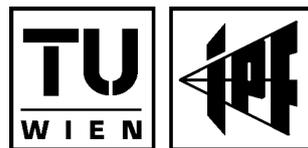


## 3D Terrain Models on the Basis of a Triangulation

von Norbert Pfeifer

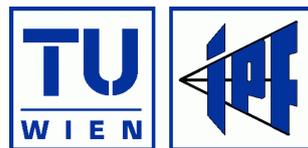


Veröffentlichung des Instituts  
für Photogrammetrie und Fernerkundung  
mit einer Förderung durch



## 3D Terrain Models on the Basis of a Triangulation

von Norbert Pfeifer



Veröffentlichung des Instituts  
für Photogrammetrie und Fernerkundung  
mit einer Förderung durch



Herausgeber und Verleger: o.Prof. Dr.-Ing. Karl Kraus  
Vorstand des Instituts für Photogrammetrie und Fernerkundung  
der Technischen Universität Wien  
A-1040 Wien, Gußhausstraße 27-29

Die Kosten für den Druck wurden aus eigenen Einnahmen des Instituts für  
Photogrammetrie und Fernerkundung der Technischen Universität Wien  
und mit Unterstützung der Erste Österreichische Sparkasse-Bank getragen.

Diese Arbeit wurde an der Fakultät für Technische Naturwissenschaften und Informatik  
der Technischen Universität Wien (Karlsplatz 13, A-1040 Wien, Österreich)  
zur Erlangung des akademischen Grades eines Doktors der Technischen Wissenschaften eingereicht.

Begutachter:

o.Prof. Dr.-Ing. Karl Kraus  
Institut für Photogrammetrie und Fernerkundung

o.Prof. Dr.techn. Helmut Pottmann  
Institut für Geometrie

Technischen Universität Wien  
A-1040 Wien, Karlsplatz 13

Tag der mündlichen Prüfung: 31. Oktober 2002

Druck: die kopie, A-1040 Wien  
Auflage: 250 Stück  
ISBN: 3-9500791-7-3

Diese Arbeit ist Teil des Forschungsprojekts „Dreidimensionales Topographisches Informationssystem“ und wurde vom Fonds zur Förderung der wissenschaftlichen Forschung (FWF) unter Projekt-Nr. 14083-MAT unterstützt.

This work is part of the research project “Three Dimensional Topographic Information System” and was supported by the Austrian Science Foundation under project no. 14083-MAT.

## Abstract

This work provides an overview on terrain modelling techniques. Terrain models, or in order to be more general, topographic surface models, play an important role in many fields of science and practice where a relation to a location, i.e. a 'geo-relation' is given. These models describe the height as a function of the location. There lies a restriction in this definition, because only *one* height is allowed at one ground-plane position. Therefore, the currently used models are often termed 2.5D terrain models. The modelling of overhangs is not possible within such an approach. The aim of this work is to put aside this limitation and provide methods for 3D terrain modelling where not only the above restrictions do not apply anymore, but also more general surfaces with tunnels and cave systems can be reconstructed. Another terrain property which plays an important role in this work is its smoothness: a model shall be smooth. An exception is introduced at so-called breaklines where the terrain shape has a sharp edge.

There are several ways in order to build terrain models with the above characteristics (fully 3D and smooth). In this work, emphasis is put on those approaches which reconstruct the surface on the basis of a triangulation. Two different techniques are treated with great detail: the patch work and the subdivision approach. For each of those two, one method was developed which considers the special requirements in terrain modelling. The main contribution of this work to terrain modelling are those new methods. Generation, improvement, and thinning of triangulations is not treated within this work, but references to the relevant literature are given.

Generally, the reconstruction of a *patch work* proceeds as follows. Given is a triangulation, which has – as expected – planar faces. For each edge a curve is determined which interpolates the end points. In the next step, triangular patches are inserted into a triple of boundary curves spanned over the edges of each triangle. As the patches interpolate the boundary curve a G0 surface (a geometrically continuous surface) is obtained. However, this is not enough, because a smooth surface (G1, geometric continuity of order one, i.e. tangent plane continuity) is desired. Adjacent patches must therefore interpolate not only the boundary curves, but also share a common field of cross boundary derivatives. This is the general approach for patch work surfaces.

The patch work method which is proposed in this work<sup>1</sup> starts with an enhancement of the triangulation. As the measurement of terrain points and lines is always burdened with random errors (depending on the measurement device characteristics) these errors should be removed first. This can be achieved by kriging, whereby for each point of the triangulation (i.e. each vertex) a filter value is determined from its neighboring points. In this step also the surface normal vectors in the points can be estimated, but alternative methods for the estimation of the normal vector, e.g. by averaging those of the triangles which are incident to that vertex, are possible, too. Now, not only the position, but also the surface normal vector is prescribed for each vertex. The patches which are to be reconstructed over each face of the triangulation shall be polynomials of degree four and they are described with Beziér triangles which allow a geometric interpretation of the coefficients of the (bivariate) polynomial. In the next step, boundary curves of polynomial degree three are computed which 'replace' the edges of the triangulation. These curves interpolate the end points of the edge and the curve tangents in those points are perpendicular to the estimated normal vectors. This determines the boundaries of each patch. The missing parameters (i.e. coefficients of the polynomial) influence the shape in the interior of the patch and also the tangent planes of the patch along the boundaries. A field of normal vectors is estimated for each boundary curve by blending the normal vectors from the end points into each other. The 'inner' parameters of a patch are now determined in a way that the normal vector fields are approximately perpendicular to the tangent planes of the patch along the boundaries in a least squares sense. As this field is 'only' approximated and not interpolated this scheme is called  $\epsilon$ G1 (i.e. approximately tangent plane continuous).

The second technique for surface reconstruction over a triangulation is the so-called *subdivision*. In this approach the given triangulation is refined in steps, and in each step new vertices and edges are inserted into the triangulation. This is performed in a way that the smoothness of the triangulation is increased in each level, the angles between adjacent triangles converge towards 180°. The limit surface, reached after an infinite number of subdivision steps, is smooth. An advantage of this approach is that the surface description is always composed of small triangles which allows to apply simple algorithms for intersections and similar tasks. The

---

<sup>1</sup>It has to be mentioned that this method was partly developed in my Diploma Thesis – written in German –, though it is not embedded in the context of terrain modelling there.

size of the triangles depends on the number of subdivision steps (i.e. the refinement level). This is the general approach for subdivision surfaces.

Also in the reconstruction technique (developed in this work) for topographic surfaces which is based on subdivision a removal of random measurement errors has to be performed first. The refinement rule applied here is the so-called edge midpoint subdivision where in one step one vertex is inserted into each edge and the triangulation is updated. The subdivision is based on the estimation of local surfaces in each vertex. A local surface is estimated which approximates the vertex of interest and its neighbors. The co-ordinates of the new points are obtained by averaging the two local surfaces in either edge end point. To achieve this, a point, representative for the edge midpoint, is computed on both local surfaces and the mean of these two is the new point. Also the 'old' points obtain new co-ordinates, namely their position on the local approximating surfaces. Special modifications are introduced in order to interpolate the originally given points.

The approaches are compared to each other with examples based on real photogrammetric and geodetic observations as well as on synthetic terrain data. It turns out that the surfaces obtained by the developed subdivision approach meet the requirements in topographic terrain modelling better.

## Kurzfassung

Der Titel der vorliegenden Arbeit ist „3D-Geländemodelle auf Basis einer Triangulierung“. Digitale Geländemodelle (abgekürzt DGM), also Beschreibungen von Höhe und Lage der Erdoberfläche in einer Form, die für die Bearbeitung auf Computern geeignet ist, werden in vielen Gebieten der Wissenschaft und Praxis erfolgreich eingesetzt. Die Anwendungen reichen von der Ableitung von Höhenschichtlinien für topographische Karten bis zur Modellierung des Wasserabflusses nach einem Unwetterereignis, um nur zwei zu nennen. In vielen geographischen Informationssystemen sind Geländemodelle ein unverzichtbarer Bestandteil. Die dort verwendeten Modelle unterliegen aber einer Einschränkung. Mathematisch formuliert sind sie Graphen bivariater Funktionen. Mehr von einer praktischen Seite beleuchtet heißt das, dass die Modellierung von Überhängen unmöglich ist, von steilen Wänden und Klippen nur sehr unzureichend möglich ist, und Höhlensysteme und Brücken ebenso nicht modelliert werden können. Um dieses Charakteristikum zum Ausdruck zu bringen werden solche Modelle oft als „2.5D“ bezeichnet. In dieser Arbeit werden Methoden zur Ableitung von Geländemodellen vorgestellt, die diesen Beschränkungen nicht unterliegen. Ein weiterer wichtiger Aspekt ist die Glattheit des Modells. Das uns umgebende Gelände ist im Allgemeinen glatt, mit Ausnahme der sogenannten Geländekanten, und daher soll auch ein DGM davon glatt sein. Es gibt verschiedene Möglichkeiten glatte 3D-Geländemodelle zu erzeugen, aber diese Arbeit ist auf jene Methoden beschränkt, die mittels einer Triangulierung erstellt werden. Die Erzeugung, Verbesserung und Ausdünnung von Triangulierungen wird nicht behandelt, aber Verweise auf die Literatur sind angegeben.

Zwei verschiedene Ansätze um aus einer Triangulierung eine glatte Oberfläche zu erzeugen werden vorgestellt: Flächenverbände mit parametrischen Patches und Subdivision (Unterteilungsflächen). Im Rahmen der beiden Zugänge ist jeweils eine Methode entwickelt worden, die den Anforderungen in der topographischen Geländemodellierung genügt.

Beiden entwickelten Verfahren ist gemein, dass zuerst eine Filterung der Triangulierung durchgeführt werden muss. Die Messung von Punkten und Linien am Gelände erfolgt immer mit zufälligen Fehlern, die von dem jeweils verwendeten Verfahren abhängen. Zur Filterung wird die sogenannte Einzelpunktprädiktion angewandt, die eine qualifizierte Eliminierung der zufälligen Messfehler erlaubt. Für jeden Punkt, also für jeden Knoten der Triangulierung, wird eine verbesserte Position aufgrund der Lage seiner benachbarten Punkte geschätzt. Die Differenz von der beobachteten zur ‚fehlerfreien‘ Position ist der Verbesserungsvektor.

Glatte *Flächenverbände* über einer Triangulierung werden im Allgemeinen schrittweise erstellt. Zuerst wird für jede Kante der Triangulierung eine Kurve auf der Fläche bestimmt, die die beiden Kantenendpunkte verbindet. Die in einem Knoten der Triangulierung, also in einem der gegebenen Punkte, zusammentreffenden Kurven müssen zu einer gemeinsamen Tangentialebene an die Fläche passen. Die Tangenten in den Kurvenendpunkten müssen also in einer Ebene liegen. Jedes Dreieck der Triangulierung wird nun durch ein gekrümmtes dreieckiges Flächenstück (einen Patch) ersetzt, das die Randkurven zu den Dreieckskanten interpoliert. Damit erhält man eine stetige Fläche, was aber noch nicht ausreicht, da eine *glatte* Fläche rekonstruiert werden soll. Bei der Bestimmung der Patches muss daher eine Konstruktion angewandt werden, die sicherstellt, dass benachbarte Patches nicht nur die Randkurve teilen, sondern auch dasselbe Feld von Tangentialebenen entlang dieser Kurve haben. Eine solche Fläche wird als geometrisch stetig erster Ordnung bezeichnet (G1).

Die entwickelte Methode basiert auf (bivariaten) polynomialen Patches vierten Grades, die als Beziér-Dreiecke beschrieben werden. Der Vorteil der Beschreibung auf Basis der Beziér-Bernstein-Polynome liegt darin, dass die Koeffizienten des Polynoms in dieser Form eine geometrische Bedeutung haben. Wie oben erwähnt, werden zuerst die zufälligen Messfehler eliminiert. In diesem Schritt kann auch die Tangentialebene an die Fläche im jeweiligen Punkt abgeschätzt werden. Alternativ dazu kann der Normalvektor beispielsweise auch durch Mittelung der Normalvektoren aller Dreiecke, die in einem Punkt zusammentreffen, festgelegt werden. Im nächsten Schritt werden die Randkurven der Patches bestimmt, die in diesem Fall (univariate) Polynome vom Grad drei sind. Die Tangenten in den Endpunkten müssen normal zu den vorher bestimmten Normalvektoren sein. Damit sind die Randkurven der Patches bestimmt, es müssen noch die ‚inneren Parameter‘ (also Koeffizienten des Polynoms) bestimmt werden, die sowohl die Form des Patches im Inneren, also auch die Tangentialebenen entlang der drei Randkurven beeinflussen. Im folgenden Schritt wird entlang dieser Randkurven ein Feld von Normalvektoren bestimmt, indem die beiden Endpunkt-Normalen ineinander überblendet werden.

Zu jedem Randkurvenpunkt gibt es somit einen Normalvektor zur Fläche. Die inneren Parameter der Patches werden nun so bestimmt, dass die Tangentialebenen der Patches möglichst normal auf die abgeschätzten Normalvektoren sind, die Minimierung erfolgt im Kleinste-Quadrate-Sinn. Da im Allgemeinen keine exakte Interpolation dieser Normalvektorenfelder, sondern nur eine Approximation möglich ist, entsteht keine vollständig glatte, sondern nur eine approximativ glatte Fläche, man spricht auch von ‚Epsilon-Stetigkeit‘ ( $\epsilon G1$ ). Wenn zwischen zwei benachbarten Patches die Winkel zwischen den aufeinander treffenden Tangentialebenen zu groß sind, müssen die Patches unterteilt werden um mehr Freiheitsgrade und somit einen besseren Übergang zu gewährleisten.<sup>2</sup>

Die zweite Methode, die zum Erstellen von glatten 3D-Geländemodellen untersucht worden ist, arbeitet nach dem *Subdivision*-Prinzip. Die Triangulierung wird dabei schrittweise unterteilt, wobei mehr und mehr Punkte und Kanten eingefügt werden. Die Winkel zwischen benachbarten Dreiecken nähern sich dabei  $180^\circ$ , wodurch die Grenzfläche, die man theoretisch nach unendlich vielen Unterteilungsschritten erhält, glatt ist. Ein großer Vorteil dieses Verfahrens ist, dass unabhängig vom Unterteilungsniveau immer eine Triangulierung vorliegt. Für diese Datenstruktur sind viele Algorithmen, also z.B. Darstellung, Verschneidung u.s.w., sehr einfach. Je nach gestellter Aufgabe kann das den Genauigkeitsanforderungen entsprechende Niveau herangezogen werden.

Auch bei der entwickelten Subdivision-Methode zur Rekonstruktion der Geländefläche müssen, wie oben erwähnt, die zufälligen Messfehler zuerst eliminiert werden. Im angewandten Unterteilungsschema wird in jeder Kante ein zusätzlicher Punkt eingefügt, der aber nicht der geometrische Kantenmittelpunkt sein muss. Ein Verfeinerungsschritt besteht im Einfügen von einem Punkt in jede Kante und der entsprechenden zusätzlichen Vermaschung der Triangulierung. Die Koordinaten der neu einzufügenden Punkte werden wie folgt bestimmt: in beiden Kantenendpunkten wird eine lokale Fläche abgeschätzt. Diese approximiert den Kantenendpunkt und seine Nachbarpunkte. Für die beiden Flächen wird dann jeweils ein für den Kantenmittelpunkt repräsentativer Punkt auf den lokalen Flächen bestimmt. Das Mittel der beiden so erhaltenen Punkte ist die Position des neu einzufügenden Punktes. Auch die Kantenendpunkte erhalten neue Positionen, nämlich jene, die den jeweiligen Punkten auf der lokalen Fläche entsprechen. In den ursprünglich gegebenen Punkten werden keine rein approximierenden Flächen verwendet, sondern die dort verwendeten lokalen Flächen interpolieren den ursprünglich gegebenen Punkt und approximieren seine Nachbarn. Dadurch interpoliert die Grenzfläche die Punkte der ursprünglich gegebenen Triangulierung.

Anhand von Vergleichen zwischen der Subdivision- und der Flächenverband-Methode wird untersucht, welches Verfahren den Anforderungen in der topographischen Geländemodellierung besser entspricht. Dazu werden sowohl tatsächliche Messungen am Gelände als auch synthetische Beispiele herangezogen. In den untersuchten Fällen sind mit den Unterteilungs-Flächen, also mit Subdivision, bessere Ergebnisse erzielt worden.

Ein kurzer Ausblick auf die Anwendungen von Geländemodellen, die Überhänge, Höhlen, etc. enthalten, wird im letzten Kapitel gegeben.

---

<sup>2</sup>Es soll erwähnt werden, dass diese Methode teilweise in meiner Diplomarbeit entwickelt wurde, obschon sie dort nicht so sehr im Kontext der Geländemodellierung steht.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Modelling of Topographic Surfaces</b>	<b>3</b>
2.1	Types of Models . . . . .	3
2.1.1	Contour lines . . . . .	4
2.1.2	Bivariate functions . . . . .	4
2.1.3	Volumetric models . . . . .	12
2.1.4	Transformation between models . . . . .	13
2.2	Global and local approaches . . . . .	15
2.3	Models in 2.5D and in 3D . . . . .	15
2.4	3D terrain models . . . . .	17
2.4.1	Problem definition . . . . .	19
<b>3</b>	<b>Algorithms for Triangulations</b>	<b>21</b>
3.1	Definition of neighborhood . . . . .	21
3.2	Parameterization of triangulations . . . . .	23
3.2.1	Projection onto a plane . . . . .	23
3.2.2	Local projection onto a plane . . . . .	23
3.2.3	Global parameterizations . . . . .	24
3.2.4	A method for local parameterization . . . . .	26
3.3	Surface approximation and estimation of geometric properties . . . . .	27
3.3.1	Normal vectors and tangent planes . . . . .	29
3.3.2	Approximating quadric as local surface description . . . . .	30
3.3.3	Approximating second order polynomial as local surface description . . . . .	31
3.4	Functionals and variational principle . . . . .	32
3.5	Mesh improvement . . . . .	33
3.6	Filtering of random measurement errors . . . . .	34
3.7	Consideration of breaklines and special points . . . . .	36
3.7.1	Neighborhood restrictions . . . . .	36
3.7.2	Prescribed tangent planes . . . . .	36
3.7.3	Surfaces and lines at special points . . . . .	37

<b>4</b>	<b>Parametric patches</b>	<b>39</b>
4.1	Patches and patch work . . . . .	39
4.2	Method overview . . . . .	41
4.3	An $\epsilon$ G1-continuous polynomial patch . . . . .	43
4.3.1	Approximate continuity . . . . .	44
4.3.2	Construction of a curve network . . . . .	44
4.3.3	Insertion of patches . . . . .	47
4.3.4	Insertion of patches and minimizing energy . . . . .	50
4.3.5	Additional splitting . . . . .	52
4.3.6	Results . . . . .	52
<b>5</b>	<b>Subdivision</b>	<b>55</b>
5.1	The subdivision paradigm . . . . .	55
5.2	Method overview . . . . .	57
5.3	Subdivision by estimation of local surfaces . . . . .	59
5.3.1	The curve case . . . . .	61
5.3.2	Surface subdivision with approximating surfaces . . . . .	62
5.3.3	Paraboloids vs. general quadrics as local surfaces . . . . .	63
5.3.4	Paraboloids vs. second order polynomials as local surfaces . . . . .	65
5.3.5	Interpolation and Approximation . . . . .	66
5.3.6	Averaging . . . . .	67
5.3.7	Roughness detection . . . . .	69
5.3.8	Results . . . . .	71
<b>6</b>	<b>Examples</b>	<b>75</b>
6.1	Vertical Wall . . . . .	75
6.2	Data set "Elev" . . . . .	78
6.3	Breaklines only . . . . .	80
6.4	Data set "Albis" . . . . .	83
6.5	Bridge . . . . .	84
<b>7</b>	<b>Conclusions and Perspectives</b>	<b>87</b>
7.1	Applications . . . . .	87
7.2	Enclaves . . . . .	90
7.3	Concluding remark . . . . .	92
	<b>Bibliography</b>	<b>94</b>

<b>A</b>	<b>Approximating surfaces</b>	<b>101</b>
A.1	Adjusting plane . . . . .	101
A.2	Adjusting quadric . . . . .	103
A.2.1	Adjusting quadric through one point . . . . .	104
A.3	Adjusting second order polynomial surface . . . . .	105
<b>B</b>	<b>Bézier triangles</b>	<b>109</b>
<b>C</b>	<b>Loop and Butterfly subdivision</b>	<b>115</b>
C.1	Loop . . . . .	115
C.2	Modified Butterfly . . . . .	116
<b>D</b>	<b>“Demo”-Example with interpolating and approximating subdivision schemes</b>	<b>117</b>
<b>E</b>	<b>Color pages</b>	<b>123</b>
	<b>Curriculum Vitae</b>	<b>127</b>

# Chapter 1

## Introduction

Digital terrain models are utilized widely in our world, reaching from map production to environmental engineering, from computer games to tourist information. They are also a scientific research topic per se. There are many different data sources for these models, each with its own characteristics. Photogrammetry, either by (digitized) film based images, by line cameras or by airborne and satellite laser scanning (also LIDAR, Light Detection And Ranging), is the discipline concerned most with the generation of terrain models. Remote sensing from satellites or airplanes in the passive optical mode or actively by InSAR (also IfSAR, Interferometric Synthetic Aperture Radar), is a supplemental method to gather digital elevation information over a very large area, up to entire planets. Ground based surveying with tacheometers, (digital) cameras, or terrestrial laser scanners, on the other hand, provides a supplemental means to record terrain shape for a big scale, but only for a limited area size. Eventually, the method of deriving a DTM (digital terrain model) from contour lines and other restituted information on terrain shape like peak points, has to be mentioned, as well as the shape from shading approach.

Many different algorithms exist for the generation of terrain models from the various data sources, and many different data structures exist for describing such models. The algorithms as well as the data structures share one limitation: the surface is considered to be the graph of a bivariate function, in less mathematical terms this is called a 2.5D model. Additionally, the parameter domain, over which the model is defined, is usually the ground plan, the height reduces to an attribute value of the point defined by its planar co-ordinates. While this enables fast algorithms, e.g. for calculating perspective views, this also has certain disadvantages. The point density in steep rugged rock or similar environments can only be very low. This is not necessarily a problem of data acquisition, but one of surface description. This problem occurs, if the surface is almost perpendicular to the reference plane. Therefore, the quality of description for these areas is poor, too. While the modelling of cliffs and similar objects is still possible, the 2.5D approach fails completely for overhangs and caves. If only overhangs are contained in the data, then it is still possible to find (compute) a parameterization of the surface but this parameterization is not a projection onto a plane anymore. For complex cave systems, but also for bridges, if they shall be modelled together with the terrain, the parameterization becomes even more complicated. The reason is that the underlying topology is not similar to the topology on a sphere anymore, but rather to a sphere with one or more handles.

