the island.

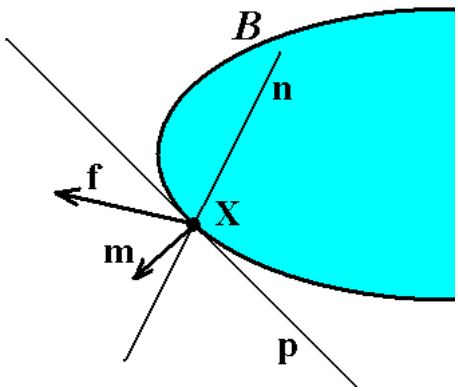


Figure F4.48-2

4.5 Properties of determination of a set of bridges at step ICM2→ICM1

At this step decomposition of a given complex hole is made by recursive binary subdivision of the hole until each resulting hole is a simple hole. The subdivision is made by construction of a bridge inside the hole contour. To track the corresponding force line, the boundary point with one of the highest values of the tension of the field is chosen as the origin. In general, calculation of the tension of field H is made in the same way as in step ICM3→ICM2, but there are several differences described below.

Each hole at step ICM2→ICM1 is considered separately. Hence, the positive direction of its contour can be chosen arbitrarily. To provide the biggest "convexity" for the boundaries of the child holes, in definition of field H the following changes are made. Let's assume, that the field affects a complex of the type $\overset{\circ\circ}{X}\{\overset{\circ}{X}\{X,\vec{\tau}^{X}\},\vec{n}^{X}\}$, and the tension of the field is defined for a complex of the type $\overset{\circ}{X}\{X,\vec{\tau}^{X}\}$; where X is a space point, $\vec{\tau}^{X}$, $\vec{n}^{X}$ are unit vectors (at step ICM3→ICM2 the field affects a complex of the type $\overset{\circ}{X}\{X,\vec{n}^{X}\}$ and the tension is defined for a space point (X)). During subdivision of a given hole we need to determine the field tension in two cases. The first one is when a considered point is on the boundary of the hole. The second case occurs when a considered point is on a tracked force line that has the given origin point on the boundary. In both cases we need to define a complex of the type adduced above at the considered point. In the first case let's define the necessary

complex assigning to the given point the tangent vector at the point. In the second case let's define the complex assigning to the given point the tangent vector at the origin point of the corresponding force line. By analogy with section 4.43 let's denote a currently considered source of the field $\overset{\circ\circ}{O}\{\overset{\circ}{O}\{O,\vec{n}^{O},\vec{\tau}^{O}\},h^{O}\}$. All the considerations and equations of section 4.43 in our case are valid with condition of replacement of "charge" h^{O} by an "effective charge" σ^{XO}. For a given source of the field $\overset{\circ\circ}{O}$ concerning a given point $\overset{\circ}{X}$ the "effective charge" is defined by the following formula:

$$\sigma^{XO} = -(\vec{\tau}^{X}\vec{\tau}^{O})h^{O} \qquad \text{(E4.5-1)}$$

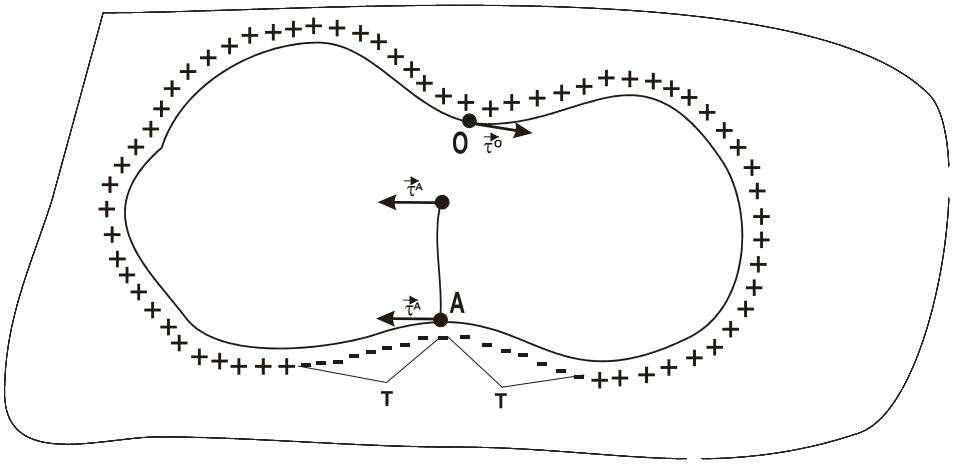


Figure F4.5-1

At determination the tension of the field at a given point on the boundary (on a current tracked force line) we consider the sign of "charge" h of a current considered source as negative, if the length of the section of the boundary between the source point and the boundary point (the origin point of the force line) doesn't exceed an established threshold value (let's denote it T). Otherwise the sign of the "charge" is considered positive (figure F4.5-1). Thus the tracked force line can't end at a point that connected with the origin point by the segment of the boundary shorter than T. It limits the minimal possible length of the boundary of at least one child hole by the value $T + L$; where L is the length of the splitting bridge.

4.6 Taking into account free points

In accordance with the concept described in section 4.31, after determination of the required bridges by tracking the corresponding force lines we need to take into account free points like perturbing factors for these trajectories.

Let's assume, that free points are sources of the defined in subsection 4.424 field G. Let's consider a bridge (figure F4.6-1). Let's choose a point $X\{\overset{\circ}{X}, \vec{n}\}$ on the bridge. Let's consider the line n that contains X and external normal vector $\vec{n}$. Let's consider the centered at X cylinder with axis n, radius R_2, and the half-height kR_2, where k is a positive factor. Let there are m free points (let's denote them P_i, $i = 1, m$) inside the cylinder. For each point P_i let's consider its projection on n, let's denote such projection P_i' and call a *pole*.

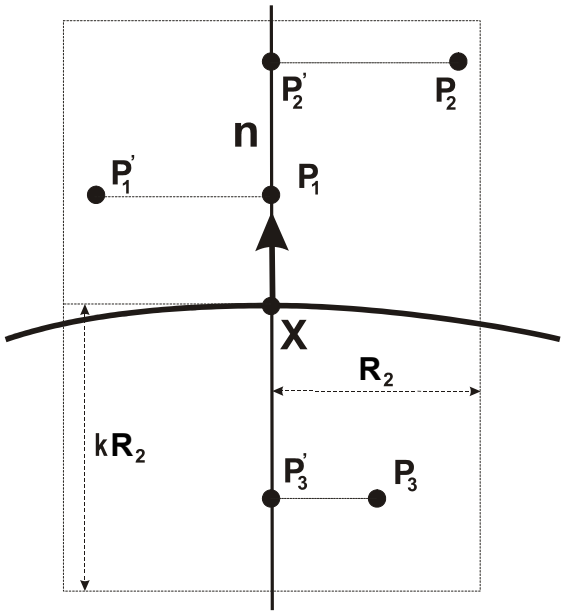


Figure F4.6-1

Let's associate with each pole P_i' the weight w_i that is equal to the tension of field G created by the corresponding free point P_i' at the given pole:

$$w_i = G(|P_i - P_i'|) \quad \text{(E4.6-1)}$$

Let's assume, that the found perturbed point corresponding point X (let's denote it X') lies on n. Let's assume that point X' is connected with the poles by springs, the stiffness of each spring is equal to the weight of the corresponding pole. The potential energy of the spring connecting X' with a given (i-th) pole is defined by the formula:

$$E_i = w_i \,|\, X' - P_i' \,| \quad \text{(E4.6-2)}$$

Let's assume that the coordinates of X' minimize the sum of the squares of the potential energies of the given springs. In this case these coordinates are defined by the following formula:

$$X' = \frac{\sum_{i=1}^{m} w_i^2 P_i'}{\sum_{i=1}^{m} w_i^2} \qquad \text{(E4.6-3)}$$

Condition C4.6-1. The trajectory of a bridge changed by taking into account free points should meet the following conditions:

$$\lim_{l \to 0} X'(l) = A \qquad \text{(EC4.6-1a)}$$

$$\lim_{l \to L} X'(l) = B \qquad \text{(EC4.6-1b)}$$

where
A, B are the supporting points of the bridge;
$X' = X'(l)$ is the parametric equation of the trajectory.

To provide fulfillment of condition C4.6-1 let's assume that the supporting points (A, B) of a bridge are sources of field G as well. Let's define the values of weights of poles corresponding to the supporting points by the following formulas:

$$w_A = kG(|A - A'|) \qquad \text{(E4.6-4a)}$$

$$w_B = kG(|B - B'|) \qquad \text{(E4.6-4b)}$$

where k is a constant.