A triangulation of the measured points, on the other hand, is capable of describing every surface, however complex. But using triangulations, two aspects come into play. The first is the question 'How to obtain the triangulation of the point set?'. For 2.5D cases this question can be answered easily. Algorithms for triangulation in 2D and optimization criterions (e.g. Delaunay) are available. The topology is defined in the ground plan only. For real 3D data, triangulation becomes more complex. For two points with similar xyz-co-ordinates it does not necessarily hold, that the points are adjacent. The points could be on the left and the right wall of a narrow corridor in a cave system. Then, their distance measured along the surface (e.g. by the length of the geodetic line) can be very high. This question is not treated within this thesis. References to the relevant literature will be given in the text. The second aspect is 'How is the surface defined?'. The triangulation itself can be used to describe the surface, too, by using one planar face for each triangle. This surface is continuous, but

not differentiable everywhere. At the edges and vertices of the triangulation the tangent planes are undefined, the surface itself has a sharp bend. In other words, the surface is not smooth.

In this thesis, methods will be described to obtain smooth surfaces defined over triangulations. Only mild restrictions are posed on the triangulation. Besides, the requirement on the smoothness, the methods must not favor any co-ordinate direction<sup>1</sup>. Additionally, the methods shall work locally, which means that the shape of the surface at a specific location is influenced only by the points in a defined – restricted – neighborhood.

Using such methods it will become possible to model vertical parts of the landscape as well as overhangs and caves. With the fusion of data from the various sensors described above, the need for such models arises.

In the following chapter (2, Modelling of Topographic Surfaces) a survey over the existing approaches for terrain modelling will be presented. The advantages and drawbacks of existing approaches (grid model, TIN, ...) will be pointed out, which establishes a framework for analyzing and testing new approaches. It is also helpful for defining the properties a 3D surface model should have, and it explains, why it should be based on a triangulation. For completeness a short section on the conversion from one model type into another is included, too. It can be skipped by the readers with DTM experience.

Chapter 3 (Algorithms for Triangulations) introduces algorithms which are defined on triangulations. Some of these methods are algorithms for estimating geometric properties (like normal vectors) at the vertices. In contrast to the previous chapter it will not consider the terrain especially. However, it is not a systematic survey but restricted to the algorithms required for terrain modelling in the following chapters. Some of the algorithms do not require that the points are triangulated, but in these cases the triangulation is convenient for another purpose. It defines neighborhood relations and can be used to restrict the set of points an algorithm is applied to, to a certain neighborhood of a point. Readers with computational geometry background should read quickly through this chapter. In the last sections of this part, the measurement process and the properties of the terrain will play a role. The filtering of random errors and breaklines in triangulations will be treated shortly.

Chapters 4 and 5 present two new methods for reconstructing a smooth surface over a triangulation. They will be presented together with comparable algorithms. The first of these two chapters (Parametric patches) is concerned with defining one patch for each triangle. Such a patch is a small curved surface which is defined over the domain of one triangle. The shape of the patch is governed on the one hand by the local behavior of the surface (curvature, ...) and on the other hand by continuity requirements across the patch edges. Adjacent patches should not only share a common patch border but also the tangent plane field along the common edge. The other chapter (Subdivision patches) describes methods to refine a triangulation of a terrain surface. In one subdivision step new vertices are inserted into the triangulation. Over each edge one new vertex is generated, new edges between these vertices are inserted accordingly. This step is repeated several times, and the limit surface of this algorithm should be a smooth surface.

In chapter 6 examples with real and synthetic data are presented and compared, also with respect to a 2.5D surface modelling technique.

Eventually, this work is concluded by its conclusions.

---

<sup>1</sup>This statement will be relativized in the conclusions.

## Chapter 2

# Modelling of Topographic Surfaces

A topographic surface is a surface which is used in topography. The most important one is the terrain surface, also called the ground surface. While its definition is easier in natural open areas, it becomes more and more complex for cultivated, vegetated, wooded or built-up areas. The reason is, that the solid ground is usually considered as the terrain surface, but this ground may be covered. One possibility is that it is covered by buildings (houses), but it may also be covered by a street. In an open area the street surface can be considered as the terrain, but in a built up area under a close inspection the terrain surface at the street, containing multiple levels, is debatable. Also for wooded ground, or at an agricultural field (possibly ploughed), different terrain definitions may differ within a few decimeters. A digital model of the terrain surface is called digital terrain model (DTM). In contrast to a DTM a DSM is used to describe the 'surface' in the sense of visible surface from an elevated position (e.g. a surveying plane). This surface runs through the canopy of trees and along the vertical walls of houses and over their roofs. Depending on the specific application a surface model may run on top of crop stems or on the 'ground' of a field. Different concepts exist for bridges in topographic models. For one group of applications it is required, that the upper surface of a bridge must be included, e.g. because of ortho photo production. What is below the upper surface is not of interest. Another group requires, that the bridges must not be included, because they are not part of the solid terrain (e.g. for watershed analysis). Finally, combining the different requirements at different topographic features (bridges, woods, fields, cities, ...) generates a huge number of possible topographic surface models. All these surfaces, including the terrain, have a micro variability which shall usually not be modelled. A generalization process is necessary in order to eliminate the micro variability. All these questions will not be treated within this work. It is assumed the points which are given lie, up to a certain accuracy, on one of the surfaces specified above. In the entire work topographic surface, terrain surface and ground surface are therefore synonymous.

In this chapter different approaches to the modelling of topographic surfaces will be presented and compared. This is a review of existing approaches and algorithms and may be read through quickly. This is followed by a short comparison of global and local methods and a closer look on 3D vs. so-called 2.5D surface description. At the end of this chapter the propositions for a 3D terrain model will be stated. It is interesting to review the different existing models under these propositions for 3D terrain modelling.

## 2.1 Types of Models

The first forms of terrain models, though not digital, were analogue topographic maps. In their early forms they merely indicated the existence of mountains and gorges, but their spatial extent, in planimetry as well as in height, could not be deduced reliably from these maps. Different forms of terrain representations were introduced later, e.g. contour lines which allow to obtain geometric properties (steepness, ...) in an exact way. However, the first models will not be treated in detail here, the discussion will be confined to models with well defined mathematical properties. On the one hand, these are the mentioned contour lines, which form an integral part of a topographic map nowadays, and on the other hand, these are functions in 3D space, which

will be treated in more detail. These functions can be considered as height functions, describing the height at a specific point, or they can be considered as the boundary surface between air and solid land. The following overview will be structured according to the parameter space, over which the height models are defined.

### 2.1.1 Contour lines

Contour lines do not only describe the terrain shape in topographic maps, in digital form they have been used as terrain models as well. The surface is not described as a function over a continuous parameter domain, but by its intersections with a predefined set of parallel planes. For many applications this description is not convenient, its existence has to be explained by the process of collecting terrain information for the production of maps. In former years the contour lines have been measured directly in a stereo plotter. Therefore, it was tempting to store this line information digitally and derive information directly from the contour lines. There are, however, severe disadvantages with this kind of model. These can be attributed to the parameter domain, over which the height is primarily defined: a set of lines. Therefore, no height information is contained between those lines. Some tasks can be solved easily within such a system. If a line is given in the ground plan, e.g. an axis of a road, it can be intersected with the ground plan of the contour lines. At the intersections, the height of the line can be determined, in between linear interpolation is applied. Problems can occur, if the end points of the line do not fall on a contour line, and also if such a line runs for a long time between two contour lines. These models can be described with a question: 'Where is a (specific) height?'

### 2.1.2 Bivariate functions

A more precise term for the models which will be discussed in this section is bivariate functions over a connected parameter domain. The large number of different models falling into this category can be described by the question: 'Which height is at a (specific) location?' These models describe the height as a function over a parameter domain, e.g. the  $xy$ -plane. Obtaining the height of a line or a curve, which is given in parameter form in the ground plan, results in inserting the definition of the ground plan co-ordinates into the function. The terrain model, which is the function, maps every point into 3D space. However, usually this function cannot be written as one simple term. The function is usually represented at specific locations, and an interpolation rule for the locations in between. The function can either be scalar-valued or vector-valued. This distinction will be elaborated later (Sec. 2.3). For the time being, the text will be restricted to a scalar-valued function which maps a ground plan point to its height. There are different models falling into this category (e.g. Raster, Grid, ...) which will be described in the following.

#### Raster

A raster elevation model is a very simple form of elevation model. The parameter domain is divided into cells of equal shape. Though there are infinitely many possible shapes which provide a segmentation of the plane into equal cells only the triangle, the rectangle and the sexangle play a role. The only form of practical importance is the rectangle, and most often it is a square.

In a raster model, one height is defined for the complete cell, which means it is some integral value for the definition area of this cell. It can either be the mean value of all heights falling into this cell, the maximum or the minimum value, or the median. Other definitions, e.g. based on the distribution of the heights in the cell are possible, too. Of course, it is also possible to take a weighted average of the values in the cell, weighted e.g. by the distance to the center point of the cell. If only some heights which have been observed within the cell, are used to determine the height value, this value is an estimation for the respective quantity. Another possibility is to determine the height of the cell by block kriging [Journel and Huijbregts, 1978]. This method also provides an estimator for the mean value of all heights falling into the cell.

Such a model is, due to implementation matters, usually defined over a rectangular domain. This domain is specified by one reference point (e.g. the center or the outermost corner of the lower or upper left cell), the

cell size, and the number of cells in the two orthogonal directions. Additionally, a rotation of these directions against the co-ordinate directions can be regarded, too, but this is not always the case. If the area of interest is not rectangular – but also in other cases – a specific value for not defined heights is used.

This model is very similar to a one band digital image. The gray value (intensity value) of the image corresponds to the height, each cell corresponds to one pixel. If different measures for the height are given within each cell (e.g. minimum and maximum height), this corresponds to a multi-channel digital image. These kinds of model can be found in raster GIS. The GIS Idrisi [Clark Labs, 2002], for example also allows to have more than one elevation layer in one ‘image’, containing e.g. the minimum, the maximum and the mode (most common) height within one cell.

To obtain the height of a point in ground plan, it is necessary to locate the cell in which the point is falling into. The height value which is stored for this cell is the height of the point. If neighboring cells have different heights there are discontinuities in the elevation at the cell borders.

Such models can be easily implemented as a real-valued matrix, with  $n$  rows and  $m$  columns. To provide the transformation from world co-ordinates to matrix indices, parameters for the “absolute orientation” of the raster are necessary:  $x_0$  and  $y_0$  of the reference point, possibly a rotation angle  $\alpha$  and the cell shape parameters, most often only the edge length  $a$  of a square.

**Examples** The German company TopoSys [TopoSys, 2002] uses such a model for the representation of the terrain. Their measurement method is airborne laser scanning, which is known to collect points not only on the ground but also on off-terrain obstacles from the point of view of the laser scanner sensor (e.g. foliage of vegetation). On the other hand, so-called long ranges, i.e. gross errors where the measured laser range is much longer than the distance from the sensor to the ground, can occur, too. Thus, the rastering strategy for obtaining the first terrain model could be to take the first quartile of the heights falling into one raster cell. This is not sufficient to eliminate all off-terrain points, or to be more precise all off-terrain heights. Therefore, filtering operations are performed consecutively. These can be defined as operations on digital images and within a digital image processing software.

The following figures show two small sample areas of a project from the Vienna Magistrate [Briese et al., 2001] (and [Pfeifer and Briese, 2001]) flown by TopoSys. Fig. 2.1 shows a set of laser scanner points and the 0.5m-raster. As it can be seen, not every raster cell contains a point. Applying the rastering strategy explained above does not assign values to all raster cells. Therefore, an interpolation step is applied to the raster. Fig. 2.2 shows the result for a different sample of the same data set, after the holes have been filled. As it can be seen, not all cells define the ground, but on the left hand side of the image the heights correspond to the low vegetation (bushes in the Belvedere park in Vienna).

## Grid

In a grid model the heights are stored for the so-called grid points. The grid is defined in the ground plan. Neighboring grid points are connected by a grid edge, and a closed loop of grid edges (containing no additional grid edge) circumscribes a grid mesh. The grid is regular, which means that all grid meshes have the same shape and size. Theoretically, not only rectangular grids are imaginable, but these are the only ones of importance. Only the so-called hybrid grids form an exception. In the following only rectangular grid meshes will be considered. Each inner grid point has four direct neighbors and another four diagonal neighbors. Like in the raster model, the neighborhood relations between the points are known implicitly. In comparison to the raster models, which are area based, the grid models are vector based.

The heights of the grid points are usually determined by some surface interpolation or approximation method. The most well-known are:

- Moving least squares (e.g. [Lancaster and Salkauskas, 1986]) with adjusting polynomials of low degree (e.g. a tilted plane or a horizontal plane). Interpolation with a moving adjusting horizontal plane is also known as inverse distance weighting, if a corresponding weight model is used.

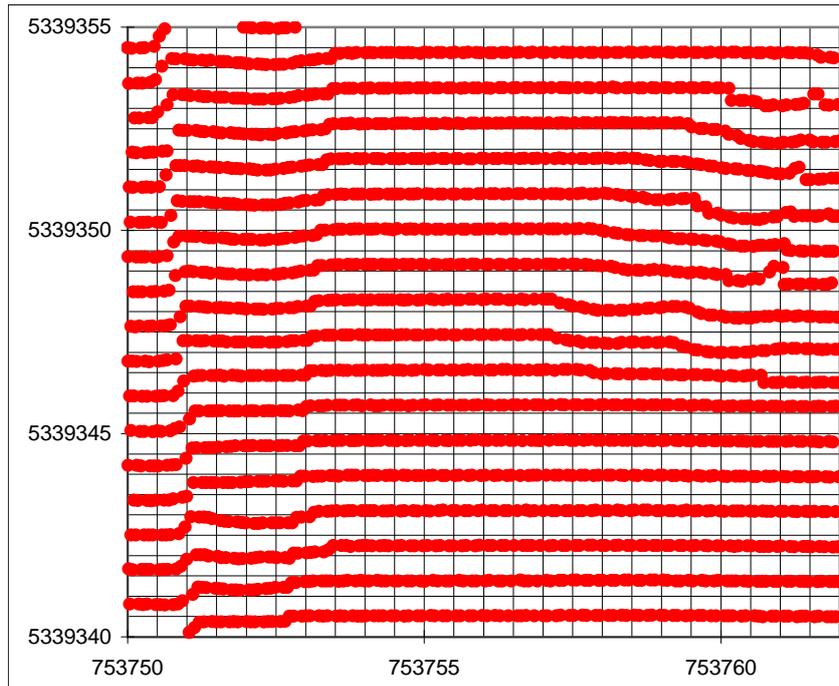


Figure 2.1: Raster model in top-view with original points. The diameter of the points is the same as the footprint of the laser scanner (TopoSys, flight direction is East-West). Only the center of a 'point' is used in order to decide which cell the point belongs to.

- Point kriging [Journel and Huijbregts, 1978], including on one hand the method of linear prediction ([Kraus and Mikhail, 1972], simple kriging with gaussian variogram), and on the other hand the multi quadrics method ([Hardy, 1971], simple kriging with unbounded variogram).
- Finite element methods (FEM) [Hoschek and Lasser, 1993], where the surface is composed of small (finite) compound surface elements, adjacent elements share a common edge. These surface elements shall have small distances to the given terrain points. Additional observations (e.g. stiffness of terrain, i.e. small curvature) are used to guarantee that the equation system for computing the grid heights is solvable and to avoid oscillations.

These methods could be used to describe the heights over the complete domain. But in a grid model the heights are only determined at specific points, the grid points. The distance between two grid points in the ground plan (i.e. the grid edge length) is the discretization interval for the interpolated surface.

An additional rule is necessary to interpolate the heights in between the grid points. In combination with this rule the discretization interval has to be chosen appropriately in order to avoid loss of precision for the determination of the height at an arbitrary point. The following list shows some methods which are used for the interpolation within the rectangular grid meshes:

1. Separation of the mesh into two triangular faces. There are two possible choices for splitting one mesh into two triangles. Another possibility is to insert one additional point in the center of the mesh and split it into four triangles.
2. The interpolation of the grid points in the mesh and possibly its neighboring grid points by a polynomial.
3. Bilinear interpolation of a mesh, this corresponds to a bivariate B-spline of degree one. In each mesh this surface is a hyperbolic paraboloid.

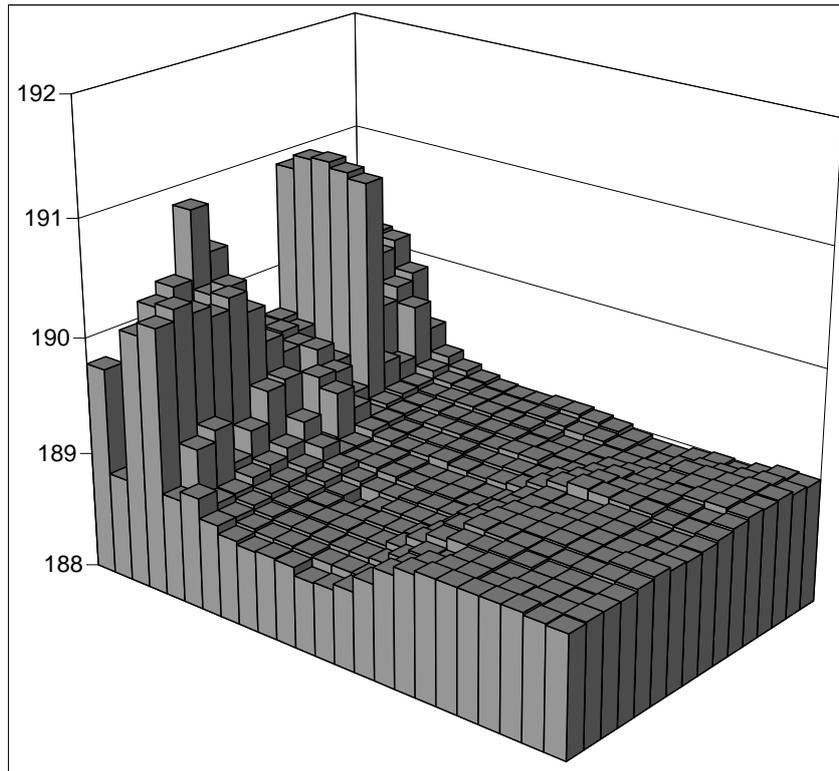


Figure 2.2: Raster model with a raster width of 0.5m, the z-values are given in meter. This model is a surface model which also includes vegetation and houses (in general non-terrain points, in this case bushes).

4. The interpolation with a tensor product surface. With a bi-cubic Bézier surface a continuously differentiable surface can be determined [Hoschek and Lasser, 1993].

Of course, the methods 2 to 4 all fall into the category where one polynomial is defined for one mesh.

A grid model is, in some aspects, similar to a raster model. An important difference is, that the heights are not defined over cells (areas), but for points. A grid and a raster are dual to each other.

It has been mentioned above, that hybrid grids form an extension to ordinary grids. A hybrid grid additionally contains lines which are most often represented as polygons, but this limitation is only caused by practicability. The hybrid grid is defined as a grid intermeshed with the special lines. This means, that the intersections of the special lines and the grid are computed and included in the model as well as the vertices of the special line. In areas where a line crosses the meshes, the surface can be defined by a triangulation (see below).

These special lines are usually *breaklines*. Along such a breakline the tangent planes of the terrain show a discontinuity. Other special lines are the so-called form lines. They are also called soft breaklines and are used to describe ridges of mountains or similar terrain features.

Another extension for grid models is that the edge length does not necessarily have to be constant throughout the complete domain but may vary (e.g. by factors of two) according to the given measurement point density.

It shall be mentioned, that grid models, just as well as raster models, can be analyzed and compressed easily by a wavelet transformation [Stollnitz et al., 1996]. Due to their regular structure the wavelet transform can be stored in the same structure, which is eventually a matrix. Algorithms to include also the breaklines of a hybrid grid in the wavelet transformation have been suggested by Beyer [Beyer, 2000].

The implementation of a grid model is similar to the one of a raster model. For a hybrid grid the line information has to be stored, too. A solution is given in [Kraus, 2000].

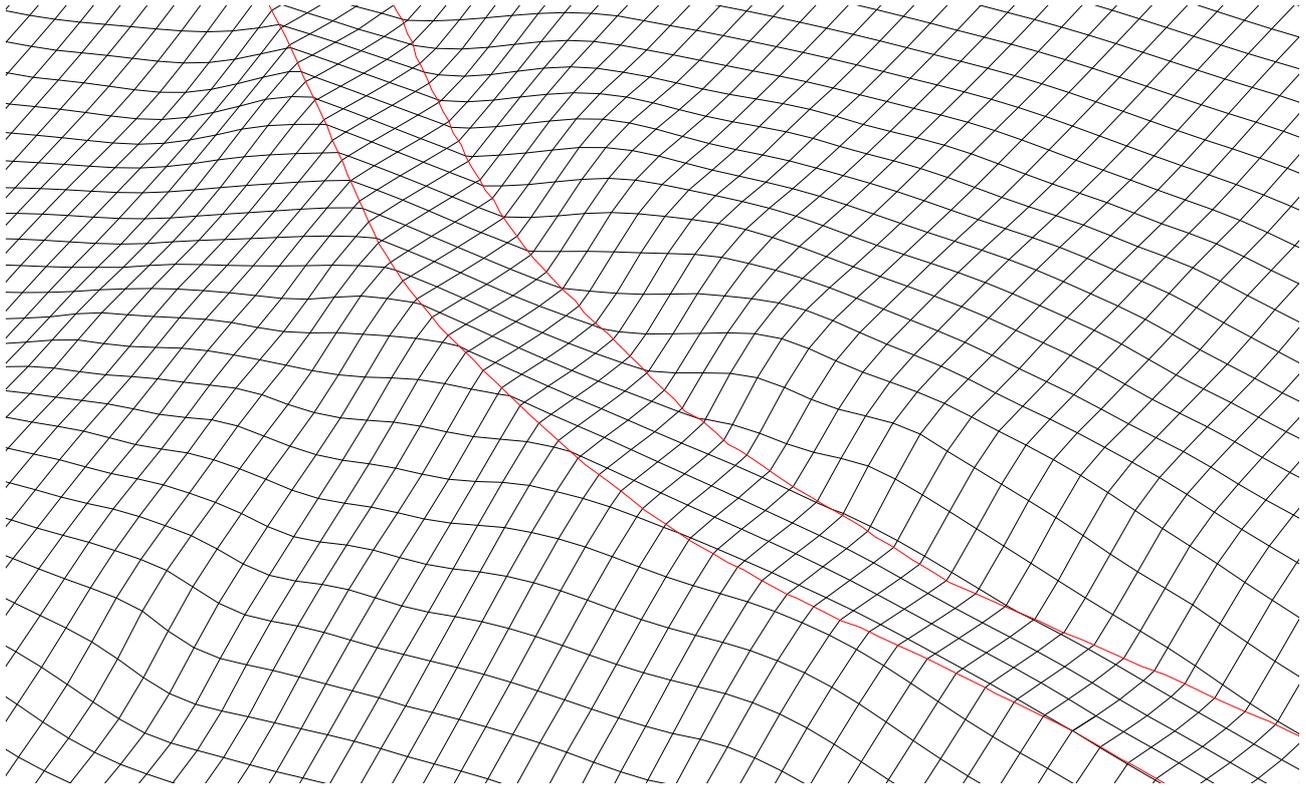


Figure 2.3: Hybrid grid model in a perspective view. The grid lines show breaks at the breaklines.

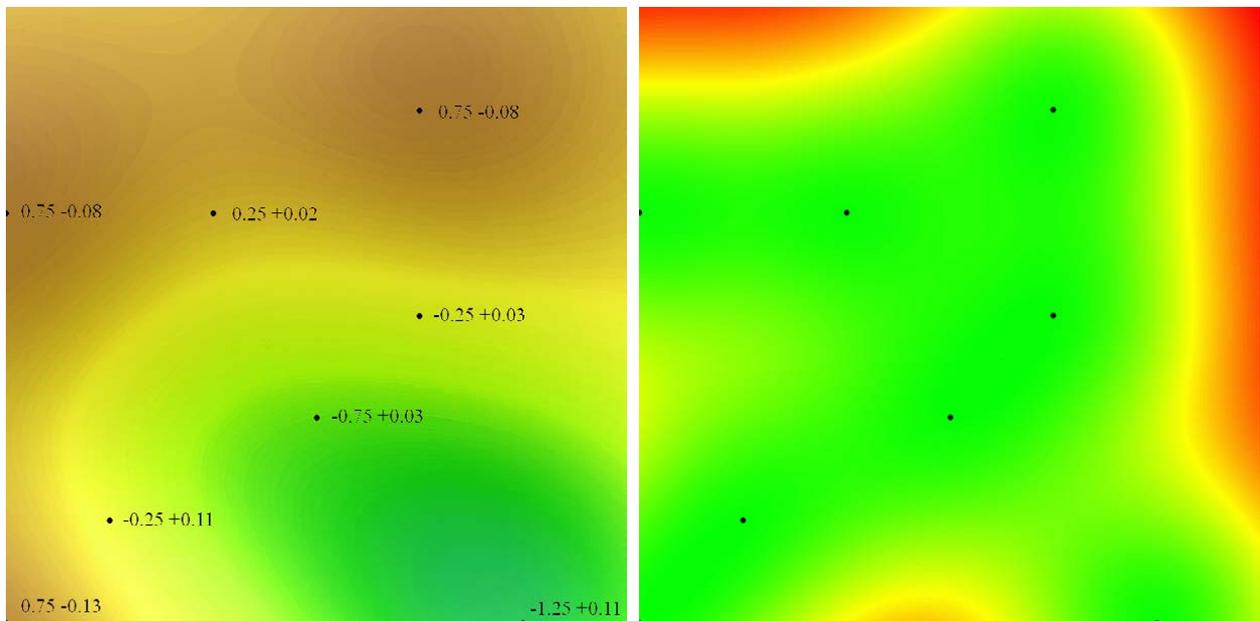


Figure 2.4: Terrain model and accuracy model. Terrain heights reach from -1.25m to 0.75m (green to brown), next to the observed heights the residuals are shown. The accuracies (right image) range from  $\pm 3\text{cm}$  to  $\pm 44\text{cm}$  (green to red). The locations of the original measurement are indicated as black dots. This figure is also shown in color as fig. E.2 on page 124.

**Examples** The terrain model program HIFI [Ebner and Reiß, 1978] uses the finite element method to compute the grid heights, the program SCOP [I.P.F., 2001] provides linear prediction and the program Surfer [Waterloo Hydrogeologic, 2002] offers kriging and inverse distance weighting.

Fig. 2.3 shows a hybrid grid model with intermeshed breaklines in a perspective view. The data was captured by laser scanning, the breaklines were determined in a semi automatic way [Kraus and Pfeifer, 2001]. The grid model was computed with SCOP [I.P.F., 2001].

Fig. 2.4 shows the original points as black circles and the grid model in a z-coding. The discretization interval has been chosen sufficiently small to avoid any effects at the shown scale. On the right side a so-called accuracy model can be seen, which shows the accuracy of the interpolated grid points. It has the same structure, but the function values are accuracies and not heights. The heights and their accuracies have been determined by kriging with a gaussian variogram model<sup>1</sup>.

Fig. 2.5 shows four methods for defining the surface within the grid heights. All methods provide continuous surfaces, but only the SCOP 12-point interpolation method and the tensor product Bézier surface provide a continuously differentiable surface. For all of the surfaces there is a discontinuity in the curvature at the mesh edges. Continuity could be reached with a tensor product Bézier surface approach by using an even larger neighborhood for the definition of the heights within one mesh and a higher degree surface.

## TIN

The methods described so far use the observed data to compute a function. If this function is found, the original points are no longer necessary. A triangular irregular network (TIN), on the other hand, is based on irregularly sampled points of the terrain. The original measurements themselves are used to describe the shape of the terrain. Instead of TIN also the term triangulation is used. As the grid method explained above, it is a vector based approach.

The edges of a TIN connect adjacent points and define neighborhood relations, but these have to be generated from the points by triangulating the data. There are different criteria for defining which points are neighbored, their applications resulting in different triangulations of the same data set (e.g. Delaunay triangulation, greedy triangulation (described for example in [Dickerson et al., 1997], triangulation with shortest total edge length), triangulations with minimal maximum angle in the triangles [Bern et al., 1993], ...). If the restriction from above (scalar-valued height function) is applied here, too, the triangulation can be performed in the xy-plane, which is the parameter domain. If the height values are used in the formulation of the triangulation criterion, then this is a so-called *data dependent triangulation* (DDT). One possibility is to minimize the angle between triangle normals [Dyn et al., 1990]. Special lines (polygons) can be considered, too, by demanding, that the edges of the polygons are also edges in the triangulation. This is a so-called constrained triangulation.

The triangulation itself does not define a surface. At first, it only establishes neighborhood relations (topology information), heights are given at the vertices. In order to obtain a height function an interpolation rule for the inner parts of the edges and the triangles is necessary. Usually, the most simple form is used, which is to interpolate the heights linearly between the three vertices of one triangle. A different possibility is to compute patches for each triangle. A patch (also called a local surface) is a small curved surface, which is defined over the domain of the triangle. If the three patch corners have the corresponding heights of the data points given at the vertex locations, then this patch is called an interpolating patch. Usually, patches are determined in a way that guarantees a certain smoothness to vary continuously over the complete surface across the patch borders. Well known patches for the 2.5D case (see below) are the so-called complete quintic patch and the Clough-Tocher triangle which provide C1 continuity (differentiability of degree 1) [Lancaster and Salkauskas, 1986].

Preserving the original points in the TIN does not necessarily provide the highest precision. The height at an arbitrary position is in general estimated by the heights of three (original) points. These three points form the triangle, the point belongs to. By applying a method like kriging the height at an arbitrary position is estimated by more points and can be more precise.

---

<sup>1</sup>In the case of bilinear height interpolation (a weighted average) the correct computation of the accuracy of the height of a point falling into a grid mesh would involve using the variogram, but also considering the correlations between the four affected heights in the averaging within the law of error propagation. Applying bilinear interpolation for the accuracies, too, gives too optimistic results. However, for smooth surfaces the error becomes smaller by decreasing the discretization interval.

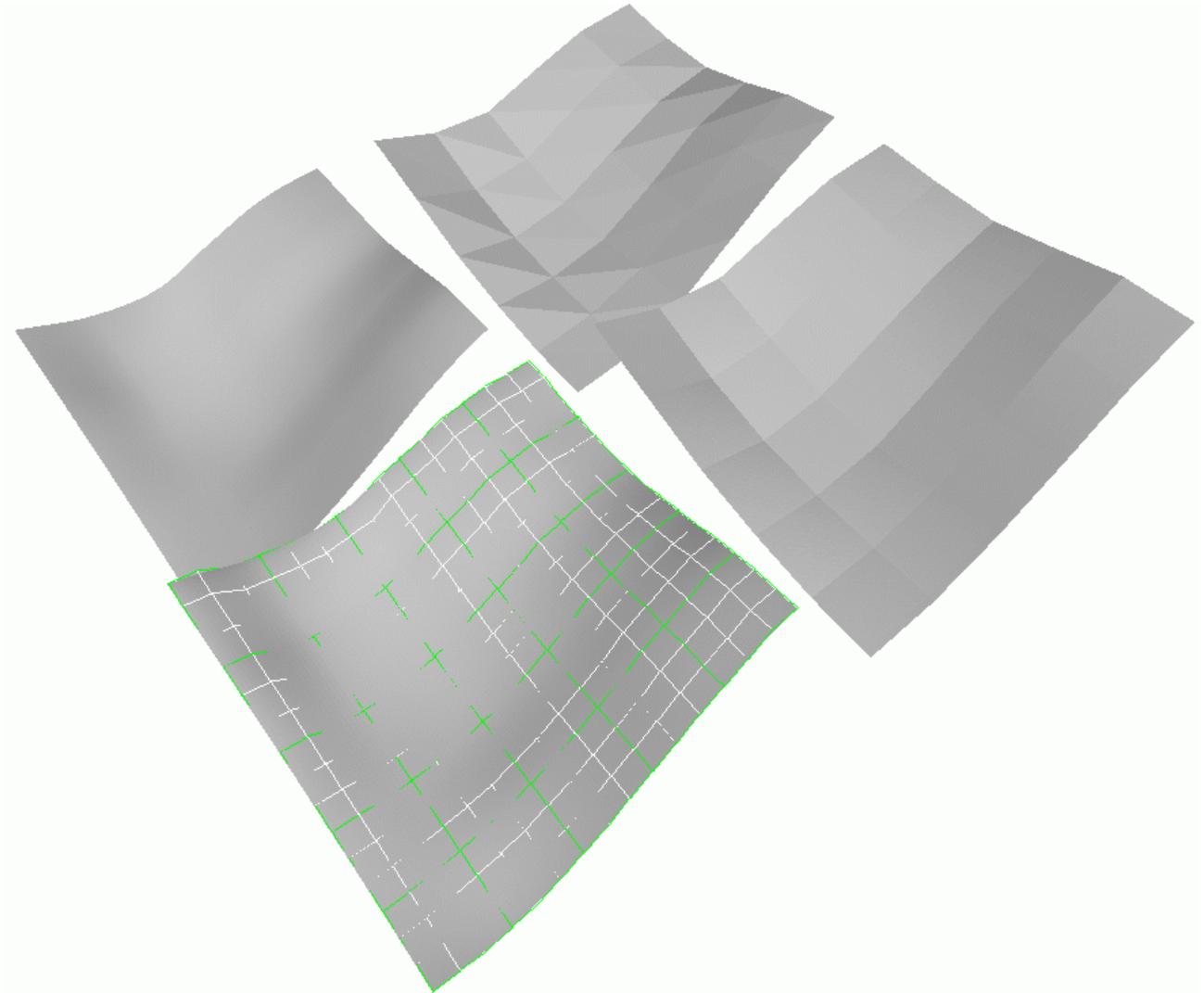


Figure 2.5: Interpolation of terrain heights in between the grid points. Upper surface with triangulation, right surface with bilinear interpolation, left surface with so-called SCOP 12-point interpolation (tangent plane continuous), lower surface is a bi-cubic tensor product Bézier surface, the control net is shown too. (Darker (in green): patch boundaries, lighter (in white): inner control points). For the Bézier surface the Bessel method was used for the generation of the curve mesh and the Bessel twist was used to determine the other parameters of the surface. Note, that none of the surfaces has continuously varying curvature. The height at an arbitrary position depends on 3, 4, 12, and 16 points, respectively.

**Examples** Fig. 2.6 shows a constrained triangulation of tacheometric and photogrammetric terrain data in the city of Vienna. Only the edges are shown, no hidden face removal was performed. The data was taken from the digital map of the Vienna Magistrate („Mehrzweckkarte“, the digital “multi purpose” map). Though the point densities vary strongly and large height jumps can be found in the data, the surface does not show unwanted undulations. Of course, this is a consequence of using the original measurements (points) and defining the surface as flat within each triangle.

In Fig. 2.7 a triangulation of a small point set is shown in the ground plan. The points were triangulated with the Delaunay criterion using the program SCOP.TRI [I.P.F., 2001]. The outer edges of the triangulation form the convex hull which leads to narrow triangles with long edges. Of course, these triangles do not describe the surface properly. In Fig. 2.8 the triangulation is shown in a perspective view with two different types of patches defined over each triangle. Left, the surface is shown with flat triangles which leads to discontinuities

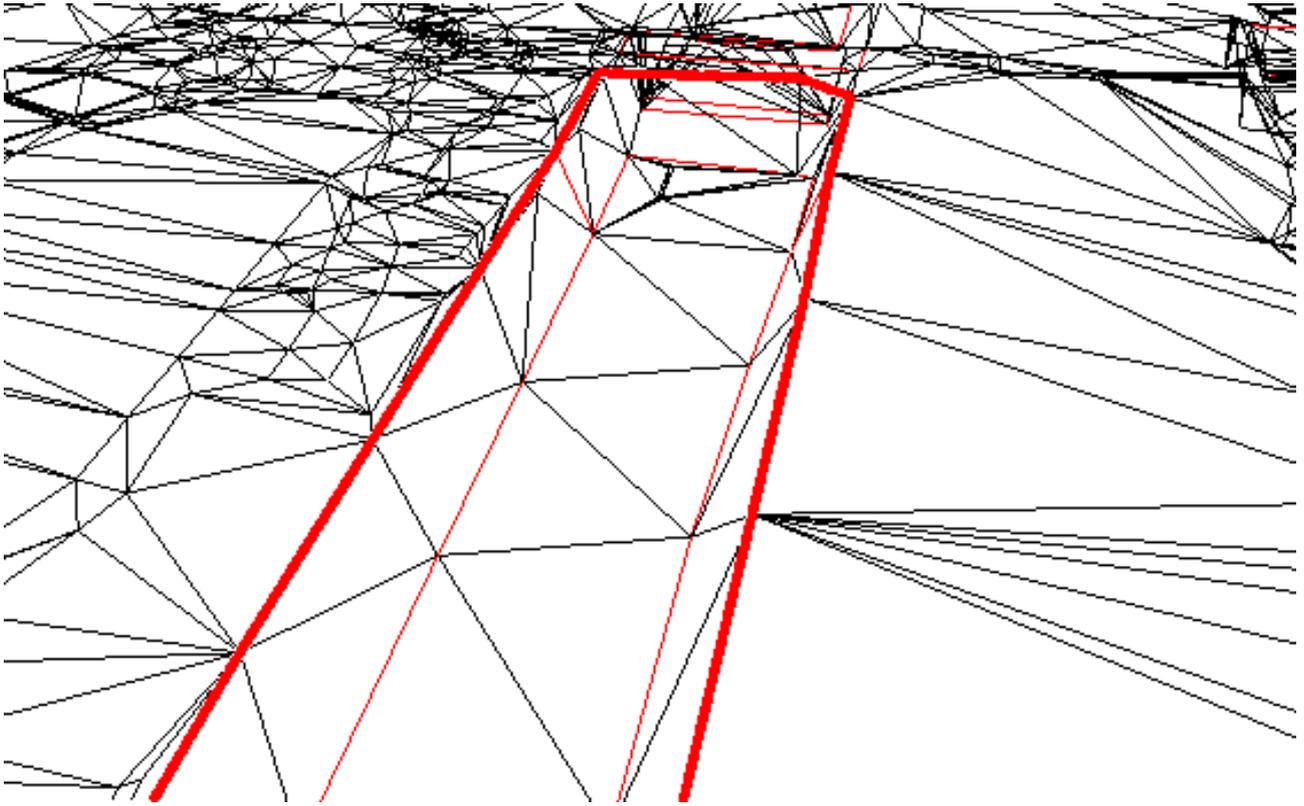


Figure 2.6: Constrained triangulation with breaklines in a perspective view. The point density varies strongly in the data set. One breakline is plotted fat.

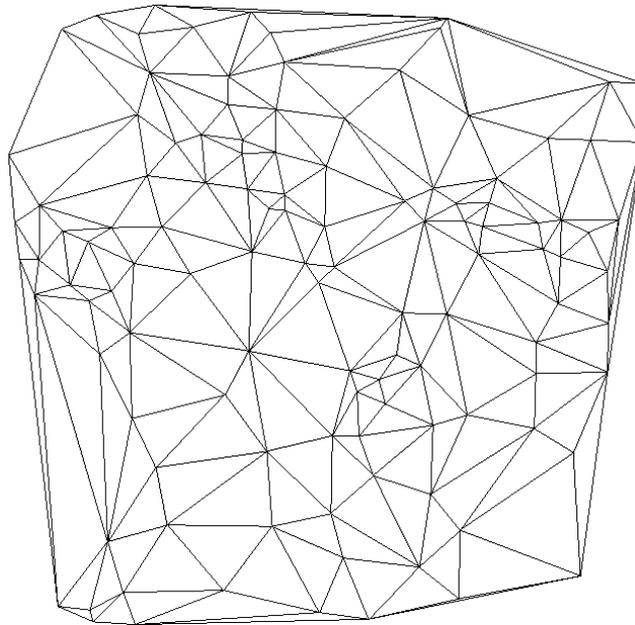


Figure 2.7: Triangulation of a point set with Delaunay criterion.

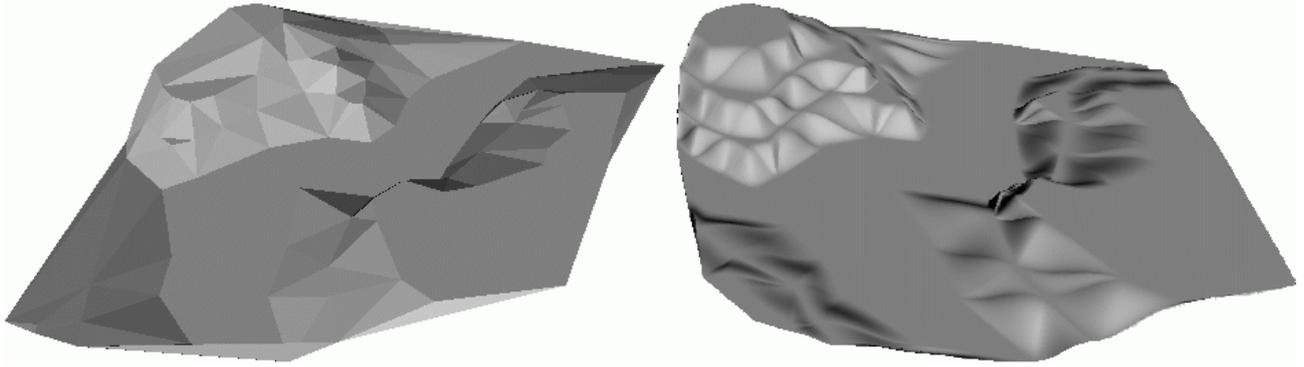


Figure 2.8: Left: surface with planar faces, Right: tangent plane continuous surface with quintic patches. Obviously, the method for computing the normal vectors is inappropriate for the given data, producing too many horizontal tangent planes (e.g. in the left background part of the surface).

of the first derivative. The right surface has been computed with the complete quintic patches over each triangle to achieve continuity of the tangent planes. Here, each patch is a polynomial of order five. Not only the heights of the points are necessary to compute the quintic patches, but also first and second derivatives in the patch corners and the first derivative across each patch edge in one additional point. All these data are interpolated by the patch. For the computation of these and the computation of the surface the algorithm from [Preusser, 1990] was used. The triangular structure is still visible. Like in Fig. 2.5 the curvature is not continuous.

### Expansion in series

Fourier expansions are a different form for the representation of height values. The heights (or function values) are stored in the coefficients of the harmonic functions of different frequencies. This technique is used in geodesy for the description of heights (e.g. geoid heights, but also for gravity values) for the complete earth using the spherical functions (so-called spherical harmonics method). The advantage lies in the treatment of the data near the poles and that no special continuity conditions have to be formulated on the border of the parameter domain, because this is given implicitly in the periodicity of the basis functions. The coefficients are integrals on the sphere over products of the harmonic functions with the height values. Of course, usually only observations at discrete positions of the heights are available.

The spherical harmonic method can also be seen as a method to generate the model, but it need not necessarily be stored in this structure. It is, of course, possible to sample the heights on the sphere, and map the sample locations into a plane. This can be performed in such a way, that the sampling points generate a regular grid in the plane. The model can then be stored as a grid model.

For not spherical parameter domains this technique plays no role in the practices of terrain and height modelling.

### 2.1.3 Volumetric models

The terrain surface is the interface between air and land (solid earth). Latter one consists of rock and soil (...) which are volumetric. In a volumetric terrain model each point  $P \in \mathbb{R}^3$  has the property to belong to the solid earth or not. Of course, this can be extended to describe other properties like rock density or type of soil. Depending on the model type, the surface itself is not represented explicitly. Those points where there is a sudden change in the property from land to air (or from a density above zero to (near) zero, ...) form the surface. Not only overhangs and caves can be modelled in this way, but also hollow spaces below the visible terrain surface.

Voxels are one possible model type (spatial occupancy), which can be handled in an octree data structure [Samet, 1989]. B-Rep (boundary representation, [Mäntylä, 1988]) models describe the volume by its bounding surface. However, these models play a role in medicine, geophysics and other disciplines, but not so much in terrain modelling for large areas. One application in terrain modelling is given in [Chen et al., 1994b] and [Chen et al., 1994a], where the terrain is modelled together with the buildings in one data structure, a tetrahedral tessellation. A raster based method is used to derive a constrained Delaunay tetrahedral tessellation, where constraints are formed by the outlines of the buildings.

#### 2.1.4 Transformation between models

In this section the discussion will be confined to those models which are relevant for practical applications (contour lines, TIN, grid and raster). The different models have different advantages and drawbacks and are more or less appropriate for specific applications. Most algorithms can be applied to all models, however, the results will be different, too. Examining the example of contour line derivation the following results can be found: Contour lines derived from a TIN usually show zig-zag lines, if narrow triangles are in the data. This is due to the missing filtering (removal of random measurement errors) in a TIN and a data distribution where some points are very close to each other, especially compared to their accuracies. Contour lines from a grid model are usually more smooth, because of the filtering and the regular point structure. On the other hand, it can occur, that an original measurement appears to be on the wrong side of a contour line. Though, from the point of filtering, this may be correct, some practitioners distrust the idea of changing (i.e. filtering of random errors) the height of a measurement.

As mentioned above, most algorithms (contour line derivation, profile computation, watershed analysis, volumetric investigations, aspect analysis, curvature computations ...) can be performed for each type of model. The advantage of grid models over a TIN is clearly, that the neighborhood relations are known implicitly and the model area can be traversed easily. In a TIN the neighborhood relations are stored explicitly which makes traversing the area more expensive, but its advantage is the ability to adapt locally to a certain point density and terrain variability.

Nevertheless, in many situations the need for a model conversion emerges – most often because a computer program for a specific application can only handle one model type.

##### From contour lines to grid, raster and TIN

To compute a grid model from contour lines, two different methods shall be mentioned.

**Profiling method:** a profile is defined in the ground plan, its center is the grid point. The first intersections of the profile with the contour lines provide two 3D-points which define a line. The height of the grid point is the height of this line at its position. The direction of the profile has to be chosen carefully. One possibility is to try several directions (e.g. 4) and take that one where the profile intersects the contour lines at angles close to a right angle. Finding a good direction is difficult [Takagi and Shibasaki, 1996]. An extension is to take not only the first intersections of the profile with the contour line, but more intersections and use an interpolating or fitting curve for the height determination.

**Window method:** For each grid point the height is determined from its neighboring contour line points by interpolating a surface with these points. The feature that these points are along a line remains unused. The size of the neighborhood and the method of interpolation are the parameters of this method.