4.7 Experimental implementation

The theoretical concepts described above have been verified by an experimental implementation. In comparison with the theoretical description the current implementation has the following properties and differences.

At step ICM3→ICM2 we consider, that the boundary of an island chosen as the base island isn't a source of field H. These changes have been made for the following two reasons:

- Necessity to take into account the field created by remote sources adequately, because as a result of restricted precision of floating point operations the contribution of such sources can be distorted or masked by the field created by nearby sources.
- Necessity to reduce the cost of determination of the field tension, because at calculation the tension at a point of the boundary of the base island (or at a point that is close to the boundary) a majority of sources of the field in the L_2-neighborhood of the point belongs to the same boundary. At the same time the influence of such sources on the

trajectory of a tracked force line is not so significant in comparison with the influence of sources belonging to other islands.

For the same reasons at step ICM2→ICM1 sources of the field having in accordance with paragraph 4.5 negative “charge” aren’t considered.

As a “darning” method (introduced in D4.1-3) the simplest method has been chosen. We just calculate the point of the geometrical center of the boundary of a considered hole, and then we make a “triangle fan” inside the hole using the obtained central point. Concerning this method a hole should have the following properties to be simple:

- the boundary of the hole should not be too long (in the implementation the threshold length is 12 edges);
- the boundary of the hole should have a simple projection on its average plane, and this projection should be a convex figure.

Since decomposition of a complex hole by the method described in paragraph 4.5 leads to appearance of child holes, meeting the last condition, to discover whether this hole is simple or not only the first condition is checked. During the tests we have not found any problems caused by this simplification.

The choice of the given “darning” method means that for reduction an ICADM of class ICM3(2) to an ICADM of class ICM1 we need to construct the greatest number of bridges in comparison with other more robust “darning” methods (like projection-based, spline-based, etc). This fact allows interpreting the obtained cost indices as the worst case for tested ICADMs. In case of using more powerful “darning” methods we need to construct a significantly lesser number of bridges, and the cost indices should be better.

4.8 Results and conclusions

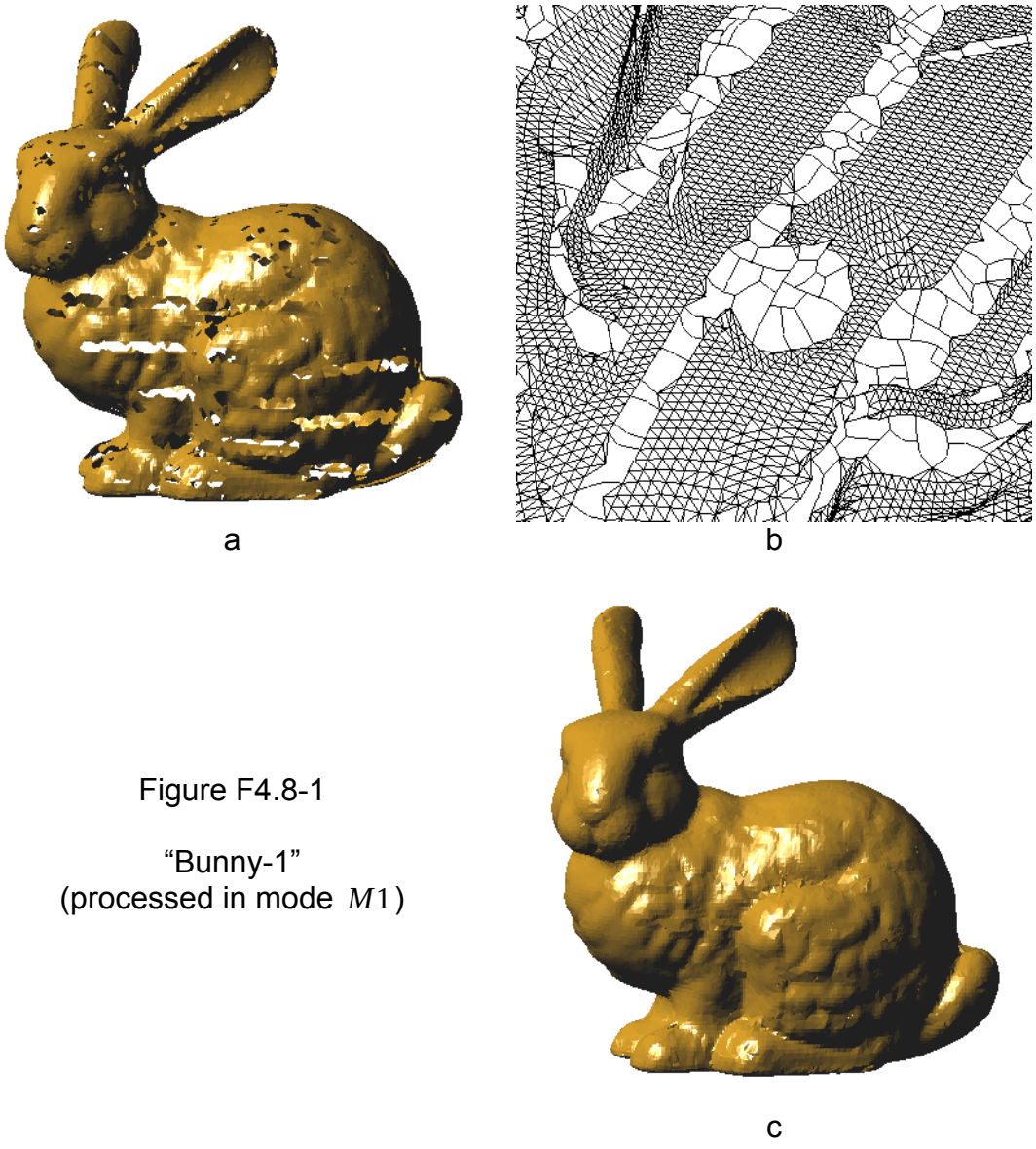
The created implementation has been applied to repairing several ICADMs. The testing has been made for each of the two following modes of work of the algorithm: “light” mode (let’s denote it $M1$) and “full” mode (let’s denote it $M2$). The “light” mode doesn’t use the shielding condition (defined by E4.43-6) and doesn’t take into account the influence of free points on the length of a bridge (introduced in subsection 4.424).