The shortcomings of both methods are obvious. The profiling method relies on a good determination of the profile direction. An optimal profile may not even be a straight line. The disadvantage of the second method is, that it does not consider the line information. Both methods produce artifacts, especially in flat regions. A short evaluation of the methods is given by [Takagi and Shibasaki, 1996].

Though the methods presented so far can also be applied to raster models, they are essentially grid methods, because they operate on points, and not on areas (cells). In the following, methods which are defined for raster models will be described.

**Chamfering method:** First, the contour line vectors are rasterized. Each cell that is intersected by a contour line is given the height of this contour. By chamfering [Borgefors, 1986] all other cells are assigned the distances to and the heights of the nearest contour lines (uphill and downhill). The height of the cell is obtained by averaging these two heights with the distances as weights. This corresponds to a linear interpolation if the contour lines run parallel. If a cell is inside of only one contour line, than it obtains the height of this line. This algorithm is applied for example in GRASS [GRASS GIS, 2002].

**Buffering method:** A different approach has been presented in [Takagi and Shibasaki, 1996]. Again, the contours are rasterized first. Intermediate contour lines are found by uniform region growing ("buffering") all contour lines until all cells are filled. At the borders of adjacent regions new contour lines are found which have the mean height of the adjacent regions. Repeating this process provides more and more contour line cells, until all cells have been assigned a height.

Problems with these algorithms occur, if more than one contour line falls into one cell. Furthermore, the heights obtained are not uniformly distributed [Carrara et al., 1997]. To some extent these effects can be decreased if the the cell size is made smaller, but at the cost of longer computation time and higher memory demands.

To convert contour lines into a TIN is – at first sight – a straight forward task. The line points are triangulated and the contours are treated as constraints. Therefore, neighboring line points of one contour line will always be connected by a triangle edge. However, this does not always yield satisfying results, because peaks are typically missing in such a model. Instead, the area circumscribed by a locally maximal contour line is flat. The same problem (horizontal region) occurs, if a contour line makes a sharp turn. Methods have been developed that improve the triangulation near flat triangles:

**Rule based classification:** as it has been proposed by [Heitzinger and Kager, 1999] and also by others (e.g. [Peng et al., 1996]). The points on the contour lines are triangulated. The lines form constraints, which means, that adjacent points on a contour line are always connected by a triangle edge. Horizontal triangles are searched and their neighborhood is analyzed according to a set of defined rules. In this way areas with ridges, peaks, sinks and saddles are found. These points and lines are then computed explicitly and inserted into the triangulation.

A comparable method has been specified in [Gonçalves et al., 2002]. The triangulation is constructed and horizontal triangles are searched. A structure line is computed along sequences of triangles with horizontal edges. In this sense, a sequence is a set of triangles where consecutive triangles share a common edge. These lines connect contours of neighboring intervals. Along these lines heights are estimated. The last step is to compute a grid model with finite element methods considering the contour heights and the estimated heights at the structure lines.

## Grid versus raster

As it has already been mentioned, grid and raster models are dual to each other. The center of a cell can be taken as the position of a grid point and vice versa. The edge length of the cell corresponds to the grid point distance. The height of a grid point is equal to the height of the cell it belongs to.

In practice, the differentiation between raster and grid model is not made as clearly, as it has been proposed in the previous section. This is because the data structure is – more or less – the same, and also because the conversion is so simple. To some extent the differentiation also appears to be unnecessary.

## TIN versus grid

A grid can easily be triangulated. Adjacent grid points are already connected by edges, only diagonally adjacent grid points have to be connected. There are two possible choices for the connection, as it has already been mentioned in the section on grid models (defining the grid heights within one grid mesh).

For converting from TIN to grid the heights of the grid points are interpolated in the TIN and stored in the grid. The DTM program Surfer [Waterloo Hydrogeologic, 2002], for example, provides this possibility.

## 2.2 Global and local approaches

A global approach uses algorithms which consider all given points at once and derives a surface, or an intermediate product, in one step. Because of this, the height at a specific location depends on the heights of all given points. Drastically speaking, for terrain modelling this has the consequence, that the height of a measured point in one valley influences the interpolated heights in a neighboring valley. Especially if a method is susceptible to oscillatory behavior (like polynomial interpolation) this prohibits its use in terrain modelling. If, on the other hand, the influence of a point decreases with increasing distance and becomes practically zero this can very well be accepted. The term ‘practically zero’ means a very small value, so that the influence of a measurement (e.g. the height change caused by the omission of this point) at a certain distance is for example below the measurement accuracy. Some methods for height determination with global behavior are kriging, linear prediction and FEM as described in [Ebner and Reiß, 1978]. Also inverse distance weighting is global, if the weight function has asymptotic behavior, i.e. if the influence of a point tends asymptotically to zero, as its distance from the point to be interpolated increases.

In local approaches the height at a specific location is only determined by the measurements within a certain neighborhood. This neighborhood can be determined by Euclidean distance or – in the case of a triangulation – e.g. by the vertices of the triangle which circumscribes the point. One method belonging to the group of local methods described above is taking the median of measured heights in a raster cell as cell height. Also moving least squares and inverse distance weighting belong to the group, if the weight function becomes zero for a distance smaller than the diameter of the study area. Eventually, all the methods described above for the interpolation of a height within one mesh of a grid model, and also the Delaunay triangulation are local methods.

Sometimes, global algorithms are applied in a local fashion. For this end the entire region is divided (i.e. split up) into smaller areas and in each of these areas a global algorithm for determining the terrain model is applied. At the borders of these areas the results have to be mixed in some way to provide continuity or even smoothness of the height values. To keep these problems small, the areas are augmented by an overlap reaching into the neighboring areas. The overlap should be large enough to guarantee that the points outside of the overlapping area have practically no influence on the surface within the ‘original’ (i.e. not augmented) area. Of course, severe problems arise, if extrapolation takes place at the borders. An example for this approach is SCOP, where the simple kriging system<sup>2</sup> is solved for rectangular patches of equal size and a number of about 1000 points per area (including the overlap).

The same strategy as above is often applied for local algorithms in order to keep the number of points in memory small. Of course, again an overlap is necessary.

## 2.3 Models in 2.5D and in 3D

In the first case (2.5D), which is usually used, the terrain model is a function where a point is mapped from the parameter domain to one real value. The parameter domain are either ground plan co-ordinates, or co-ordinates in a specific plane forming a local co-ordinate system, a state wide co-ordinate system or geographical co-ordinates (longitude and latitude), which is most often used for planet-wide elevation models.

The 2.5D case is also called  $2\frac{1}{2}$ D case and 2D+1D case<sup>3</sup>. The surface is a set of points  $(x \ y \ z)^\top$  with

$$z = f(x, y)$$

in the explicit form with function  $f$  over domain  $D \subset \mathbb{R}^2$ . In the implicit form it is given as set of zeros  $f(x, y) - z = \tilde{f}(x, y, z) = 0$ . Of course, the explicit form is much more practicable for the analysis of terrain elevation.

---

<sup>2</sup>Simple kriging is kriging where the expectancy of the spatial random variable is zero.

<sup>3</sup>In [Kraus, 2000] the term 2D+1D is, however, used for cases where the dependency of the function value (i.e. ‘z’) on the metrical ground plan co-ordinates is less strong. In a topographic information system with layer structure the function value is not necessarily measured in length units, but may indicate other attributes as street quality, land usage or soil pollution.

In the 3D case, the surface can be described as:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathbf{f}(u, v) = \begin{pmatrix} f_x(u, v) \\ f_y(u, v) \\ f_z(u, v) \end{pmatrix}$$

with  $(u, v) \in D$ , and the domain  $D$  again a subset of  $\mathbb{R}^2$ . (Note, that  $f_x$  does not denote a derivative, but the  $x$ -component of the function  $\mathbf{f}$ .) If some measurement (points and lines)  $\mathbf{P}_i$  are given, they are only determined with their location in 3D space, but usually no parameterization  $\mathbf{U}_i = (u_i \ v_i)$  is available with the measurement of  $\mathbf{P}_i$ <sup>4</sup>. After a parameterization has been computed, each of the functions  $f_x, f_y, f_z$  can be determined like in the 2.5D case. The properties of the functions are – under certain conditions – transferred to the three dimensional surface. (If  $f_x, f_y$  and  $f_z$  are smooth and the derivatives  $\partial\mathbf{f}/\partial u$  and  $\partial\mathbf{f}/\partial v$  are linearly independent, then  $\mathbf{f}$  is smooth.) Complementing to what was mentioned in the previous section on bivariate elevation functions, the 2.5D case corresponds to a *graph of a bivariate function* (or scalar-valued bivariate function) whereas the 3D case corresponds to a vector-valued bivariate function.

However, it is not always necessary to compute a parameterization even if the data is fully 3D. If the points are given in a 3D triangulation, this triangulation can substitute the need for a parameterization. It can be said that the surface is parameterized over the triangulation. Points which are close to each other in the parameter domain are also close to each other on the surface (in 3D space). However, two points with a very small Euclidean distance in 3D space are not necessarily close to each other measured along the surface and in the parameter domain. If it is sufficient to know the neighborhood of a point it is therefore also sufficient if a triangulation of the data is available. The surface can then be parameterized locally over each triangle.

If a triangulation is already given, a parameterization of the points can be computed with different algorithms. Some algorithms and references are presented in Sec. 3.2. For obtaining a 3D triangulation of the data, different algorithms exist:

If the data are given as contour lines, as in [Crippa et al., 1998], the triangulation can be performed with ‘simpler’ algorithms. Adjacent points on the contour lines have to form a triangle edge. The remaining two edges for each triangle have to connect to a contour line point of a consecutive height. A different approach is presented in [Choi et al., 1988] which requires one point from which all the data is visible. In photogrammetric applications this could e.g. be the position of the camera. For airborne laser scanning [Verbree and van Oosterom, 2001] describe an algorithm which proceeds by building a tetrahedralization first. Next, tetrahedrons above the ground are removed, using knowledge on the flying path: every measured point must be visible from the position the airplane (the sensor) had, when the point was measured.

Another approach is taken in [Mencl, 1995] which constructs first a tree where each point is connected to at least one other point of the entire point set<sup>5</sup>. Edges are inserted consecutively, until a triangulation is obtained. A set of rules is used to find which edges may be inserted and which must not be inserted. A somehow opposite approach is taken in [Kós, 2001]. First, a neighborhood graph is built: every point is connected to its neighbors according to some close-ness criterion (e.g. Euclidean distance). Then, the triangulation is extracted from this graph.

In [Heitzinger, 1999] first a tetrahedralization is built, then the triangulated surface is extracted from it, again using a rule-based approach. A method based on the Delaunay tetrahedralization of the given points augmented by the vertices of their Voronoï diagram is given in [Amenta et al., 1998].

Terrain modelling in 3D based on a 3D triangulation has been proposed in [Schlüter, 1999] for object reconstruction from images but it has also been embedded in a more general context by [Heitzinger et al., 1996] and [Pfeifer and Pottmann, 1996].

<sup>4</sup>Terrestrial laser scanning forms one exception. The measurements of one positioning are usually performed in a spherical coordinate system. The points  $\mathbf{P}_i = (x_i \ y_i \ z_i)^\top$  have therefore parameterizations  $\mathbf{U}_i = (\alpha_i \ \beta_i)^\top$  with  $\alpha$  and  $\beta$  as angles of the measurement.

<sup>5</sup>Precisely, each point is connected to its nearest neighbor. This graph is called the Euclidean minimum spanning tree.

## 2.4 3D terrain models

The propositions which are underlying the search for appropriate models of the terrain in 3D are the following:

1. The surface shall be smooth.
2. The surface shall not be restricted to the graph of a bivariate function.
3. Structural information (breaklines, peaks, ...) shall be considered.

Proposition 1 and 3 are contradictory at first sight. If breaklines are in the terrain, than the terrain is – in the mathematical sense – not differentiable. It is, however, smooth everywhere except at the breaklines. In the following the term smooth is also to be understood as ‘smooth except at the prescribed breaklines’.

Additional requirements arise from practical aspects and the available computer environment.

- Local algorithms should be applied.
- One data structure shall be used for the complete model.
- Random measurement errors must be eliminated from the data.

**Smoothness** Proposition one is deduced from observations in nature. The terrain is – in general – smooth. Exceptions occur along special lines which are treated in proposition three. Another source of exceptions is formed by special points, e.g. peaks of mountains or on the bottom of dolinas in karst regions. If higher derivatives also vary smoothly or not can be argued. Various natural processes (wind, water, rock movements, ...) form the terrain, and in some areas these may well add to terrain forms where the curvature varies continuously. While from the theoretic point of view a curvature continuous terrain model might be justified in some regions, eventually the applications of the model define the required properties in the end. Smoothness of the first derivative is e.g. necessary for ortho photo production. Otherwise, linear features like streets would show breaks across their axis at the points of lacking smoothness, making an ortho photo map harder to read. Of course, the list of applications requiring smoothness can be extended. On the other hand, the application can also be used to define the level of smoothness required. While smoothness means mathematically the continuity of the first derivatives, it can also be defined via the level of allowed ‘un-smoothness’, which remains without negative consequences for a certain application. Isolines may have e.g. breaks of up to  $2^\circ$  without disturbing a human interpreter.

However, the smoothness proposition rules out a surface model which consists of planar faces spanned between the originally measured points. Concerning volumetric models it shall be noted, that the smoothness requirement applies to the boundary surface. Using a volumetric approach can therefore be helpful in finding this surface, but does not necessarily support smoothness. Moreover, voxel models, like raster models show discontinuities at the cell borders. B-Rep models do not have necessarily planar faces, but finding curved surfaces for each face in a way that these patches join smoothly along their boundary curves is even more complex than for triangular faces. This is because each face may have an arbitrary number of vertices ( $\geq 3$ ) and inner loops (holes). Of course, each B-Rep description of an object can be converted into a triangulation of the object. If a triangulation is used, curved patches must be determined for each triangular face in a way that these patches meet with the desired smoothness along their common boundary curves. If the surface is differentiable of degree  $n$  for the whole domain, the surface is said to have  $C^n$  continuity ( $C1, C2$ ). Often, this continuity is only established geometrically, which means that the surface function itself is not necessarily differentiable, but the geometric properties of the corresponding order (tangent plane, curvature elements, ...) change continuously. This is called G1 (also V1 for visual continuity and GC1) for tangent plane continuity, and G2 (...) for curvature continuity.

Smoothness can also be seen as a modelling tool. In this context smoothness of order  $n$  is prescribed for certain regions, or lines, and the computed surface has to have the specified (geometric) differentiability in the corresponding regions. It may also be possible to determine the appropriate smoothness from the data itself. However, G2 smoothness goes beyond the scope of this work, and the above suggestion will not be followed.

The smoothness proposition rules out a surface model with flat faces. The terrain model shall be G1.

**3D model** Proposition two states, that the approach must be capable of modelling overhangs and caves. In order to have a real 3D representation either a parameterization of the complete data (measurements) or a triangulation is necessary. In the first case a vector-valued function would be determined and stored e.g. in three grid models, one for each co-ordinate direction. These could, of course, also include breaklines. However, a global calculation of a parameterization is inappropriate (see below). Therefore, a triangulation of the data is necessary. This triangulation is needed in order to establish the neighborhood relations of the points and to navigate or move along the surface. However, there are some approaches which do not require a triangulation but operate on a different graph which is simpler to obtain. In this graph each point is connected to its neighbors. The neighbors must be chosen in a way, that a complete fan is built around each point, the result is a so-called fan cloud [Linsen and Prautzsch, 2001]. The fans of adjacent points may overlap. Operations like smoothing and visualizations can be performed for this fan cloud as well. However, in the context of the posed problem this would only be applicable if a very high point density (i.e. an over sampling of the surface) could be assumed.

Together with the question of the 3D abilities also the question of the genus of the surface arises. A surface of genus zero is topologically equivalent to a sphere or an open disc, depending on the surface being closed or not. A surface of genus one is equivalent to a torus, and generally, a surface of genus  $g$  is equivalent to a sphere with  $g$  handles. A 2.5D surface is always of genus zero. If a surface, and therefore also a triangulation of this surface, is of genus zero, than that triangulation is a planar graph<sup>6</sup>. Planar graphs are easier to handle, for example the algorithms referenced in Sec. 3.2 (parameterization of a triangulation) require that the graphs are planar as does the algorithm in [Choi et al., 1988] (3D triangulation where one point must exist, from which all surface points can be seen). For the algorithms developed in the following chapters no restriction is posed, that the graph is planar as a whole. It is, however, assumed that the subsets of the graph, which are used for local computations, are always planar.

The 3D proposition rules out raster models.

**Structural information** Proposition three requires that breaklines, but also special points like peaks (...) are considered in the data model and in the algorithmic approach. To some extent breaklines and peaks can be modelled implicitly by a very dense sampling. In a triangulation the point density next to a breakline could for example be increased to such a high level, that a rendered perspective view leaves the observer with a perfect impression of the breakline. However, often it is required that the information is available explicitly, not only implicitly. Additionally, knowing the breaklines explicitly allows to handle only the relevant and therefore fewer points, which results in smaller data sets and (hopefully) faster algorithms.

What is more, breaklines and peaks are not the only terrain features of interest. Sometimes a mountain peak may be described by the fact, that it is the highest point in a neighborhood, sometimes it may also show discontinuities in the first derivative. In the first case this can be described by a horizontal tangent plane, in the second case the smoothness requirement is suspended for one point. Horizontal tangent planes can also be found at saddle points. Other features are river lines, which have the property of monotonously decreasing height values – at least if an orthometric height system<sup>7</sup> is used. If a contour line is given, this does not only describe the height at a location, but also the  $xy$ -direction of the terrain gradient, which is orthogonal to the contour line.

So-called *form lines* are ‘soft versions’ of breaklines. They can be found between adjacent slopes of a mountain. The geometric description of form lines is – to some extent – unsettled. In *one* description breaklines are defined via the contour lines. At the intersection with the form line the contour lines show a local maximum in curvature. According to this definition, form lines are not necessarily gradient lines. Second order derivatives are necessary for their description. *Another* possibility is to consider water sheds. Ideally, the water always runs down along the gradient. A form line is then a special gradient line to which adjacent gradient lines converge asymptotically. In this case the contour lines are orthogonal to the form lines. A *third* definition for a form line is based on the curvature of the surface in the form line points. It states that the normal curvature in

<sup>6</sup>A graph is planar, if it can be embedded in (mapped into) the plane in such a way, that no two edges intersect, except at a vertex that these two edges have in common [Wilson and Beineke, 1979].

<sup>7</sup>In an orthometric height system points with equal gravity potential have the same height.

the direction of the form line is zero (parabolic surface point) or at least very small compared the curvatures in other directions.

This requirement rules out voxel and raster models, but enforces the use of a hybrid grid (Fig. 2.3) or a triangulation (Fig. 2.6). The algorithms for determining the terrain model must be capable of fulfilling geometric boundary conditions (e.g. a tangent plane).

**Locality** The first additional requirement states that local algorithms are to be preferred to global algorithms. It is known that global algorithms produce ‘nicer’ results than local ones. Considering all points at once produces a more homogeneous result, than considering each point and its neighborhood on its own. However, with the large data sets in terrain modelling the computation time and memory requirements would be too high. Fast growing computing power and memory supply on the one hand are compensated by advances in sensor technique and an increase in measurement density on the other hand. Likewise, the demands on accuracy and detailed terrain description are growing, too. Additionally, a local approach offers the possibility to change one part of the model without the need to recompute the whole model.

As mentioned above, this requirement prohibits the possibility of computing a global parameterization of the given point set.

**One data structure** The second additional requirement arises from practical considerations. Having to deal with different data structures makes computer programs more complex and prone to errors. Additional problems arise at the boundaries of the different regions. This is also an argument against a 3D model which is built from multiple regions with 2.5D models, where each of these models has its own reference plane. Not only the separation of the data set in appropriate 2.5D regions is an additional problem but also the smoothness requirements along the boundaries would have to be solved.

**Filtering** Additional requirement three deals with measurement errors. It arises from practical experience and is not necessarily connected to the problem of finding a way to model the terrain. For judging the results it is however necessary to operate on filtered point sets. Otherwise, the shape of the surface is not influenced by the algorithms alone, but also by the random measurement errors.

If filtering is applied, the surface does not run through the originally measured points, but through the ‘filtered versions’ of these points. This means, that the co-ordinates of the point are changed. The vector from the measured to the filtered point is called the residual vector. Often only the length of this vector is denoted as residual, with positive sign, if the original point is below the surface, and negative otherwise. The negative of the residual is called the filter value [Kraus, 2000].

Filtering should always be performed in a qualified way, which means that the residuals or filter values should be in the range of the measurement accuracy. If a gaussian normal distribution of the random measurement errors is assumed, this means that 66% of the filter values should be within the measurement accuracy, 95% within the double measurement accuracy (, ...). Linear prediction – just like kriging – offers this possibility [Kraus and Mikhail, 1972].

### 2.4.1 Problem definition

Taking the propositions and the practical requirements into account the solution is to base a G1 3D terrain model on a triangulation. Methods have to be found which provide a smooth surface on this basis, interpolating the measured points after their random errors have been filtered. Furthermore, these methods must have a geometric foundation to allow the formulation and interpolation of special terrain features, i.e. the so-called structural information.



## Chapter 3

# Algorithms for Triangulations

This chapter describes methods and concepts necessary for the modelling of surfaces in 3D space on the basis of a triangulation. Special consideration will be given to the algorithms required in topographic surface modelling, though the algorithms themselves are not tailored especially for surface modelling or topographic applications. Not all algorithms explained here require that the points are given together with their neighborhood relations. However, the need for a triangulation in these cases arises because the algorithms are restricted to a limited neighborhood of a point<sup>1</sup>. Thus, the first sections take a closer look on the definition of ‘neighbored’ points. Next, the parameterization of triangulations is treated. Methods are described to map a triangulation into the plane. This is followed by algorithms to determine the tangent plane (i.e. the normal vector) at a vertex. This can also be seen as the determination of an approximating surface of degree one. Consecutively, methods for the approximation of the triangulation by surfaces of higher degree are described. This means, that the underlying unknown surface which is represented by the triangulation is approximated better and better. A section on quadratic functionals explains how to compute the surface energy and other measures of a triangulation. Mesh improvement, i.e. the enhancement of the given triangulation but also its simplification is treated in Sec. 3.5. The section on the filtering of random measurement errors in the triangulated surface points and the section on the consideration of breakline information conclude this chapter with a more topographic emphasis.

Many of the algorithms require that the triangulation is a planar graph. As mentioned in chapter 2 this means, that the genus of the triangulated surface is zero, i.e. the surface is topologically equivalent to a sphere. If a bridge is modelled together with the valley it crosses as well as the neighboring terrain, then the planarity requirement is violated. Algorithms for determining whether a graph is planar or not are referenced in [Wilson and Beineke, 1979]. In the following it is assumed, that the triangulation is a planar graph. For most of the algorithms it is sufficient, that the triangulation of a certain neighborhood of a point is a planar graph.