For creation of the tested ICADMs the following method has been used. A cloud of points representing one of “classic” model (such models are widely used for testing by various authors) is chosen as an input. Then this cloud is artificially damaged by removing sampled points from selected regions. The density and uniformity of sampled points in these regions is reduced to the values, which are typical for regions shaded during scanning. Then for such cloud a surface reconstruction algorithm is applied.

Here the results for four such ICADMs are adduced. The visual results are shown in the figures F4.8-1, F4.8-2, F4.8-3, and F4.8-4. For each one the figure

with letter “a” shows the input ICADM, the figure with letter “b” shows a fragment of the set of constructed bridges, and the figure with letter “c” shows the finally obtained CAD-model.

In figure F4.8-2 along with examples of successful application of the concept of perturbed BBSR to reconstruct surface inside complex holes a defect that can’t be fixed within the framework of this concept is marked by circle. Note, that incompleteness of the given ICADM in the corresponding region doesn’t allow considering this region as one having a practical value.



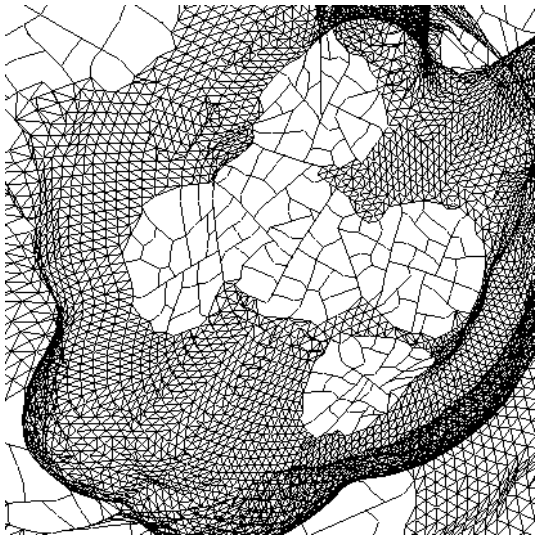
a b

c

Figure F4.8-1

“Bunny-1”
(processed in mode $M1$)