### 3.1 Definition of neighborhood

The neighborhood of a specific point  $\mathbf{P}$  is a set of points which are – with respect to a certain norm – close to the specific point. Normally, this neighborhood is defined via Euclidean distances, but if the points are given in a triangulation, ‘topological norms’ can be defined, too. A point is a neighbor of point  $\mathbf{P}$  of generation  $n$ , if it is connected with the specific point by at least  $n$  edges<sup>2</sup> (Fig. 3.1, middle). The neighborhood of generation one is also called 1-ring neighborhood. Another possibility is to define a *star shaped* neighborhood. The points belonging to the first generation neighborhood (dotted (green) line, Fig. 3.1, right) are augmented by those points, which belong to a triangle having an edge which connects two adjacent points of generation one.

<sup>1</sup>Of course, a triangulation is not the only data structure, that allows to access the neighbors of a certain point. In an octree, for example, a neighborhood of a point can be given by the enclosing cell and its neighboring cells. A fan cloud (see p. 18) is another possibility.

<sup>2</sup>In graph theory two vertices have distance  $n$ , if the shortest paths between these two points have length (i.e. number of edges)  $n$ .

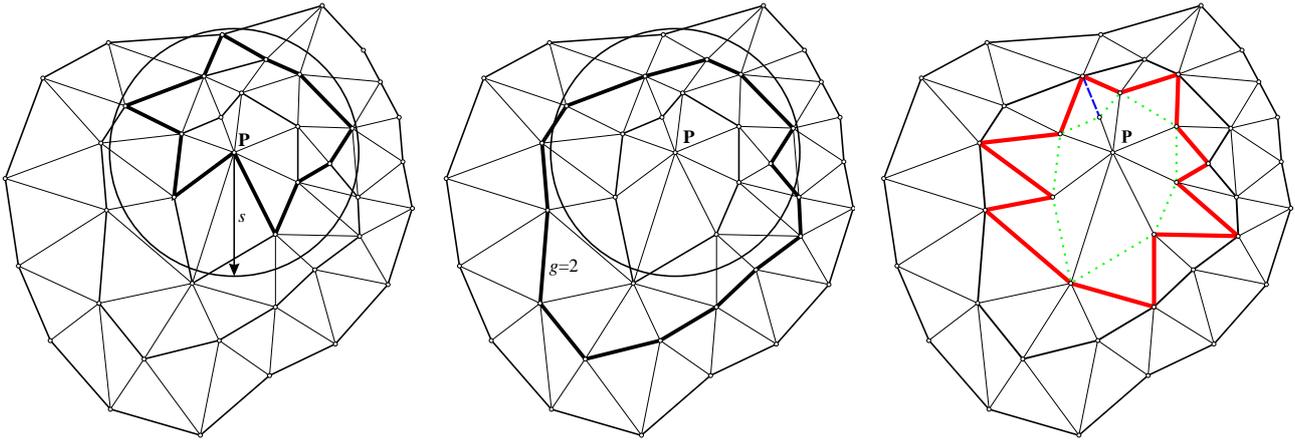


Figure 3.1: Neighbors of  $P$ , Left: within Euclidean distance  $s$ , Middle: neighbors of generation 2 (circle only for comparison), Right: star-shaped neighborhood

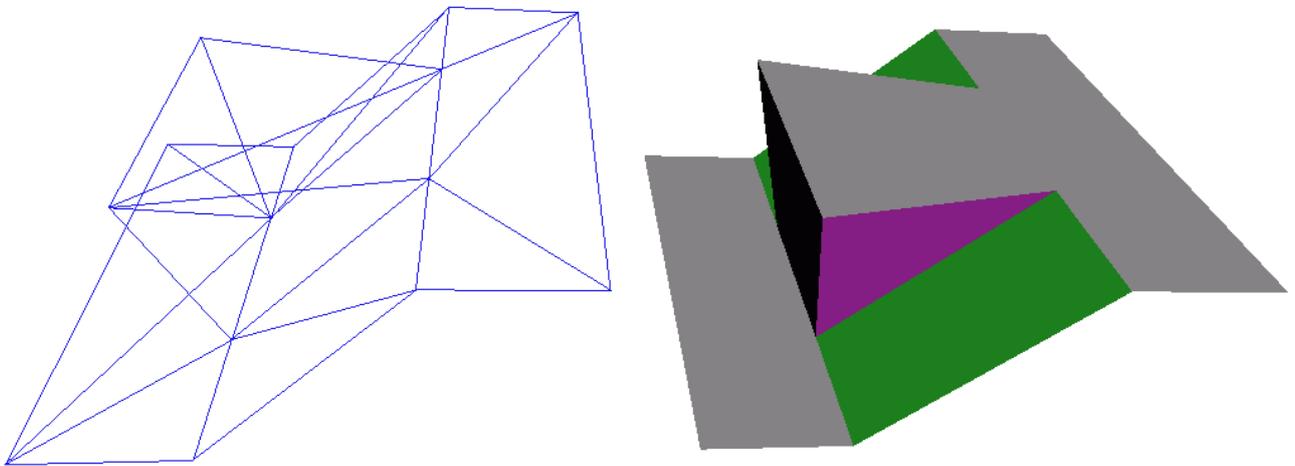


Figure 3.2: Synthetic data of an overhang in perspective views. Left: wire frame of the triangulation, Right: faces with the same normal vector are shown in the same color. Three faces (four triangles) of the terrain are overhanging.

The outline of the star is shown as fat line (in red) in the figure. A difference between the neighborhood of generation two and the star shaped neighborhood can be found at those first generation vertices where the valency (number of emanating edges) is larger than five or smaller than four<sup>3</sup>. The first generation neighborhood is a subset of the star shaped neighborhood which is a subset of the generation two neighborhood. In a regular triangulation, where each vertex has six neighbors, the number of neighbors for these three types of neighborhoods is 6, 12, and 18, respectively. For the star shaped neighborhood also the term neighborhood of generation one and a half will be employed. In this work, no neighborhoods beyond the second generation will be used.

For the Euclidean norm the space in which the distance is computed has to be chosen first. In the 2D and the 2.5D case this can be the distance in the ground plan, as it is shown in Fig. 3.1 on the left side. However, for 3D triangulations, the ‘normal’ Euclidean distance measured in 3D space between two points can be very small, though these points are not neighbors on the surface. This can occur e.g. in a cave, where the ceiling and the floor run parallel to each other with a small height difference. In this case the distance has to be measured along

<sup>3</sup>The term ‘star shaped’ may not be optimal, because the union of triangles which have one vertex in common (generation one neighbors) is sometimes also referred to as ‘star’. Furthermore, in a 2D star polygon at least one point exists, which can be connected to every other point in the polygon by a line segment that lies entirely within the polygon.

the surface defined by the triangulation. A method to compute path length along a triangulated polyhedron is given in [Opitz and Pottmann, 1994].

## 3.2 Parameterization of triangulations

In Sec. 2.3 the need to parameterize triangulations has already been mentioned. Figuratively this can be imagined as pressing the triangulation into a plane. Of course, the triangles will be distorted in this process, at least if they are not on a developable surface. Additionally, the triangles must not overlap in the parameterization. Formulated more mathematically, finding a parameterization of a triangulation with the points  $\mathbf{P}_i \in \mathbb{R}^3$  is equivalent to finding a map  $M : \mathbf{P}_i \mapsto \mathbf{U}_i$ , with  $\mathbf{U}_i \in \mathbb{R}^2$ . This maps the original points to the parameterization plane. An edge  $e_{ij}$  which connects  $\mathbf{P}_i$  and  $\mathbf{P}_j$  also connects  $\mathbf{U}_i$  and  $\mathbf{U}_j$ . Again, the triangles formed by three pairwise connected points in the parameterization plane must not overlap, i.e. edges between the vertices must not cross. In the map (but also in the triangulation) the edges can be classified as inner edges and outer edges. Outer edges are adjacent to only one triangle, they form the boundary of the polygon in the parameterization plane. The corresponding edges between those  $\mathbf{P}_i$  form the boundary of the triangulation. Boundary edges connect outer points, all other points are called inner points.

For many applications it is not necessary to consider the graph (edges and points) in the parameterization plane, but only its vertices. In [Floater and Reimers, 2001] a method for finding a parameterization of a (planar) point set is given which does not require a triangulation. It is found by solving a global equation system.

### 3.2.1 Projection onto a plane

Often, the points can be parameterized by projection into a plane. In the 2.5D surface case, the mapping is very simple:  $M : \mathbf{P}_i \mapsto \mathbf{U}_i$  with  $\mathbf{P}_i = (x_i \ y_i \ z_i)^\top$  and  $\mathbf{U}_i = (x_i \ y_i)^\top$ . This is an orthogonal projection of the points into the plane  $z = 0$ . For 3D surfaces this is in general not possible. An example for this is the overhang shown in Fig. 3.2, where the bright faces (in gray) are horizontal. The triangular face almost parallel to the paper plane (in magenta), the dark faces (two triangles) and another face (not visible) are overhanging surface parts.

### 3.2.2 Local projection onto a plane

For point sets in a real 3D surface it is often possible to parameterize the points locally, i.e. in a certain restricted neighborhood of the point  $\mathbf{P}$ , by projection into a plane. For this, the points are projected orthogonally onto this plane. One possible plane for this projection is the tangent plane in  $\mathbf{P}$ . Of course, the tangent plane is generally not known, thus an approximation has to be used, methods for its computation are given below (Sec. 3.3.1). If the plane goes through  $\mathbf{P}$  with unit normal vector  $\mathbf{n}$ , then the formula for the projection  $\mathbf{Q}_p$  of a point  $\mathbf{Q}$  onto the plane is:

$$\mathbf{Q}_p = \mathbf{Q} - ((\mathbf{Q} - \mathbf{P})^\top \mathbf{n}) \mathbf{n} \quad (3.1)$$

Of course, this provides only the projection, but no parameterization is computed, yet. For this end an  $uv$ -co-ordinate system has to be spanned in the plane ( $\mathbf{n}^\top \mathbf{u} = 0$ ,  $\mathbf{n}^\top \mathbf{v} = 0$ ). It is possible to specify the  $\mathbf{u}$ -vector by demanding, that one of its co-ordinates is zero. If that co-ordinate is chosen which has the highest absolute value in  $\mathbf{n}$ , there is always a solution. The vector  $\mathbf{u}$  is normalized and  $\mathbf{v}$  is obtained by  $\mathbf{n} \wedge \mathbf{u}$ . The parameterization of  $\mathbf{Q}$  is the point:

$$\mathbf{U}_Q = (u_Q, v_Q)^\top = ((\mathbf{Q} - \mathbf{P})^\top \mathbf{u}, (\mathbf{Q} - \mathbf{P})^\top \mathbf{v})^\top = (\mathbf{u}, \mathbf{v})^\top (\mathbf{Q} - \mathbf{P}) \quad (3.2)$$

The height of the original point over the parameterization plane (a scalar value) is:

$$w_Q = \mathbf{n}^\top (\mathbf{Q} - \mathbf{P}) \quad (3.3)$$

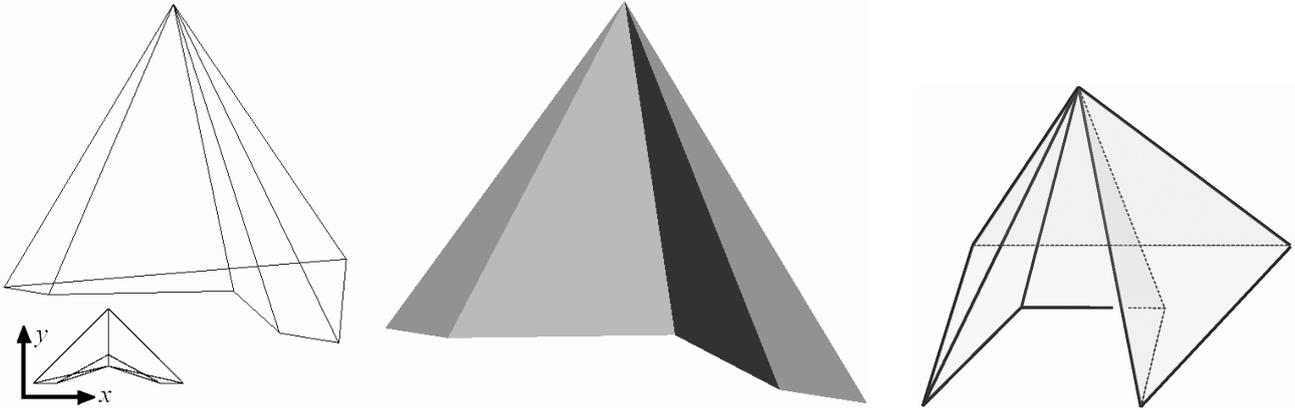


Figure 3.3: Left and middle: Surface where the parameterization by projection onto a plane is still possible. A wire frame model and the faces are shown in a perspective view, in the lower left a ground sketch of the wire frame model is shown. Right: A triangulation of one vertex and its neighbors which cannot be parameterized by projection onto a plane.

For larger areas a parameterization over the tangent plane is not always possible. For the overhang example it is not possible for the left foreground corner point to parameterize the neighbors up to generation 2 over a horizontal plane, though it might be a good approximation of the tangent plane in this point. Even if only the point and its first generation neighbors are considered, this kind of parameterization is not always possible.

An extreme example is shown in Fig. 3.3 (left), where a parameterization above a plane is still possible. However, the projection plane has to be chosen very carefully, so that the edges between the points in the orthogonal projection do not cross<sup>4</sup>. Though this example may be considered as a pathological case which does not occur in practice, the problem is obvious and remains. This situation may also appear because of small data errors. If neighbors of higher generations (two, ...) are considered, the problem becomes more and more severe. Of course, it is possible to detect these cases, but still a different form of parameterization would have to be searched for these special cases.

An algorithm specified in [Opitz and Pottmann, 1994] proceeds as follows. The triangulation is considered as a polyhedron. To parameterize the neighborhood of  $\mathbf{P}$ , first a tangent plane  $\tau$  has to be estimated, and  $\mathbf{P}$  is mapped onto itself ( $\mathbf{U} = \mathbf{P}$ ). For each point  $\mathbf{P}_i$  of the neighborhood of  $\mathbf{P}$  the shortest path along the polyhedron is determined. This is a polygon with vertices at the edges of the polyhedron. The point  $\mathbf{P}_i$  is mapped (parameterized) by developing the polygon starting in  $\mathbf{P}$  onto  $\tau$ . The starting segment of the polygon is projected orthogonally on  $\tau$ . This defines the direction of a line segment starting in  $\mathbf{P}$ . Its length is set to the length of the polygon. The end point of the line segment is  $\mathbf{U}_i$ , the map of  $\mathbf{P}_i$  into the parameterization plane. This is an estimation of the so-called exponential map. The polygon and the line segment are both geodesic curves for the corresponding surfaces, the tetrahedron and the plane, respectively. If there is more than one shortest polygon connecting  $\mathbf{P}$  to  $\mathbf{P}_i$ , than  $\mathbf{P}_i$  has multiple maps in  $\tau$ . This method does not necessarily produce a correct parameterization of the triangulation, because the edges between the vertices may cross. However, for this method the aim is not to draw the triangulation in a plane but to find a local parameterization of the neighboring points of  $\mathbf{P}$ .

### 3.2.3 Global parameterizations

A different approach is to compute the parameterization of a planar triangulation globally, which means that one equation system is solved, and as a solution the locations  $\mathbf{U}_i$  in the parameterization plane are found. The

<sup>4</sup>It is not hard to construct an example where they do cross. If the neighbors of generation one lie in a common plane, then the edges between them form a polygon. If no point exists in the interior of this polygon from which all other vertices are 'visible' (i.e. the line segment from this point to the vertex lies in the interior of the polygon), then an arbitrary point can be chosen as the vertex of interest (generation zero). In any orthogonal projection of this triangulation the edges will cross.

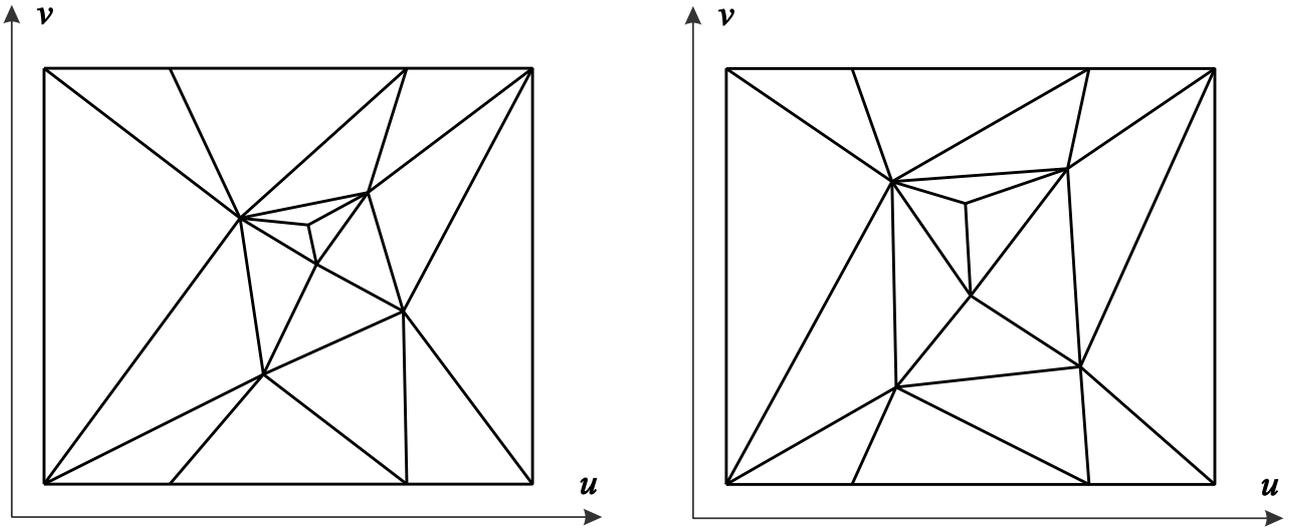


Figure 3.4: Parameterization of the triangulation of the overhang (Fig. 3.2). Left the barycentric mapping is shown, right the weighted least squares mapping with  $q = 2$ .

methods investigated have in common, that the boundary of the triangulation is mapped to a convex polygon first. This convex polygon can be specified arbitrarily, only the sequence of the vertices must be the same as in the triangulation. This is of course necessary, because the edges must not cross.

In the so-called ‘barycentric mapping’ [Tutte, 1963] (according to [Floater, 1997]) first the vertices of the boundary are mapped to a convex polygon in the parameterization plane. Then, each inner point is determined to be the barycenter (center of gravity) of its neighboring vertices.

$$\begin{aligned} \mathbf{U}_i &= \frac{1}{d_i} \sum_{j=1}^n \lambda_{ij} \mathbf{U}_j & \mathbf{U}_i \text{ refers only to inner points} & \quad (3.4) \\ \lambda_{ij} &= \begin{cases} 1 & \dots & \text{if } \mathbf{P}_i \text{ and } \mathbf{P}_j \text{ are joined by an edge (i.e. neighbored)} \\ 0 & \dots & \text{otherwise} \end{cases} \end{aligned}$$

Here,  $d_i$  is the valency of vertex  $\mathbf{P}_i$ , and  $n$  is the number of vertices in the triangulation. The  $\lambda_{ij}$  are the entries of the adjacency matrix. This leads to a linear system of equations which has a unique solution. For the example of the overhang (Fig. 3.2) the parameterization after this method is shown in Fig. 3.4. It shall be noted, that this method is not restricted to triangulations, but can be applied to any simple<sup>5</sup> planar graph.

Different methods are specified in [Floater, 1997]. Again, the  $\mathbf{U}_i$  have to be defined for the boundary vertices first. As in Eq. 3.4 each inner point is a *convex combination* of its neighbors (the weights  $w_{ij} = \lambda_{ij}/d_i$  are all positive and sum up to one for a given point index  $i$ ). If the weights are chosen to be multiples (factor  $\mu$ ) of  $w_{ij} = 1/\|\mathbf{P}_i - \mathbf{P}_j\|^q$  a family of parameterizations is obtained. The factor  $\mu$  has to be chosen in a way so that  $\mu \sum_j w_{ij} = 1$  (the sum over the weights associated with the edges emanating from  $\mathbf{P}_i$ ). By rearranging the formulas, it can be shown that this is equivalent to minimizing a functional over the inner points:

$$F(\mathbf{U}_1, \dots, \mathbf{U}_n) = \sum_{i,j} w_{ij} \|\mathbf{U}_i - \mathbf{U}_j\|^2 \quad (3.5)$$

In Fig. 3.4 the parameterization for  $q = 2$  can be seen. Because a quadratic form is minimized it is called least squares parameterization.

A different setting of the  $w_{ij}$  is found by first constructing a local parameterization for each point  $\mathbf{P}$  and its first generation neighbors. This local parameterization is obtained by ‘emulating the so-called *geodesic map*,

<sup>5</sup>A graph is simple, if it contains no multiple edges and no loops (edges which join a vertex to itself) [Wilson and Beineke, 1979].

[...] which preserves arc length in each radial direction' [Floater, 1997]. If the neighbors of  $\mathbf{P}$  are called  $\mathbf{P}_i$  ( $i = 1, \dots, d$ ), then the sum of the triangle angles with the vertex  $\mathbf{P}$  can be written as

$$\theta = \sum_{i=1}^d \text{angle}(\mathbf{P}_i, \mathbf{P}, \mathbf{P}_{i+1}) \quad (3.6)$$

Here, the function  $\text{angle}(\mathbf{R}, \mathbf{S}, \mathbf{T})$  denotes the angle between the vectors  $\mathbf{R} - \mathbf{S}$  and  $\mathbf{T} - \mathbf{S}$ . (Additionally,  $\mathbf{P}_{d+1} = \mathbf{P}_1$ ,  $d$ =valency.) The map of the  $\mathbf{P}_i$  are then found with:

$$\begin{aligned} \|\mathbf{U} - \mathbf{U}_i\| &= \|\mathbf{P} - \mathbf{P}_i\| \\ \text{angle}(\mathbf{U}_i, \mathbf{U}, \mathbf{U}_{i+1}) &= \frac{2\pi}{\theta} \text{angle}(\mathbf{P}_i, \mathbf{P}, \mathbf{P}_{i+1}) \end{aligned} \quad (3.7)$$

This determines the map apart from a translation and a rotation, but these are irrelevant for the following. Next a convex combination for  $\mathbf{U}$  depending on all  $\mathbf{U}_i$  is determined in this map. This is repeated separately for each point and in this way all  $w_{ij}$  are determined which are used to solve an equation system similar to Eq. 3.4. Because only the geodesic map is of interest for the following, the interested reader is referred to [Floater, 1997] for details on the determination of the  $w_{ij}$ .

In [Eck et al., 1995] a different method for setting the  $w_{ij}$  of Eq. 3.5 is discussed. An overview on global parameterization methods for triangulations can be found in [Hormann, 2001]. Some methods how to find a suitable map (i.e. parameterization) of the boundary are presented there as well.

It has to be mentioned that these methods of parameterization work only for planar graphs. In the context of the posed problems, the given graph is not necessarily planar. Additionally, the properties of the graph (i.e. its planarity, ...) are often not known in advance.