a



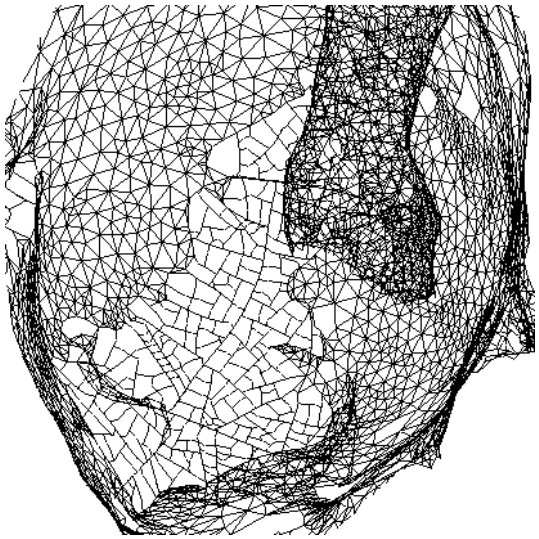
b

c

Figure F4.8-2

"Bunny-2"
(processed in mode $M2$)

a



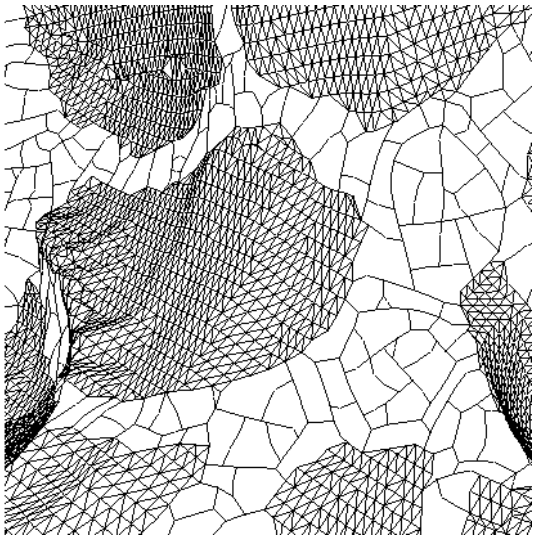
b

c

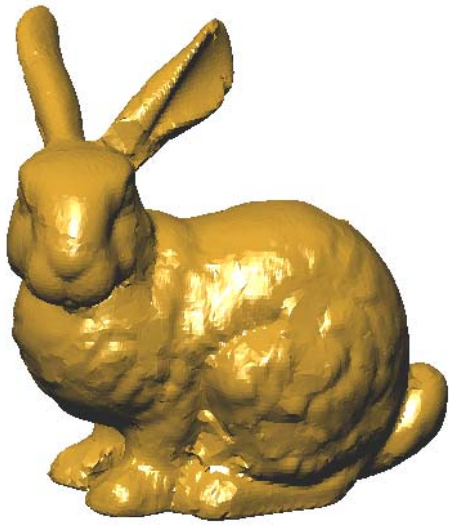
Figure F4.8-3

"Woman-1"
(processed in mode $M1$)

a



b



c

Figure F4.8-4

"Bunny-3"
(processed in mode $M2$)

a b

Figure F4.8-5

In figure F4.8-5 an example (a fragment of model “Woman-1”) of influence of the factor of interpolation (k_i) introduced in E4.427-2 is shown. F4.8-5a shows the result for $k_i = 0$ and F4.8-5b shows the result for $k_i = 1$.

The numerical results of the tests are shown in table T4.8-1. In this table the following denotations are used:

N_{bnd} is the total number of boundary points;

N_{bdg} is the total number of iterative points of constructed bridges;

for each processing mode the corresponding time (denoted by t) of the reconstruction is adduced; for processing in mode $M1$ there is an additional information field (denoted by “Art.”) about appearing any artifacts in comparison with the result of processing in mode $M2$;

χ, γ, k_i are the factors introduced in section 4.45 and subsection 4.427 respectively, which have influence on behavior of force lines of the field.

These results have been obtained on a PC with a 500MHz Pentium-3 CPU.

ICADM	Class	N_{bnd}	N_{bdg}	$M1$		$M2$	χ	γ	k_i
				t,s	Art.	t,s			
“Bunny-1”	ICM2	2424	749	11	No	30	2.5	1.0	0
“Bunny-2”	ICM2	2319	3615	16	Yes	66	2.5	1.0	0
“Woman-1”	ICM2	1526	7938	34	No	283	1.7	1.0	1.0
“Bunny-3”	ICM3	3272	4150	48	Yes	185	2.5	1.0	0

Table T4.8-1

For model “Woman-1“ the smallest step of the tracking a force line has been chosen, it explains the greatest value of N_{bdg} for this model. The artifacts that have been found for mode $M1$ for the ICADMs “Bunny-2” and “Bunny-3” are shown in the figures F4.8-6 and F4.8-7 respectively.