### 3.2.4 A method for local parameterization

The methods of parameterization described in the first parts of this section were projections onto a plane. The method specified in [Opitz and Pottmann, 1994] is a map into a plane which can be described geometrically: the local parameterizations over the plane start from a specific point which is the center of interest. The global methods, on the other hand, start from the boundary. In this section a parameterization for a star shaped neighborhood will be presented, ideas for it are taken from the works of [Opitz and Pottmann, 1994] and [Floater, 1997]. This method works also for parameterizing the first generation of neighbors, which is a subset of the star shaped neighborhood.

At start, the geodesic map (Sec. 3.2.3) around  $\mathbf{P}$  is constructed for the first generation neighborhood. The translation and the rotation parameter are not of interest here, too. However, for computing purposes these values have to be set and it stands to reason to specify  $\mathbf{U} = (0, 0)^T$ , and  $\mathbf{U}_1 = (\|\mathbf{P}_1 - \mathbf{P}\|, 0)^T$ .

If  $\mathbf{P}$  is an inner point the angles of the emanating triangles are stretched to the interval  $[0, 2\pi]$  as written above. If the point is a boundary point, then the angles of the emanating triangles are spread over the interval  $[0, \pi]$ . This setting is also proposed in [Lee et al., 1998]. However, if a border vertex and its neighbors lie in a common plane, than the angles should not be stretched at all in order to avoid distortions. Generally, stretching of the angles can lead to degenerate triangles. In this work always the range  $[0, \pi]$  was used for boundary points.

If the star-shaped neighborhood shall be parameterized, then further rules for the mapping of the remaining points (triangles) are necessary, as the geodesic map only provides a parameterization of the first generation neighbors. There are different possibilities for the parameterization of the 'outer' points:

- The points are chosen so that the edges from the first generation neighbors to the second generation neighbors have the same length in the triangulation and the parameterization. This is not always possible, because the triangle inequality may not be fulfilled, if the angles around the center vertex are stretched or compressed strongly.
- The points are chosen in order to construct a triangle in the parameterization plane which is similar (i.e. has the same angles) to the triangle in the triangulation.

- The ‘outer’ triangle point is parameterized by the foot of its altitude<sup>6</sup> relative to the end points of the opposing edge and the length of the altitude. These measures are used to construct the point in the parameterization.

All these methods have in common, that at (interior) neighbors of generation one which have valence four, the corresponding point of generation two has two images in the mapping (see also Fig. 3.1, blue edge). This is not necessarily bad. Of course, if a point is mapped twice, it has a higher weight in subsequent calculations. This could be handled by giving these points only half the weight of a normal point. But, on the other hand, the double image of this point reflects also, that it is reachable from the center vertex over three paths with length two<sup>7</sup>, whereas a ‘normal’ point of generation one and a half is only reachable via two such paths. If, on the other hand, an interior first generation neighbor has valence three, than also the two neighboring first generation points are connected by an edge. In this case no neighbors of generation one and a half (i.e. those which belong to the star shaped neighborhood but not to the generation one neighbors) exist for this point.

If  $P$  is an interior point, than it holds that if all neighboring points lie in a plane, any of the above parameterization methods will reproduce the triangulation in the parameter domain without distortions.

In Fig. 3.5 a parameterization of the data from the left part of Fig. 3.3 (the overhanging mountain peek) is given. The center vertex is the peek itself, its neighborhood does not reach beyond the first generation. In the right part the  $x$ -,  $y$ - and  $z$ -values, respectively, are shown above the parameterization domain.

Another example is demonstrated in Fig. 3.6. One vertex of the triangulation of the overhang (Fig. 3.2) and its star shaped neighborhood are parameterized. The center point is that end of the base of the overhang which is closer to the viewer (point C). The triangulation is shown in the left part of the figure, the three parameterization methods for the neighborhood of point C are shown in the right part. For the first generation the parameterizations are equal (the geodesic map). The triangles are thinner than in the triangulation, because the sum of angles in the vertex is  $7.57$  ( $434^\circ$ ) which is compressed to  $2\pi$ . This corresponds to a compression factor of 1.2.

The method where the edges to the points of generation one and a half keep their lengths is shown in green. The outer triangles are compressed (stretched, if the sum of angles is smaller than  $2\pi$ , respectively) to a similar amount as the inner triangles. The distances<sup>8</sup> to the outer points are in this case lengthened. The parameterization method with similar triangles is shown in blue. With this method the distances to the outer points are shortened, if the sum of angles in the center vertex is greater than  $2\pi$ . The parameterization by the altitudes through the outer points is shown in red. The angles at the base of the outer triangles keep their property of being above or below, respectively,  $\pi/2$  (acute and obtuse angles).

As it can be seen in the figure, the three different methods produce – in this case – similar results. This is a slight indication, that the choice of method has not a great impact on the result obtained on the basis of the different parameterizations.

### 3.3 Surface approximation and estimation of geometric properties

It can be assumed, that a triangulation – its vertices – is obtained by the sampling of a surface  $s$ . The surface is in general unknown apart from the locations of the vertices. The general aim is to reconstruct this unknown surface, but often only an approximation of  $s$  is searched. An approximation of first order is the tangent plane, at least in the vicinity of a surface point. This means, that in a vertex of the triangulation the underlying surface can be approximated to the first order by evaluating the tangent plane which touches the underlying surface. Of course, the tangent plane can only be estimated. Thus an estimated approximation of the surface  $s$  is obtained by computing a tangent plane in a vertex. This derivation is performed with a certain model of the surface in mind, e.g. that the surface is smooth, and is computed with the points in the neighborhood of the vertex. If an approximation of second order is searched, this means that an *osculating surface* to  $s$  in a vertex

<sup>6</sup>In German: „Höhenfußpunkt“

<sup>7</sup>A path is a connected sequence of edges in a graph and its length is the number of edges.

<sup>8</sup>Here, the distances on the surface, i.e. the length of the geodesic, is meant.

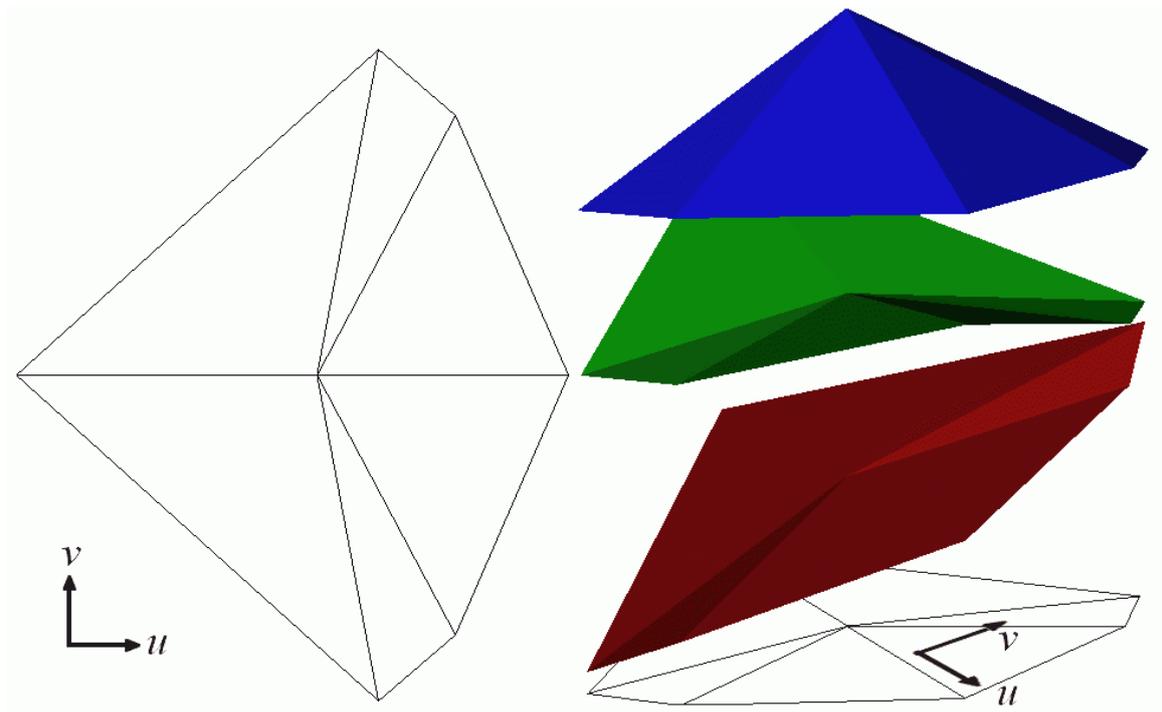


Figure 3.5: Left: Parameterization of the data from Fig. 3.3 (left) by stretching of the angles (factor =  $2\pi/3.58 \simeq 1.8$ ) in the center vertex and keeping the edge length undistorted. Right:  $x$ -,  $y$ - and  $z$ -surface shown from bottom to top (in red, green, and blue) over the parameterization. (The lower left image in Fig. 3.3 shows an orthogonal projection onto the  $xy$ -plane with the  $x$ -axis pointing right.)

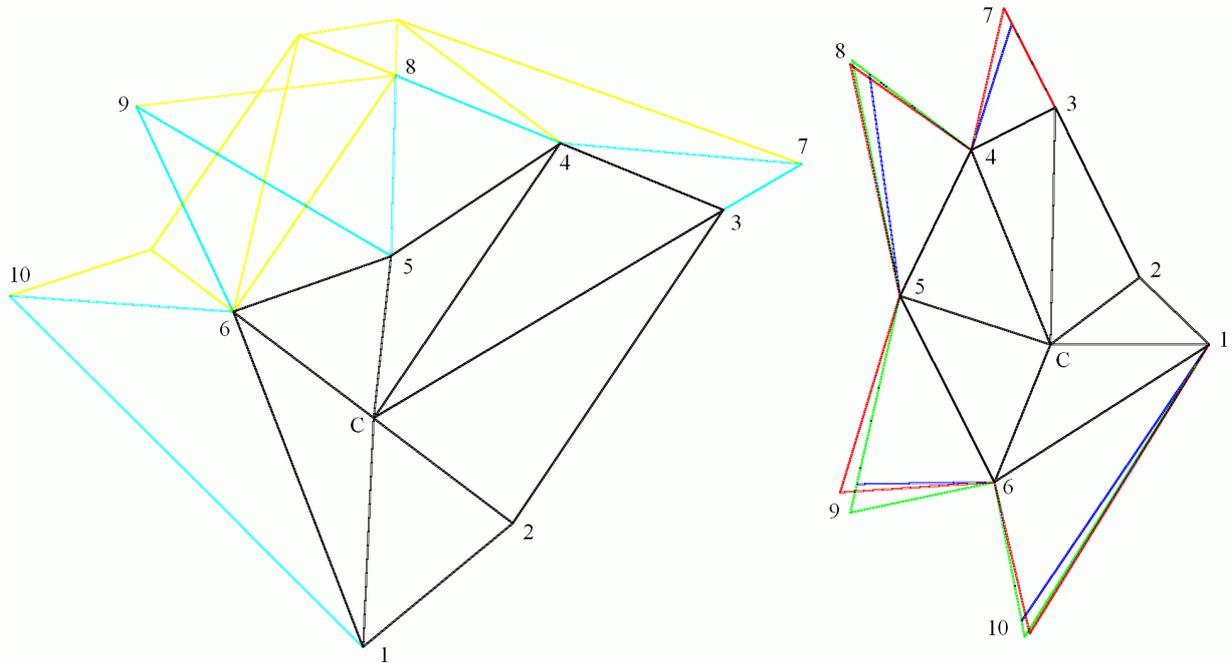


Figure 3.6: Parameterization of the star-shaped neighborhood of one point (C) from the overhang. Left: perspective view of the given data, the yellow edges do not belong to the star shaped neighborhood of C. Right: parameterization by three different methods, the inner triangles are drawn in black, those of generation one and a half have different colors according to the method. Green: edge length, blue: similar triangles, red: foot of altitude. This figure is also shown in color as fig. E.3 on page 124.

has to be found (estimated). Two surfaces are osculating, if parameterizations for both surfaces exist, which are identical up to the second order terms. Besides the identities in the point and the normal vector, this also means identity in the principal curvatures and its directions.

Often it is necessary to estimate the normal vector or the tangent plane in a node (vertex). These two are equivalent, because the tangent plane interpolates the given node and is orthogonal to the normal vector. If the approximation shall reach further (smaller distances from  $\mathbf{s}$  to its approximation), a surface of higher degree has to be used. This could be a paraboloid (obtained e.g. by Taylor expansion), a general quadric or a vector-valued second order polynomial. If the elements of curvature are to be estimated, one approach is to determine e.g. a quadric interpolating the point and use its curvature elements at the node. To estimate these properties a certain neighborhood of the node has to be used. The higher the order of the estimated surface, the larger the neighborhood must be, measured in terms of number of neighbors.

### 3.3.1 Normal vectors and tangent planes

The normal vector at a given location is an important surface property, its direction plays an important role in rendering as well as in water shed analysis, to name only two of its applications. Different algorithms have been developed for its estimation at the vertices of a triangulation. In [Meyer et al., 2002] operators are defined which are discrete versions of the respective operators in differential geometry. In other approaches the normal vector of the adjacent triangles are averaged, with weights according to the area of the triangle or the triangle angle in the vertex (e.g. [Heitzinger, 1999]). A different approach is to assume that the irregular triangulated points can be approximated locally by a simple surface like a sphere or a plane. The normal vector of this surface is then used as the normal vector in the vertex. For this end a certain neighborhood of the vertex is utilized. If the approximating surface is described in an algebraic way (as a zero set), no parameterization for the neighboring points is needed. Such algebraic surfaces can be planes or quadrics. In [Opitz and Pottmann, 1994] and [Pfeifer, 1997] a different approach was investigated. First, the neighborhood is parameterized locally over a plane which has to be a first approximation of the tangent plane. Then the multi-quadrics method and linear prediction, respectively, are used to determine an interpolating surface. The tangent plane to the interpolating surface at the vertex was used as an improved approximation of the tangent plane for the parameterization and the approximation was repeated (iterative scheme). In the shown examples the change of the normal vector after its first determination and later iterations is below  $1^\circ$ .

One technique for estimating the tangent plane within this work was based on determining an adjusting plane to the vertex of interest and its neighbors. As neighborhoods the first generation, the second generation and the star shaped neighborhood have been used. The sum of the squares of the orthogonal distances from the points to the planes were minimized. The problem can be stated as following: Given are  $n$  points  $\mathbf{P}_i$  and associated weights<sup>9</sup>  $w_i$ . The task is to find a plane  $\tau$  with normal vector  $\mathbf{n}$  and constant  $c$  for which the sum of the squares of the weighted orthogonal distances becomes a minimum.

$$\Delta = \sum_{i=1}^n (\mathbf{n}^\top \mathbf{P}_i + c)^2 w_i \quad \text{with } \|\mathbf{n}\| = 1 \quad (3.8)$$

The weights can be chosen e.g. inversely proportional to the distances from the center vertex in order to give the closer points more influence. Because these formulae are well known they are only repeated in the Appx. A.1.

This method was not used to determine the normal vector, but only for finding a plane over which the triangulation could be parameterized locally (e.g. in Sec. 3.3.3). Of course, the plane  $\tau$  would have to be shifted to interpolate the vertex of interest in order to become an estimation of the tangent plane.

It shall be mentioned that in [Meyer et al., 2002] not only methods to estimate the normal vector, but also estimators for the principal curvature and its directions directly from the triangulation are presented.

---

<sup>9</sup>If no weighting is used, then all weights are implicitly specified to be one.

### 3.3.2 Approximating quadric as local surface description

A plane, as mentioned in the previous section, can be used to determine an approximation for the tangent plane. If an approximation of higher degree is searched in a point (vertex), an approximating surface of higher degree has to be used. Such a surface can also be used to estimate the elements of curvature at the vertex of interest. The basic idea behind this is, that the points of the triangulation lie on a (unknown) surface. If the approximating surface is osculating the unknown surface, then its curvature elements are the same as the ones of the underlying surface.

Using a quadric for this task has the advantage, that no parameterization of the points is needed. A quadric is a second order algebraic surface. Planes, cones, cylinders, spheres, paraboloids, and one and two sheeted hyperboloids are all special quadrics. A quadric  $Q$  is the set of points  $\mathbf{P} = (p_x, p_y, p_z)^\top$  for which the ‘distance’  $Q(\mathbf{P})$  is zero (set of zeros):

$$Q(\mathbf{P}) = \mathbf{P}^\top \mathbf{A} \mathbf{P} + \mathbf{b}^\top \mathbf{P} + c = 0 \quad (3.9)$$

with  $\mathbf{A}$  a symmetric real-valued  $3 \times 3$  matrix,  $\mathbf{b} \in \mathbb{R}^3$  and  $c \in \mathbb{R}$ . It can be re-written:

$$(p_x \ p_y \ p_z \ 1) \begin{pmatrix} a_{11} & a_{12} & a_{13} & b_x/2 \\ & a_{22} & a_{23} & b_y/2 \\ & & a_{33} & b_z/2 \\ \text{symm.} & & & c \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = 0 \quad (3.10)$$

There are 10 values, but they are only determined up to a scale factor. Therefore, a quadric has 9 parameters, which means that at least 9 points are necessary for its determination.

In contrast to the plane, where Eq. 3.8 minimizes the orthogonal distances, inserting  $\mathbf{P}_i$  in the quadric definition (Eq. 3.9) yields algebraic distances. Only for the special case of a plane these algebraic distances can also be Euclidean distances. For a sphere and a cylinder of revolution these distances can be squares of Euclidean distances. In general the distances are measured with the scale and along the direction of  $\nabla Q(x, y, z)$ <sup>10</sup>. If  $\mathbf{P} = (x \ y \ z)^\top$  is a point of the quadric, then  $\nabla Q(x, y, z)$  is the normal vector of the quadric in  $\mathbf{P}$ .

Adjusting quadrics can be found in a way similar to adjusting planes, the formulae for it are given in Appx. A.2. In this case all points are approximated. In general this is inappropriate because the sum of squares of algebraic distance would be minimized. A different solution has been proposed in [Benkő et al., 2001]. The quadric is forced to interpolate a point  $\mathbf{P}$  and determined in a way that at this point  $\|\nabla Q\| = 1$ . This provides approximations of Euclidean distances from the quadric in the vicinity of  $\mathbf{P}$ . How to compute these quadrics is described in Appx. A.2.1. In the context of [Benkő et al., 2001] the quadric was used to get an estimation for the tangent plane.

For the computations in the following chapters the star shaped neighborhood and alternatively the neighbors up to generation two were used. In general there are not as many points as required in the first generation neighborhood to determine a quadric.

The advantage of using a quadric as an approximation of the surface in a vertex and its neighborhood (given through the triangulation) is that no parameterization is required. Additionally, the quadric is the most general surface of degree two, and it has been shown that quadrics can approximate surfaces better than with degree two, though an approximation of degree three cannot be achieved [Bol, 1967]. Especially, a fitting quadric can approximate a surface better than a paraboloid.

However, in the data which have been studied, the quadrics did not prove useful. The reason is that in flat areas, very narrow ellipsoids and two sheeted hyperboloids were found as solutions. This means, that the points were lying on different sides (e.g. different sheets) of the adjusting quadric, and therefore the quadric was no good approximation of the surface through the given point set anymore. This is illustrated schematically in Fig. 3.7. The advantage of the quadric, i.e. that no parameterization is needed, turned into a disadvantage, because the neighborhood relations were completely turned over.

<sup>10</sup>The  $\nabla$ -operator determines the gradient of  $Q$ , the operator is defined as  $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})^\top$ . For a point  $\mathbf{P} = (x \ y \ z)^\top$  the gradient  $\nabla Q(x, y, z) = (\frac{\partial Q(x, y, z)}{\partial x}, \frac{\partial Q(x, y, z)}{\partial y}, \frac{\partial Q(x, y, z)}{\partial z})^\top$ . For a plane it is possible, that  $\|\nabla Q(x, y, z)\| = 1$  and orthogonal to the plane, providing Euclidean distance measures.

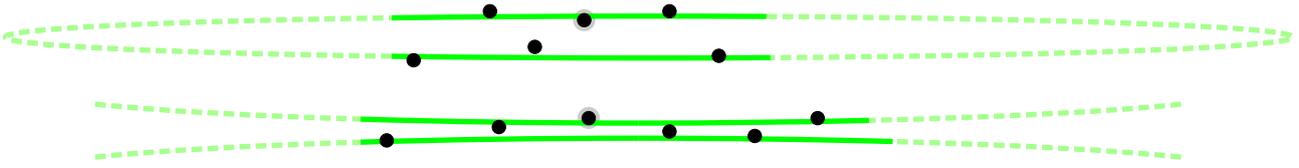


Figure 3.7: Schematic figure: If the points lie on a small part of a quadric only, then it cannot be determined reliably. Though the shown curves (representative for an ellipsoid and a hyperboloid) may be the best fitting ones, the points should be aligned from left to right on the ‘real’ terrain.

It is also mentioned in [Hartley and Zisserman, 2001] (p. 283) that problems arise if a quadric shall be estimated from a few points which only cover a small part of the quadric. In this case, it is argued, that the tangent plane can be estimated reliably, but not the quadric itself.

### 3.3.3 Approximating second order polynomial as local surface description

The motivation for determining an adjusting polynomial surface at a vertex is the same as in Sec. 3.3.2. However, one way to determine an osculating surface which definitely obeys the neighborhood relations of the given point set is to use a parametric surface instead of an algebraic surface, like in Sec. 3.3.2 (the quadric). A paraboloid is a special form of a quadric, and for each point on a smooth surface an osculating paraboloid exists. Even more general, a vector-valued second order polynomial surface<sup>11</sup> can be used as local surface description.

The first step for computing it is to define the neighborhood which shall be used and find a parameterization for it. Then the adjusting polynomial surface can be determined. Of course, individual weighting of the points is possible, too.

The formulae for a paraboloid and a general vector-valued second order polynomial surface, respectively, are:

$$\begin{aligned} q(x, y) &= ax^2 + bxy + cy^2 + dx + ey + f \\ \mathbf{q}(u, v) &= \mathbf{a}u^2 + \mathbf{b}uv + \mathbf{c}v^2 + \mathbf{d}u + \mathbf{e}v + \mathbf{f} \end{aligned}$$

With  $a, \dots, f \in \mathbb{R}$  and  $\mathbf{a}, \dots, \mathbf{f} \in \mathbb{R}^3$ . The first equation describes a scalar-valued paraboloid, parameterized over the  $xy$ -plane, the second a bivariate vector-valued second order polynomial<sup>12</sup> parameterized over  $u$  and  $v$ . Setting  $f$  and  $\mathbf{f}$ , respectively, to zero always yields a surface which interpolates the origin of the co-ordinate system.

The equations for computing these surfaces are given in Appx. A.3. A total of six points is necessary for computation, but this often leads to oscillating surfaces (unwanted undulations). Two possible counter measures are to ensure that more than six points are used, and to apply regularization. This is also treated in the appendix.