Figure F4.8-6

Figure F4.8-7

In general, the cost of the algorithm mainly depends on properties of input topology of boundaries and distribution of free points. Therefore behavior of the cost of the algorithm can’t be expressed by a simple “cost equation”. Very roughly the cost can be estimated by the following expression:

$$C \sim kR^{\alpha+1}L(\frac{S_{\bar{\mathrm{A}}}}{S_{smp}})^{\beta} \qquad \text{(E4.8 -1)}$$

where

R is the typical remoteness of a boundary from neighbor boundaries;

L is the total length of the boundaries;

$S_{\bar{\mathrm{A}}}$ is the total area of $\bar{\mathrm{A}}$;

S_{smp} is the typical area of a simple hole;

$\alpha \in [1,3]$ but in practice α doesn’t essentially differ from 1, the part “$+1$” expresses the cost of calculation of the shielding factor δ defined by E4.43-6;

$\beta \in [0,\frac{1}{2}]$ with dependence on the topology;

k is a constant.

A comparison with the warping-based methods (mentioned in paragraph 4.2) has been made analytically, because the cost of solution of the used type of differential equations is known and information about the number of iterations is adduced in the corresponding works. The following two cases of the modeling

have been considered: relatively simple topology, when only local modeling of the membrane (mesh) in regions corresponding to $\overline{A}$ is required; complicated case, when to obtain a proper result the modeling entire membrane is needed. During the analysis it has been supposed, that the warping process use at the beginning step a “magic” approach that can construct a proper initial shape (that requires about 50 iterations to get satisfactory approximation) without a cost. Even in this case the proposed method shows convincing theoretical advantage over the warping in the first case while working in mode $M1$ and in the second case while working in mode $M2$.

Thus, the presented method substantially meets the requirements formulated in paragraph 4.2. As its disadvantages the high cost and sensitivity to precision of floating point operations can be mentioned. In general, it seems, a method having a low cost can’t be created for the considered problem. At the same time the described method has very good capabilities for parallelization, owing to the fact that the processes of determination of the field tension at points are absolutely independent. And the processes of determination of contributions of considered field sources to the total field tension at a given point are independent as well.

5. GENERAL CONCLUSION AND FUTURE WORK

In general, the new methods proposed for solution of the problems formulated in paragraph 1.1 confirmed the expected properties. In addition to the results adduced in the corresponding parts of the chapters 3 and 4 let's note, that the described in the given work methods have proved the capability to make a self-dependent surface reconstruction system within the framework of CCAA. For example, the results for the models "Bunny-1", "Bunny-2", "Bunny-3" adduced in paragraph 4.8 have been obtained in the following way. To realize step PointCloud->ICADM the algorithms described in the paragraphs 3.3, 3.4 have been used as the start algorithm and as the filtering algorithm respectively. And final processing of the obtained ICADMs has been made by the method, described in chapter 4.

To designate the field of future work, let's note that the natural field of use of the designed methods is a joint usage (possible with another surface reconstruction algorithm) for processing big clouds of points having problems of sampling. But the current implementation of the method described in chapter 4 can't work with a large input. Thus, as a future work a further improvement of both, the principles and the implementation of the method is planned to create a system for processing big damaged clouds of points that can be used in practice.

References

[ACD00] Amenta N., Choi S., Dey T., Leekha N. A simple algorithm for homeomorphic surface reconstruction. In Proc. of 16th Sympos. Comput. Geom., 2000, pp. 213-222.

[AD97] Attali D. r-regular shape reconstruction from unorganized points. In ACM Symposium on Computational Geometry, Nice, France, 1997, pp. 248-253.

[AGJ02] Adamy U., Giesen J., John M. Surface Reconstruction using Umbrella Filters. Computational Geometry Theory & Applications 21(1-2), pp. 63-86, 2002.

[AS95] Algorri M-E., Schmitt F. Deformable Models for Reconstructing 3d Data. CVRMed 95, Ed. Ayache.

[AS96] Algorri M-E., Schmitt F. Surface reconstruction from unstructured 3d data. Computer Graphics forum, 15(1), pp. 47-60, 1996.

[BB97] Bernardini F., Bajaj C. Sampling and reconstructing manifolds using alpha-shapes. In Proc. of the Ninth Canadian Conference on Computational Geometry, August 1997, pp. 193-198.

[BBC97] Bernardini F., Bajaj C., Chen J., Schikore D. Automatic reconstruction of 3d cad models from digital scans. Technical Report CSD-97-012, Department of Computer Sciences, Purdue University, 1997.

[BBX95] Bajaj C., Bernardini F., Xu G. Automatic reconstruction of surfaces and scalar fields from 3d scans. Computer Graphics Proceedings, SIGGRAPH'95, Annual Conference Series, pp. 109-118, 1995.

[BBX97] Bajaj C., Bernardini F., Xu G. Reconstructing surfaces and functions on surfaces from unorganized 3d data. Algorithmica, 19, pp. 243-261, 1997.

[BE95] Bern M., Eppstein D. Mesh Generation and Optimal Triangulation. Computing in Euclidean Geometry, D.-Z. Du and F.K. Hwang, eds., World Scientific, 1992, 2nd edition, 1995.

[BH93] Baader A., Hirzinger G. Three-dimensional surface reconstruction based on a self-organizing feature map. In Proc. 6th Int. Conf. Advan. Robotics, Tokyo, 1993, pp. 273-278.

[BH94] Baader A., Hirzinger G. A self-organizing algorithm for multisensory surface reconstruction. In International Conf. on Robotics and Intelligent Systems IROS'94, Munich, Germany, September 1994.

[BJ84] Boissonnat J-D. Geometric structures for three-dimensional shape representation. ACM Transactions on Graphics, 3(4), pp. 266-286, 1984.

[BS96] Bajaj C., Schikore D. Error-bounded reduction of triangle meshes with multivariate data. In Proceedings of SPIE Symposium on Visual Data Exploration and Analysis III, January 1996, pp. 34-45.

[BTG95] Bittar E., Tsingos N., Gascuel M-P. Automatic reconstruction of unstructured data: Combining a medial axis and implicit surfaces. Computer Graphics forum, 14(3), pp. 457-468, 1995. Proceedings of EUROGRAPHICS'95.

[DG01] Dey T., Giesen J. Detecting undersampling in surface reconstruction. Proc. 17th Sympos. Comput. Geom., 2001, pp. 257-263.

[DH94] Delingette H. Simplex Meshes: a General Representation for 3D Shape Reconstruction. INRIA, Research Report No. 2214.

[DRH73] Duda R., Hart P. Pattern Classification and Scene Analysis. Wiley and Sons, Inc., 1973.

[EH92] Edelsbrunner H. Weighted alpha shapes. Technical Report UIUCDCS-R-92-1760, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1992.

[EPM92] Edelsbrunner H., Mucke E. Three-dimensional alpha shapes. ACM Transactions on Graphics, 13(1), pp. 43-72, 1994.

[ES02] Emelyanov A., Skala V. Surface reconstruction from problem point cloud. In Proc. of Graphicon’02, Nizhny Novgorod, September 2002, pp.31-37.

[GK02] Gopi M., Krishnan S. A Fast and Efficient Projection-Based Approach for Surface Reconstruction. SIBGRAPI 2002, Fortaleza, Brazil, October 2002.

[GMW97] Guo B., Menon J., Willette B. Surface reconstruction using alpha-shapes. Computer Graphics forum, 16(4), pp. 177-190, 1997.

[HDD92] Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W. Surface reconstruction from unorganized points. Computer Graphics, 26(2), pp. 71-78, 1992. Proceedings of Siggraph'92.

[HDD93] Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W. Mesh optimization. In Computer Graphics Proceedings, Annual Conference Series, pp. 21-26, 1993. Proceedings of Siggraph'93.

[HG92] Huang W-C., Goldgof D. Sampling and Surface Reconstruction

with Adaptive-Size Meshes. Proc. SPIE Appl. of Art. Intell. 1798, pp. 760-770, 1992.

[HH94] Hoppe H. Surface Reconstruction from Unorganized Points. PhD thesis, Univ. of Washington, Seattle WA, 1994.

[HJL93] Hoschek J., Lasser D. Fundamentals of Computer Aided Geometric Design. A.K. Peters, 1993.