An example is shown in Fig. 3.8. In one point of the triangulation of the overhang (Fig. 3.2) a vector-valued (yellow) and a scalar-valued (red) polynomial are determined. In both cases the neighborhood reaches to the points of generation one and a half (shown as blue edges). Both surfaces interpolate the center vertex. The parameterization of the points for the scalar-valued paraboloid was computed by projecting them onto an adjusting plane. The normal vector of this reference plane is shown in black in the figure, the normal vector of the paraboloid in the center vertex is drawn in red. The paraboloid is close to most of the points, but the two on the overhang are missed completely. For the vector-valued polynomial the parameterization method

<sup>11</sup>A second order polynomial surface can be – but does not have to be – a paraboloid. Apart from congruency transformations the paraboloids are described as the set of points  $\mathbf{P} = (x, y, z)^\top \in \mathbb{R}^3$  with  $z = ax^2 + by^2$  (canonical form,  $a, b \in \mathbb{R}$ ). A paraboloid is always a second order polynomial surface.

<sup>12</sup>These surfaces are special „Steinersche Römerflächen“. The Roemer surfaces are embedded in projective space with points  $(w : x : y : z)$ . Each co-ordinate of such a surface is a quadratic polynomial in two parameters  $u, v$ . The Roemer surfaces also include the quadrics in the sense that the points  $(x/w, y/w, z/w)$  can describe them. The restriction which is applied in the following is that  $w(u, v) \equiv 1$  and therefore all points are contained in  $\mathbb{R}^3$ .

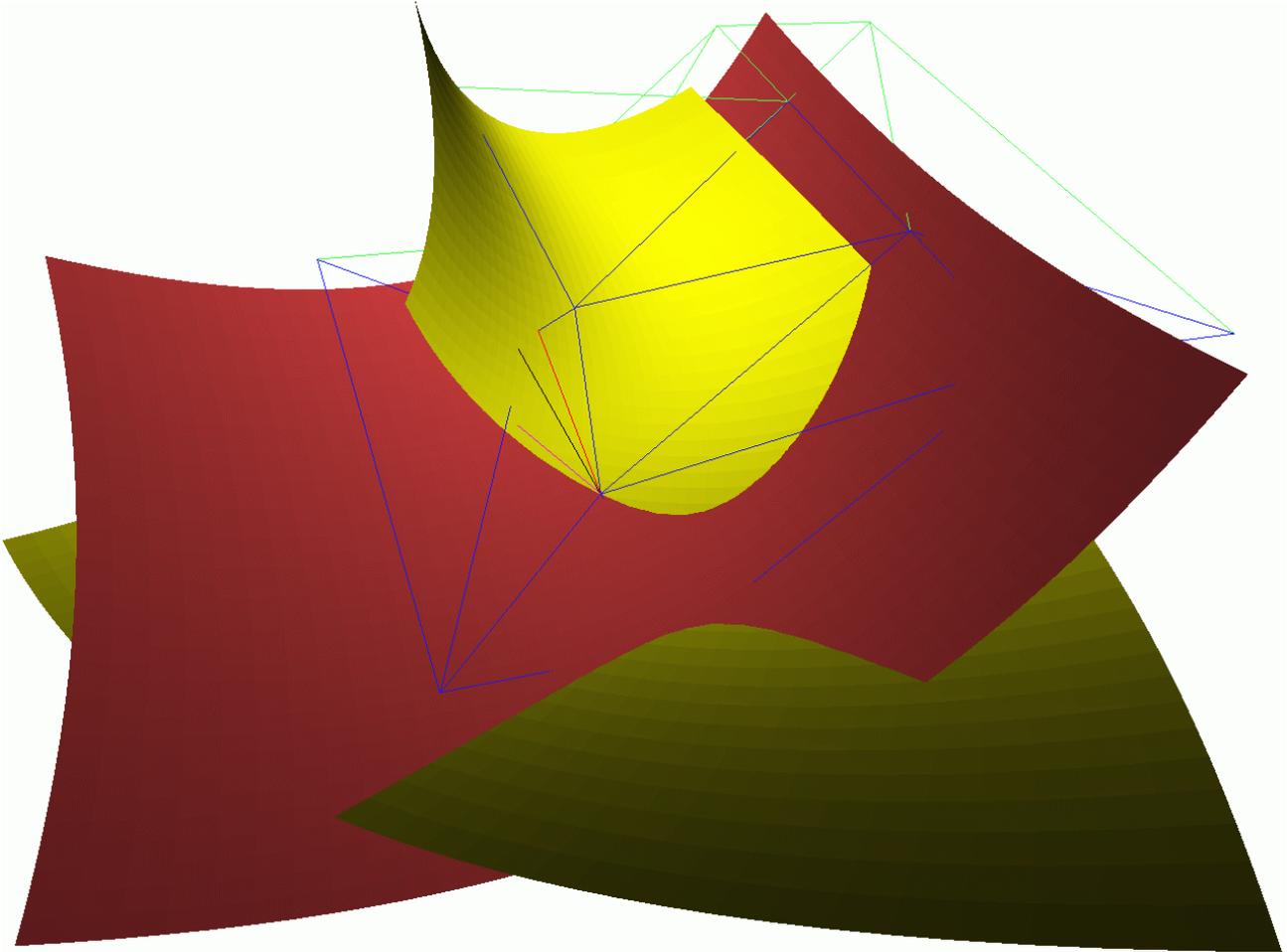


Figure 3.8: Adjusting second order polynomials through one point of the overhang. The blue edges are those which belong to the star shaped neighborhood of the center vertex, the green ones do not. The red paraboloid is the graph of a bivariate function which has as parameter domain an adjusting plane (black normal vector) through the points. The yellow surface is a vector-valued polynomial and describes the overall shape of the surface (i.e. the overhang) better. This figure is also shown in color as fig. E.1 on page 123.

described in Sec. 3.2.4 was used (based on the altitudes for the outer triangles). The normal vector in the center vertex is shown in magenta. The overall shape of the triangulation is followed much better by this surface, though the points along the left and front border are approximated less well. The surface has self-intersections but the shown area is free of those. As it can be seen, not only the normal vector, but also the curvatures of the terrain in the center vertex are estimated in this example much better by the vector-valued surface than by the scalar-valued.

### 3.4 Functionals and variational principle

A functional over a triangulation assigns a real value to the triangular network. Often, these functionals are approximations to corresponding continuous case functionals for parametric surfaces  $\mathbf{f}(u, v)$ . Two examples for such functionals are the *thin plate energy* and the *membrane energy* [Kobbelt et al., 1998b]:

$$E_{TP} = \int \mathbf{f}_{uu}^2 + 2\mathbf{f}_{uv}^2 + \mathbf{f}_{vv}^2 \, dudv \quad (3.11)$$

$$E_M = \int \mathbf{f}_u^2 + \mathbf{f}_v^2 dudv \quad (3.12)$$

where  $\mathbf{f}_u = \partial \mathbf{f} / \partial u$  denotes the first derivative of  $\mathbf{f}$  after  $u$  (etc.). The second functional is smaller for functions with smaller surface area, while the first functional punishes strong bending. In the curve case the natural spline minimizes this functional over the space of joint cubic polynomials [Hoschek and Lasser, 1993]. These functionals depend on the parameterization of the surface, other functionals (e.g. integral over the mean curvature and quadratic approximations for it) not depending on the parameterization are given in [Kolb et al., 1995] and [Greiner, 1994].

Using the notation introduced in sec. 3.2 ( $\lambda_{i,j}$  is 1, if points with indices  $i$  and  $j$  are neighbors, 0 otherwise) a class of functionals can be defined for triangulations:

$$E_T = \sum_{i,j}^n \left( \lambda_{i,j} w_{i,j} \| \mathbf{P}_i - \mathbf{P}_j \|^2 \right) \quad (3.13)$$

Here,  $n$  is the total number of vertices in the triangulation. This functional describes the energy of the triangulation with a spring for each edge with the spring constant  $w_{i,j}$ . Setting the values of the  $w_{i,j}$  allows to approximate certain physical or mathematical properties. Of course, the simplest choice is  $w_{i,j} = 1$  for all edges of the triangulation. Examples for other choices are given in Sec. 3.2.3. Using the so-called ‘Umbrella’-vector a functional of the type of Eq. 3.13 can be defined which approximates (under certain conditions) the integral over the mean curvature [Kobbelt, 1995]. These functionals have in common, that they are quadratic, i.e. they depend in a quadratic way on the values of the co-ordinates of the vertices  $\mathbf{P}_i$  of the triangulation, most often on co-ordinate differences (i.e. edge lengths, e.g. Eq. 3.5). In [Singer, 1995] functionals are utilized, which describe the surface (sum over all triangle areas) and the surface tension. These functionals are, however, not quadratic.

As mentioned before, functionals can be used to describe a property of a triangulation of a surface. But functionals can also be used in order to determine free parameters of a surface. In these cases, the parameters are determined in a way which minimizes the chosen functional. This is applicable for the continuous case as well as for triangulations. This approach is known as ‘surface fairing’ and ‘discrete fairing’ if applied to triangular meshes [Kobbelt, 1998]. For a triangulation the parameters of the surface can be the co-ordinates of the vertices itself. So, if some or all points of a network (a triangulation) are unknown, then the co-ordinates of these points can be chosen, such that a certain functional is minimized (variational principle). Of course, the neighborhood relations have to be known in advance. In the examples of [Singer, 1995] only the boundary curve is known and minimal surfaces are computed. In other examples the surface tension is minimized for a given volume yielding soap bubble surfaces. In [Kobbelt, 1995] new vertices are inserted into an existing triangulation over each edge and their co-ordinates are determined in a way which minimizes the overall curvature by minimizing an appropriate functional.

If the functional is quadratic then it has an absolute minimum (or maximum). Differentiating after the unknown co-ordinates leads to a linear system of equations, which has a unique solution.

### 3.5 Mesh improvement

Very narrow triangles often pose problems to algorithms on triangulations. Inner points with low valency (e.g. three) are often vertices in such narrow triangles, but also points with higher valencies can be vertices in such narrow triangles. Very narrow triangles can occur on the boundary of the triangulation as it can be seen in Fig. 2.7, but also in the interior (Fig. 6.6, upper left). If two nearby points form one edge of such a triangle, then this triangle can be very steep (relative to the ground plan), though the height difference between the two points is only small.

The examples computed within this work show, that these triangles pose problems to the algorithms used. Thus, these triangles should be avoided during data capturing (if possible) or have to be eliminated otherwise. One criterion for the elimination is to replace a pair of points by only one (new) point, if their distance is

below a certain threshold value. Its size should be determined by the measurement process<sup>13</sup> or guided by knowledge on the terrain (e.g. expected curvature, ...). The average of two such points can be taken, or one point can be favored, e.g. if it is a breakline point. A different criterion for enhancing the triangulation is to eliminate the edge of a triangle with small angles. The two points opposing this narrow angle have to be replaced by one point. The advantage of this criterion is, that it adapts locally to the given point density.

Of course, points which are surrounded by many other points in a flat region can also be eliminated. These points do not provide additional information on the shape of the terrain. Different algorithms for the elimination of spurious points have been developed. These possibilities were not used in this work. However, on the subject of mesh improvement much literature is available. In mesh decimation vertices are removed depending on curvature criterions, but also other criterions can be used (e.g. [Kobbelt et al., 1998a]), whereas in the field of mesh optimization the vertices may be moved, and of course also removed, in order to get a simplified mesh which approximates the original one (e.g. [Hoppe et al., 1993]). An approach where even the topology is simplified is described in [Garland and Heckbert, 1997], surveys are given in [Heckbert and Garland, 1997] and [Gotsman et al., 2002].

### 3.6 Filtering of random measurement errors

Each measurement process (i.e. making an observation) works with a certain accuracy, with other words measuring always includes making an error. In general this error is unknown, but often statements on the stochastic properties of the error are possible. If measurements have been performed in a way overdetermining the quantity (or object) to be recorded, improved measurement results can be estimated. This leads to an improved 'version' of the sought quantity (or object). In contrast to Sec. 3.3, not geometric but stochastic properties of the vertices of a triangulation are treated in this chapter. Accordingly, not a geometric but a stochastic model of the underlying surface, but also of the measurement process, is necessary. Of course, the methods of Sec. 3.3 can also be used to estimate some residual vectors in the vertices, but these are not based on a stochastic model and will therefore – in general – not be compliant with the error model of the observations.

Algorithms for the removal of systematic errors or gross errors (blunders) were not considered. Elimination of gross errors for triangulations is possible e.g. with mathematical morphology [Vosselman, 2000]. For filtering of random measurement errors the standard approach of linear prediction [Kraus and Mikhail, 1972] and [Kraus, 2000] (also known as kriging) was adapted to triangulations. As the main characteristics of this approach are known (e.g. the possibility of individual accuracies for each point [Kraus, 1997]), and also because the filtering is not the main task in this work, only an outline of the method will be given.

The estimation of the residual is performed separately for each point. This means, that for each point a linear system of equations has to be solved. It is based on the neighborhood of the point, for which a parameterization has to be computed first. If the points are parameterized by projection onto a plane, only one height function has to be determined, otherwise three, one for each co-ordinate direction. The value of the height function at the parameter location of the point is the filtered 'version' of the point. The difference from the filtered to the observed point is the residual or filter value, its negative value is the correction.

The stochastic properties of the surface (the covariance function) have been chosen in advance. Methods for estimating the covariance function for 2.5D data are given in [Kraus, 2000]. The size of the neighborhood can be chosen according to the covariance function, because this function attenuates and therefore limits the influence of a point with increasing distance from it. The residuals are easily obtained from the kriging system.

If the points are parameterized by orthogonal projection onto a plane (Sec. 3.2.2), then the heights to be interpolated – but also the residuals – are measured in the direction of the normal vector  $\mathbf{n}$  of this plane. (These heights are given in Eq. 3.3.) If the residual at the point  $\mathbf{P}$  of interest is  $r$ , then the residual vector  $\mathbf{r} = r\mathbf{n}$  and the improved position of the point is  $\mathbf{P} - \mathbf{r}$ . If the parameterization plane is an approximation of the tangent plane, this means that filtering is always performed orthogonal to the tangent plane. It is doubtful, if this direction is always the likely direction of the error in the measurement process. However, for smooth surfaces, errors in tangential directions are hard to detect, anyway.

<sup>13</sup>For laser scanning a lower threshold value could be given e.g. by the footprint size of the laser beam. Points with overlapping regions of distance measurement integration are not independent from each other.

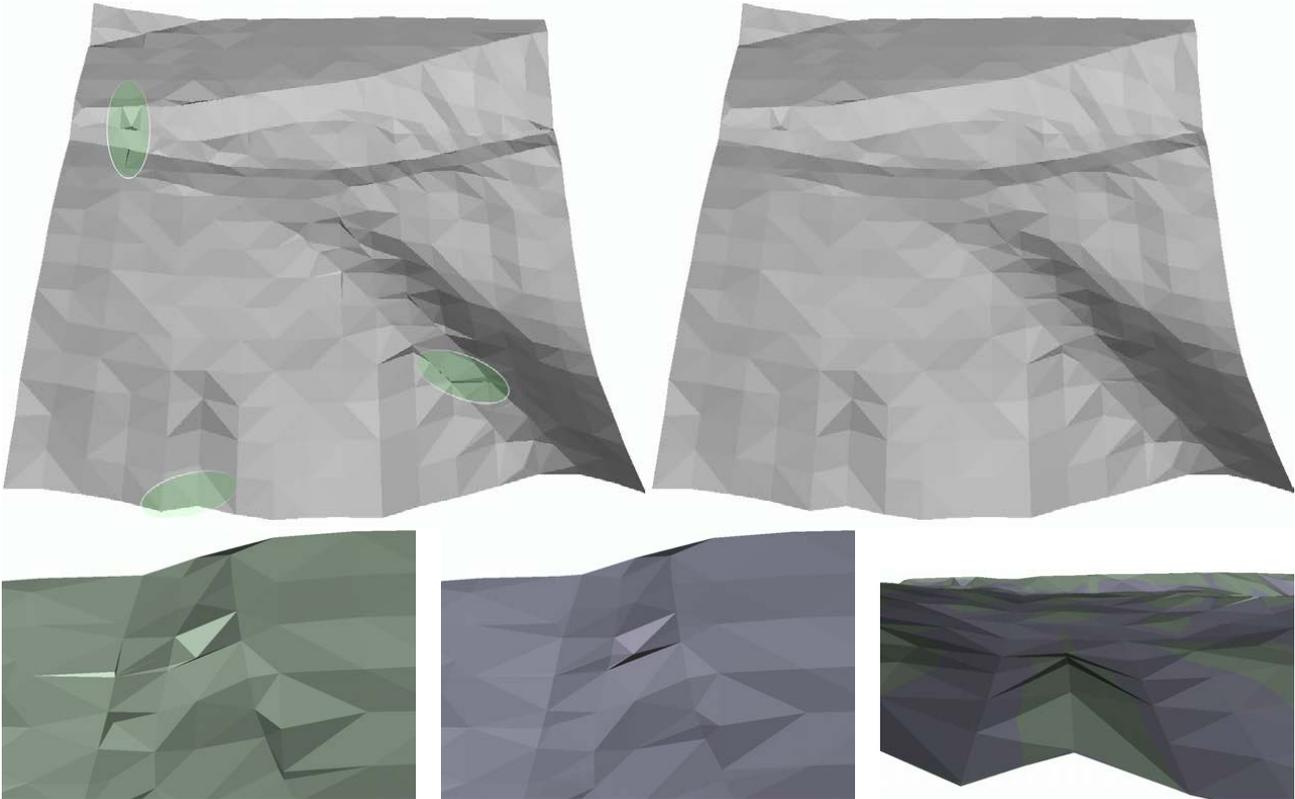


Figure 3.9: Filtering of random measurement errors in a triangulation. Upper left shows the original triangulation, Upper right the filtered version. The first two images in the lower row show the uppermost indicated area viewed from the right hand side. The last image shows an overlay of both triangulations at the lowest indicated area. The green triangles belong to the original triangulation, the blue ones to the filtered one. This figure is also shown in color as fig. E.6 on page 126.

One example of a triangulation before and after filtering of random measurement errors can be seen in Fig. 3.9. The steep thin triangles, which can be found at two of the indicated areas (green ellipses) in the figure originate from two surface points which are almost at the same ground plan position but do have different height observations. In the filtered triangulation the gradient of these triangles conforms to the gradients of the neighboring triangles (lower left images). Of course, a mesh improvement as described in Sec. 3.5 could solve this problem, too, but the stochastic properties should be considered. However, a filtering is always a low-pass operation which can be seen in the lower right part of Fig. 3.9. The peaks and the sinks have been levelled slightly. By the covariance function of the kriging system the levelling can be controlled, but this would also effect the levelling of the steep triangles.

If the points are parameterized by any of the other methods specified above, three surfaces ( $x(u, v)$ ,  $y(u, v)$ ,  $z(u, v)$ ) have to be computed and residuals for each co-ordinate direction are obtained.

In computational geometry other algorithms for the ‘denoising’ (smoothing) of meshes have been developed, e.g. [Desbrun et al., 1999], but these methods do not have the same generality as kriging. On the other hand, these methods are much faster. A different approach is presented in the work of [Mokhtarian et al., 2001] where smoothing is applied iteratively. Each point is smoothed by convolution with a gaussian bell curve. Next, the mesh is decimated to remove superfluous points (Sec. 3.5). These two steps are repeated.

## 3.7 Consideration of breaklines and special points

Breaklines are lines where a discontinuity of the surface tangent planes is allowed and desired. Two neighboring areas, each one smooth, which are separated by a breakline share only this breakline, but the 'flow of information' from one side to the other side is prohibited. One strategy for their consideration is to limit the neighborhood of a point by breaklines as described below.

Peak points and saddle points are other special features of the terrain. The terrain may be smooth except at a peak, therefore a neighborhood limiting property may be considered, too. A different approach is to consider the special geometric properties at such points. At a saddle point the terrain has a horizontal tangent plane. In general this means, that for certain points the tangent plane may be known in advance (Sec. 3.7.2). Other features have been mentioned in Sec. 2.4. These features impose geometric conditions which have to be fulfilled. Finally, in some cases special modelling techniques can be used to describe the topographic features appropriately (Sec. 3.7.3).

### 3.7.1 Neighborhood restrictions

A breakline forms a boundary condition for each of its adjacent regions. If the breakline is for example a cubic spline, then this cubic spline has to be interpolated by the adjacent regions.

For the determination of each region, the breakline can be considered as a border line. A breakline therefore limits the neighborhood of a point. Points on the other side of a breakline must not belong to the neighborhood of a point on the first side. This approach has its limits, if the end of a breakline does not meet another breakline. With other words, this approach is sufficient, if the breaklines together with the boundary of the area partition the complete region into independent subsets and the breaklines are the borders of these regions.

If a breakline ends without meeting another line, than the breakline can be used, nevertheless, to limit the neighborhood of a point. Points which lie behind the breakline do not belong to the neighborhood. For the term 'behind' a parameterization of the unrestricted neighborhood is necessary<sup>14</sup>. Alternatively, those neighbors where the geodesic line in the triangulation ([Opitz and Pottmann, 1994]) crosses the breakline can be considered to be behind it. Another possibility is to forbid the paths used for measuring generational distance in the triangulation to cross the breakline. The last approach was used in this work.

Examples of neighborhood definitions at or near a breakline are given in Fig. 3.10. Left a star shaped neighborhood of a point adjacent to a breakline point is shown. In the middle, one of the two possible generation one neighborhoods of a breakline point is shown. In the right part of the figure the neighbors up to generation two of a point adjacent to a breakline end point are shown. The bright ray starting at the specified point and passing through the breakline end point offers one possibility to discriminate between neighbors and 'invisible' points.

In the case of a breakline ending in the middle of a region the above definition is one solution, but it is not completely satisfactory. There is a sudden step from dependency (neighbored) to independency (invisible). A slower transition between these two stages may be more appropriate, but studies on this subject are not covered here.

### 3.7.2 Prescribed tangent planes

In order to consider geometric boundary conditions it is sometimes sufficient to include the specific requirements in the local modelling of the surface. This can replace or complement the methods of Sec. 3.3 (estimation of geometric properties) and Sec. 3.6 (filtering of measurement errors). If a normal vector has to be estimated in each point of the triangulation and a specific point is known to be a saddle point then the estimation of the normal vector can be neglected for this point because its normal is known to be  $(0 \ 0 \ 1)^T$ . The geometric properties of the model are then conforming to the special terrain feature.

<sup>14</sup>Of course, if the data is only 2.5D the ground plan can be used for this

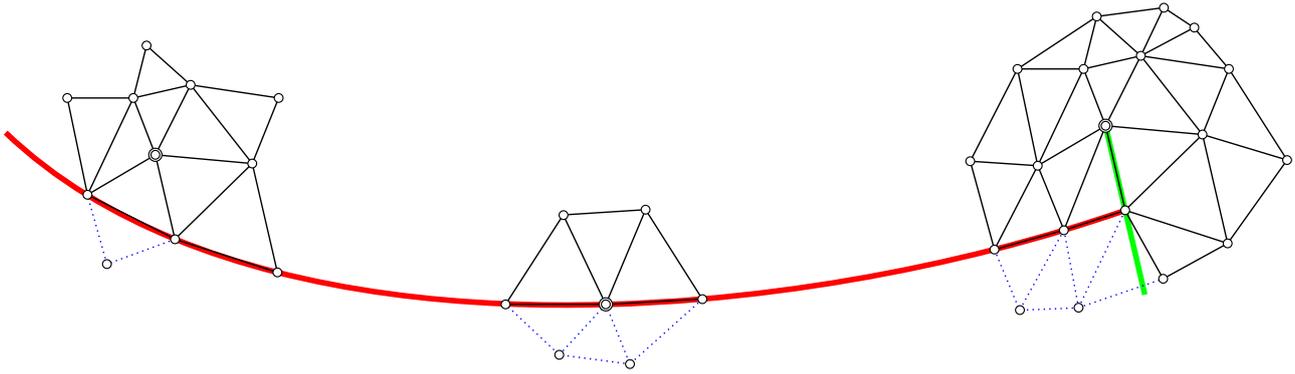


Figure 3.10: Neighborhood of a specified point (two circles) under the consideration of a breakline (continuous line from left to right). The edges (dotted or solid) show the neighborhood under the absence of a breakline. The points which have only dotted edges emanating do not belong to the restricted neighborhood.