[HL79] Herman G., Liu H. Three-dimensional displays of human organs from computed tomograms. Computer Graphics and Image Processing, 9, pp. 1-21, 1979.

[IBS97] Isselhard F., Brunnett G., Schreiber T. Polyhedral reconstruction of 3d objects by tetrahedra removal. Technical report, Fachbereich Informatik, University of Kaiserslautern, Germany, February 1997. Internal Report No. 288/97.

[LC87] Lorensen W., Cline H. Marching cubes: A high resolution 3d surface construction algorithm. Computer Graphics, 21(4), pp. 163-169, 1987.

[MBL91] Miller J., Breen D., Lorensen W., O'Bara R., Wozny M. Geometrically deformed models: A method for extracting closed geometric models from volume data. Computer Graphics, pp. 217-226, 1991, Proceedings of SIGGRAPH'91.

[ME93] Mucke E. Shapes and implementations in three-dimensional geometry. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, September 1993.

[MM97] Mencl R., Muller H. Graph-based surface reconstruction using structures in scattered point sets. Research Reports No. 661, 662, Fachbereich Informatik, Lehrstuhl VII, University of Dortmund, Germany, 1997.

[MM98] Mencl R., Muller H. Interpolation and Approximation of Surfaces from Three-Dimensional Scattered Data Points. State of The Art Report, EUROGRAPHICS'98, pp. 51-67.

[MS91] Muraki S. Volumetric shape description of range data using “blobby model”. Computer Graphics, pp. 217-226, 1991. Proceedings of SIGGRAPH'91.

[PS85] Preparata F., Shamos M. Computational Geometry: An Introduction. Springer Verlag, 1985.

[RA00] Aldasoro C., Algorri M-E. A combined algorithm for image segmentation using neural networks and 3D surface reconstruction using dynamic meshes. Revista Mexicana de Ingeniería

Biomédica, XXI(3), pp. 73–81, 2000.

[RGW97] Roth G., Wibowoo E. An efficient volumetric method for building closed triangular meshes from 3d image and point data. In Graphics Interface'97, pp. 173-180, 1997.

[RM95] Ruprecht D., Muller H. Spatial free form deformation with scattered data interpolation methods. Computers and Graphics 19, pp. 63-71, 1995.

[SB97] Schreiber T., Brunnett G. Approximating 3d objects from measured points. In Proceedings of 30th ISATA, Florence, Italy, 1997.

[SRT92] Szeliski R., Tonnesen D. Surface modeling with oriented particle systems. Computer Graphics, 26, pp. 185-194, 1992. Proceedings of Siggraph'92.

[ST97] Schreiber T. Approximation of 3d objects. In Proceedings of the 3rd Conference on Geometric Modeling, Dagstuhl, Germany, 1997.

[TM91] Terzopoulos D., Metaxas D. Dynamic 3d models with local and global deformations: Deformable superquadrics. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(7), pp. 703-714, 1991.

[TWK88] Terzopoulos D., Witkin A., Kass M. Constraints on deformable models: Recovering 3d shape and nonrigid motion. Artificial Intelligence, 36, pp. 91-123, 1988.

[VR94] Veltkamp R. Closed object boundaries from scattered points. In Lecture Notes in Computer Science 885. Springer Verlag, 1994.

[VR95] Veltkamp R. Boundaries through scattered points of unknown density. Graphics Models and Image Processing, 57(6), pp. 441-452, 1995.

[VT92] Vasilescu M., Terzopoulos D. Adaptive Meshes and Shells: Irregular Triangulation, Discontinuities, and Hierarchical Subdivision. Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR'92), Champaign, IL, June, 1992, pp. 829–832.

[WF97] Weller F. Stability of voronoi neighborship under perturbations of the sites. In Proceedings 9th Canadian Conference on Computational Geometry, Kingston, Ontario, Canada, August 1997.

Publications

Emelyanov A., Skala V. Surface reconstruction from problem point cloud. In Proc. of Graphicon'02, pp.31-37.

Emelyanov A., Skala V. An algorithm for problem point cloud processing. In Proc. of VEonPC'02, pp.68-76.

Emelyanov A. Robust Surface Reconstruction Strategy for Large Clouds of Points. Technical Report DCSE/TR-2003-07, Department of Computer Science and Engineering, University of West Bohemia in Plzen, 2003.

Emelyanov A., Skala V. A Method for Repairing Triangulations. Submitted to Graphicon'04.