The interpolation constraining to a prescribed tangent is possible for many interpolation methods like linear prediction and polynomial interpolation. In [Kraus, 2000] and [Pfeifer, 1997] formulas are given for the interpolation of a prescribed tangent for the multi-quadrics method, but of course these formulae can be used for point kriging, too. Considering two such tangents in different (orthogonal) directions in one point prescribes a tangent plane.

### 3.7.3 Surfaces and lines at special points

Special points like mountain peaks may sometimes be modelled as intersections of breaklines. In this case the methods mentioned above can be applied. However, the existence of breaklines is – at least theoretically – not necessary. Discontinuities can occur at a single point, not only along a line. A simple mathematical model for this is a cone. It is differentiable everywhere except at its vertex. Using such a model (e.g. a cone with a vertical axis and an elliptic cross section) the surface can be estimated locally obeying the prescribed conditions.

Line data is usually only provided in polygon form, though breaklines are differentiable. It is only the surface across the breakline which is not differentiable. Given this ‘raw’ breakline data, the random measurement errors should be filtered first. Such a filtering is performed only for the line itself (i.e. without neighboring surface points). Algorithms for the filtering, also extended to line networks, are given in [Forkert et al., 1995]. In this approach the breakline points are not only filtered, but cubic splines are determined, too. These splines form boundary conditions, which have to be fulfilled by the algorithms which are used to determine the surface afterwards. If properties like surface normal vectors have to be determined for breakline points, then these properties have to be determined two times, once for each side. For this the neighborhood limitation as described above applies.

Another possibility has been sketched in [Kraus and Pfeifer, 2001]. The surrounding area of the breakline segment is modelled as a pair of surfaces (e.g. planes) which intersect in the breakline.

For form lines (definition p. 18) the consideration is more complex, also because the geometric properties are less well defined. If the form line is a terrain gradient line, then the interpolation of this data leads to contour lines of the surface that intersect the form line at right angles. Latter can be achieved by prescribing horizontal tangents. If a form line is defined as a geodesic line, then the curve normal vector<sup>15</sup> field has to be estimated. In this case the curve normal vectors are also surface normal vectors. If the form lines are defined as lines with (almost) parabolic surface points, then second order data has to be prescribed in the form line points<sup>16</sup>.

<sup>15</sup>For a curve the term ‘normal vector’ refers to the normal vector defined by the Frenet frame (also Frenet trihedron) of space curves. This normal vector lies in the osculating plane of the curve.

<sup>16</sup>In Chap. 5 a surface reconstruction technique will be described which requires that in each measured point a local surface – an approximation to the underlying surface – is estimated. In this estimation the zero curvature in a given direction can be considered by choosing an appropriate local surface type, e.g. a general cylinder.

Considering inequalities is, again, more complex, one such example at terrain features are river lines. The height at a point is always lower than the height of the previous point. One method is to estimate the height difference and introduce it as an observation with a certain accuracy. This accuracy has to be chosen carefully to allow the model on one hand to adapt to the surrounding data but to prohibit height increases on the other hand. Of course, the monotony of the height values cannot be guaranteed in this fashion, but it is – at least – enforced.

# Chapter 4

## Parametric patches

In this chapter existing methods and a method already sketched in [Pfeifer and Pottmann, 1996] and also in [Pfeifer, 1997] for the generation of a smooth surface based on a triangulation will be presented. These methods have in common, that for each triangular face one parametric patch, i.e. one local surface, is computed, which interpolates the vertices of the triangulation. These patches are therefore called *triangular interpolating parametric patches*. Some of the methods do not achieve exact smoothness, but continuous differentiability within a certain error bound. This is called approximative smoothness. All presented methods proceed in a similar way in reconstructing the surface. The patches interpolate the vertices of the triangulation, and adjacent patches share the same boundary curve, which guarantees a continuous surface. If adjacent patches have the same field of tangent planes along the mutual boundary curve, then these two patches describe a tangent plane continuous surface (geometric continuity of order one, G1). Generally, a surface is continuous of order  $n$ , if the derivatives up to order  $n$  vary continuously. Two surfaces (or curves) meet with continuity of order  $n$  ( $C^n$ ), if they have parameterizations, such that the derivatives up to order  $n$  are the same along their common boundary. They are *geometrically continuous*, if the corresponding geometric quantities (tangent plane, curvature elements, ...) are the same. This requirement is weaker, because the parameterizations of the two surfaces are not involved. However, if two surfaces meet geometrically continuous of order  $n$ , a parameterization always exists, so that the compound surface is continuous of order  $n$ . Finding this parameterization can be difficult<sup>1</sup>. For the present work, only surfaces with a well defined tangent plane in each point are considered (regular surfaces). Also geometric continuity shall be considered under this point of view<sup>2</sup>.

After a definition of the terms ‘patch’ and ‘patch work’ (a continuous surface made up of single patches), an overview of existing methods will be presented. The last part will be devoted to the developed patch. From its definition it is able to fulfill the requirements stated in Sec. 2.4.

In this chapter the theory on Bézier triangles is assumed to be known. Introductions can be found in the book [Hoschek and Lasser, 1993] and in [Farin, 1986]. In Appx. B the theory on Bézier triangles – which is necessary for Sec. 4.3 – is presented in compact form.

### 4.1 Patches and patch work

A triangular patch is a surface  $\mathbf{f}$  over the triangular domain  $T \in \mathbb{R}^2$ . The location of  $T$  in the plane is not of interest. For the parameterization of  $T$  barycentric co-ordinates are used. The corners of  $T$  are called  $\mathbf{R}$ ,  $\mathbf{S}$  and  $\mathbf{T}$ , and any point in  $\mathbb{R}^2$  can be described uniquely by its barycentric co-ordinates  $r, s, t$  (Fig. 4.1).

$$\mathbf{U} = r\mathbf{R} + s\mathbf{S} + t\mathbf{T}, \quad r + s + t = 1 \quad (4.1)$$

Because the ‘weights’  $r, s, t$  sum up to one, this scheme is *affine invariant*. As all triangles are affinely equivalent,

<sup>1</sup>For a curve this is simpler, because it can always be parameterized by its arc length.

<sup>2</sup>For the univariate case this means, that a curve with a cusp can never be geometric continuous of order one or higher.

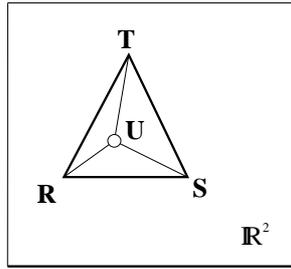


Figure 4.1: Barycentric co-ordinates, for  $U$  the co-ordinates are  $(r, s, t) = (0.458, 0.250, 0.292)$

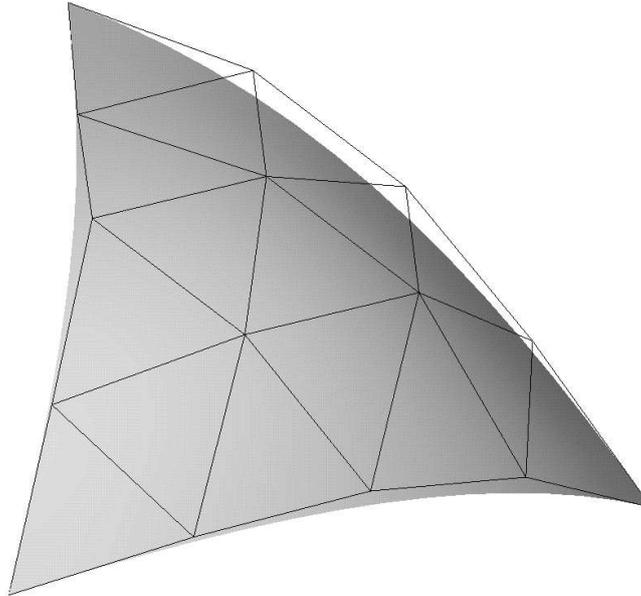


Figure 4.2: A triangular Bézier patch of degree four, here it is shown with its control net. The patch describes a curved surface over a triangular domain.

only the position of  $U$  relative to the triangle corners is of interest. If all three barycentric co-ordinates are greater or greater equal to zero<sup>3</sup>, the point lies in the interior or on the border, respectively, of  $T$ . If one weight is zero, the point lies on the line passing through the other two points. A triangular patch can be defined as a map from  $T$  to  $\mathbb{R}^3$ :

$$\mathbf{f}(r, s, t) = \begin{pmatrix} f_x(r, s, t) \\ f_y(r, s, t) \\ f_z(r, s, t) \end{pmatrix} \quad r + s + t = 1 \text{ and } r, s, t \geq 0 \quad (4.2)$$

If the functions  $f_x, f_y$  and  $f_z$  are bivariate polynomials, then it is advantageous to describe this patch by a Bézier triangle [Farin, 1986]. Fast and numerically stable algorithms for the decomposition of Bézier triangles into triangle meshes exist. Other possible patches are rational Bézier triangles [Hoschek and Lasser, 1993], implicit algebraic patches [Bajaj, 1990] and other patches [Mann et al., 1992]. An example for a Bézier triangle can be seen in Fig. 4.2.

If a triangulation is given, then it is possible to define for each of its triangles one triangular patch. The triangles are the parameter domain for each patch. The surface composed of all patches is called a patch work (also patch net work), it is parameterized over the triangulation. The usual approach to compute such a compound surface is to define the boundary curves first. Then, patches are searched which interpolate the boundary curves and join smoothly – up to a certain order – along these curves. Often, a field of tangent planes or

<sup>3</sup>Together with the affine invariance this yields a *convex combination*.

normal vectors along the boundary curves is prescribed, which have to be interpolated by the patches on either side. Necessary and sufficient conditions for G1 continuity between adjacent Bézier triangles (i.e. bivariate polynomials) are given in [DeRose, 1990] an overview of the patch methods which generate G1 continuity is given in [Mann et al., 1992].

If the boundary curves are defined first, and if twice continuously differentiable patches (i.e. the patches themselves are C2) are searched which have G1 continuity across their borders, this leads to the so-called vertex consistency problem. The G1 continuity conditions set up a circle of constraints around each inner vertex of the triangulation. If only one vertex is considered and Bézier triangles (bivariate polynomials) are used, the following holds: If the number of patches is odd, there is exactly one solution to the vertex consistency problem, if the number is even, one or more solutions exist only under a certain condition.

## 4.2 Method overview

Different methods exist for defining a smooth surface over a triangulation with interpolating parametric triangular patches. There are approaches which provide continuity up to second order (G2), which means that the curvature elements (i.e. principal curvatures and their directions) vary continuously. Other types of patches are defined for general polygons instead of triangles (e.g. the so-called Gregory patch, a survey is given in [Várady, 1987]), but the discussion here will be confined to interpolating triangular patches.

There are many types of patches, an overview and an evaluation of surfaces with these patches is given in [Mann et al., 1992]. As it is mentioned there, the different approaches fulfill the requirement that the surface is G1, but most approaches produce surfaces of low quality and have shape defects. The curvature is spread unevenly over the patch. Many of these patches are not completely determined by the continuity conditions which leaves some parameters to be chosen. Usually the choice of the parameters is based upon some heuristics. However, if these parameters are chosen in a way, that a certain functional (e.g. Eq. 3.11 or Eq. 3.12) is minimized, the surface quality increases, but it leads to a global equation system (e.g. [Kolb, 1995]).

As mentioned above, it is desirable to build a surface composed of Bézier triangles, but the vertex consistency problem has to be solved. There are different possibilities to fulfill the condition arising from the vertex consistency problem or to circumvent the vertex consistency problem.

### Appropriate boundary curves

The net of curves has to be constructed in a way that guarantees that the vertex consistency problem can be solved. A sufficient condition is that the curves meeting in one vertex fit to the same curvature element [Peters, 1991]. It is shown there, that in general a TIN cannot be interpolated smoothly with quartic Bézier triangles. An indication is given, that quintic patches are sufficient. However, quintic patches are of high degree which evokes oscillatory behavior and is therefore not applicable for the present work. By adding a functional for fairing purposes (cf. Sec. 3.4, e.g. thin plate energy, minimum variation of curvature, ...) these undulations can be eliminated.

In [Loop, 1994] one polynomial patch of degree six is computed per triangle, after the construction of boundary curves which fit to a common tangent plane and curvatures in each vertex. In [Hahmann and Bonneau, 2000] (also [Hahmann and Bonneau, 2002]) a method is described where appropriate boundary curves are constructed first and then a split of the domain triangles into four triangles is performed, based on the edge midpoints. Then polynomials of degree five are computed for each patch. For the setting of free parameters the minimization of a functional is used.

## Split domain schemes

A field of tangent planes is prescribed along each boundary curve first. Alternatively the normal vectors can be given or vectors in the tangent plane which are not coincident with the boundary curve tangents. The field must be compatible with the boundary curve itself and in case of polynomial surfaces one order lower than the degree of the patch. Next, each patch is split into three micro-patches. Each of these patches interpolates one boundary curve and its tangent planes. Each micro-patch has two inner boundary curves where it has to meet the other micro-patches with G1 continuity. There are enough degrees of freedom in the choice of the inner boundaries in order to fulfill the continuity conditions. The key idea behind this approach is that the circle of conditions (vertex consistency problem) is broken apart (decoupled) by introducing new boundaries.

Different methods have been proposed. Shirman und Sequin [Shirman and Séquin, 1987] use Farins conditions for G1 continuity [Farin, 1982]. Jensen uses the remaining degrees of freedom to achieve C1 continuity between the micro-patches [Jensen, 1987]. Piper describes an approach, where a patch is constructed for the complete triangle first. This patch is subdivided with the de Casteljau algorithm [Hoschek and Lasser, 1993] into three patches describing parts of the original patch. These patches are modified in order to fulfill G1 requirements and are close to the original patch [Piper, 1987].

The advantage of the split domain approach is that polynomial patches can be used and that these polynomial patches are of low degree (e.g. three).

## Convex combination schemes

These approaches are also called blending methods, because for each face of the triangulation one patch is computed by a blend of basic patches. Also here, cross boundary derivatives (i.e. e.g. the tangent plane fields) are determined first. Next, patches are computed for each triangle which interpolate the boundary data on one side, but not necessarily on the other two sides. Finally, these patches are blended together with appropriate weight functions. These weight functions must give full weight to the basic patches on their corresponding boundary curves and attenuate slowly enough to guarantee also the full weight for the first derivative along the boundary curve. The sum of the weight functions must be one (convex combination). The resulting surface patches are no longer C2, because the derivatives in the corners of the patch are not consistently defined.

If three patches ( $\mathbf{f}_i(r, s, t)$ ,  $i = 1, \dots, 3$ ) are determined over the common parameter domain  $T$ , then the patch  $\mathbf{f}$  can be defined as:

$$\mathbf{f}(r, s, t) = w_1(r, s, t)\mathbf{f}_1 + w_2(r, s, t)\mathbf{f}_2 + w_3(r, s, t)\mathbf{f}_3$$

One possible definition of the weight functions is:

patch	weight function	interpolates tangent plane field
$\mathbf{f}_1$	$w_1 = \frac{st}{rs + st + rt}$	of boundary curve 1 ( <b>ST</b> )
$\mathbf{f}_2$	$w_2 = \frac{rt}{rs + st + rt}$	of boundary curve 2 ( <b>TR</b> )
$\mathbf{f}_3$	$w_3 = \frac{rs}{rs + st + rt}$	of boundary curve 3 ( <b>RS</b> )

It shall be noted that this is only one possible definition of the weight functions, additionally it is possible to blend more than three patches to one common patch.

In [Mann et al., 1992] some blending methods are described. Other methods – also using the weight functions defined above – are given in [Foley and Opitz, 1992] and a curvature continuous interpolant is described in [Hagen and Pottmann, 1989].

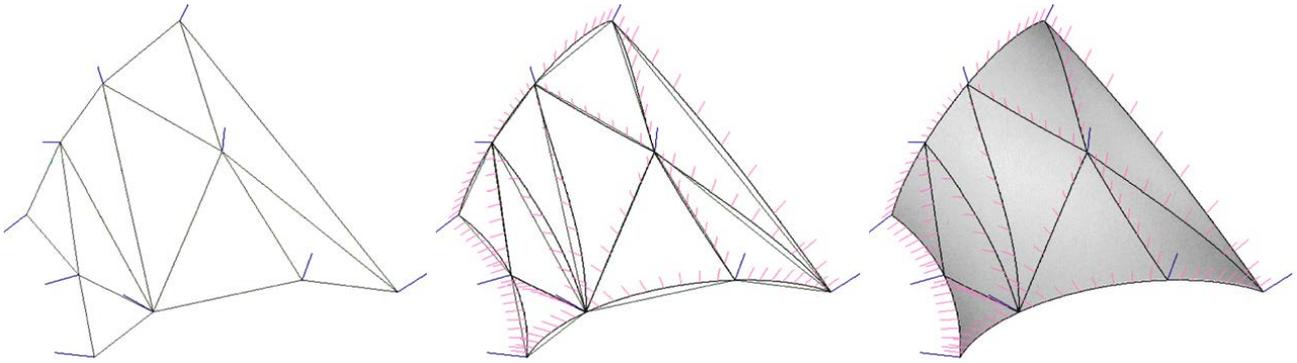


Figure 4.3: Step-wise construction of patches. Left: triangulation and estimated normal vectors in the vertices. Middle: boundary curves and normal vector fields are the patch boundary data. Right: patches for each triangular face.

### Approximate continuity schemes

Approximate continuity has been proposed by [DeRose and Mann, 1992]. So far, three methods have been described to handle the vertex consistency problem: the construction of an appropriate curve network, splitting of patches and blending of patches. The latter two methods often produce patches which are formally G1 but of poor surface quality ([Mann et al., 1992] and [DeRose and Mann, 1992]). One such artefact is that the curvature is concentrated along the boundary curves and the interior of the patches is flat. A different approach is to aim at a better overall shape of the surface and to loosen the continuity requirements on the other hand.

In [DeRose and Mann, 1992] an approach is described which uses in each vertex of the triangulation not only the position and the tangent plane, but also curvature elements. The patches are polynomials of degree three, thus each patch has altogether ten control points. First, planar boundary curves are determined which have the prescribed tangents and curvatures in the end points [de Boor et al., 1987]<sup>4</sup>. Then the remaining degrees of freedom (only one inner Bézier point) are used in order to interpolate the curvatures in the patch corners. This approach is formally G0, but it turns out that the maximum angle between normal vectors of adjacent patches along their common boundary curve is usually small. If this value is larger than a specified value  $\epsilon$  for some boundary curves, then the faces adjacent to these curves are split and the process is repeated for the new patches. Because of the small deviation  $\epsilon$  this scheme of approximate continuity is called an  $\epsilon$ G1 scheme.

### 4.3 An $\epsilon$ G1-continuous polynomial patch

As it has been mentioned above, the survey [Mann et al., 1992] shows that the generation of high quality surfaces is demanding and cannot easily be achieved with the known methods. Applying the convex combination schemes it is much more expensive to compute a surface point and also for split domain schemes the computation takes longer than for one single polynomial patch.

In the following a method to compute one polynomial patch per triangle will be given which achieves approximate continuity. This kind of continuity is called  $\epsilon$ G1. By defining appropriate tangent plane fields the patch is capable of fulfilling the requirements stated in the introduction (Sec. 2.4).

The description will be given step by step. First, normal vectors are determined in the vertices (Fig. 4.3, left), next curves are constructed over each edge of the triangulation. Along these curves a normal vector field is estimated (Fig. 4.3, middle). The final step is to compute patches which interpolate the boundary curves and approximate the normal vector field (Fig. 4.3, right). In flat areas the determination of the patch is instable, thus regularization by minimization of energy functionals is applied additionally. If the approximation of the

<sup>4</sup>There is not always a solution to this system and if a solution exists it is not necessarily unique.

tangent plane field is not sufficiently good, the patch has to be split to achieve the continuity within a certain error bound.

### 4.3.1 Approximate continuity

In this section a definition of approximate continuity [DeRose and Mann, 1992] will be given. It is applied here for Bézier triangles, but the concept itself is much more general.

Two adjacent patches join smoothly if they have the same tangent planes (tangent plane fields) along their common boundary curve (geometric continuity of order one, G1). If the tangent planes of adjacent patches do not match exactly, then the tangent planes of the two patches enclose an angle in each boundary point. If the tangent planes of the patches deviate from each other by a small angle only, then these patches are approximately smooth. Therefore, this kind of approximate continuity of first order is called  $\varepsilon$ G1-continuity, having in mind that  $\varepsilon$  denotes a small positive value. The deviation of the tangent plane fields can be judged in two ways:

1. Comparing two adjacent patches directly: the maximum deviation angle is denoted as  $\gamma$ .
2. Comparing the boundary normal vector field of a patch with a given normal vector field: The maximum deviation angle is denoted as  $\frac{\gamma}{2}$ ; then the first criterion is satisfied as well.

If the angle  $\gamma$  is larger than a defined quality measure  $\varepsilon$ , the approximate continuity was not reached. Choosing the second criterion makes the patches independent from each other. If two adjacent patches are not smooth, but only for one patch  $\frac{\gamma}{2} \leq \varepsilon$  could not be fulfilled, counter measures can be restricted to one patch only.

### 4.3.2 Construction of a curve network

#### Vertex normals

The determination of the normal vectors in the triangulation vertices can be performed with any of the methods described in Sec. 3.3. For the examples given in this chapter the iterative approach was used. The tangent plane is estimated and the points are parameterized by projection onto this plane. The neighbors of the first generation have been used.

#### Boundary curves

The boundary curves are the curves between adjacent patches. They are polynomials and have degree three. This choice is widely used in CAD programs, photogrammetry [Kraus, 2000] and therefore guarantees the compatibility with different algorithms and programs (for example [Forkert et al., 1995] for the adjustment of line networks).

For each edge the boundary curve is determined independently. Therefore it is sufficient to describe only the determination of one curve. Given are two points  $\mathbf{P}_0$  and  $\mathbf{P}_1$  and the surface normal vectors  $\mathbf{n}_0$  and  $\mathbf{n}_1$ . The curve  $\mathbf{c}(t), t \in [0, 1]$  must fulfill the following conditions:

$$\mathbf{c}(0) = \mathbf{P}_0 \tag{4.3}$$

$$\mathbf{c}(1) = \mathbf{P}_1 \tag{4.4}$$

$$\mathbf{n}_0^\top \dot{\mathbf{c}}(0) = 0 \tag{4.5}$$

$$\mathbf{n}_1^\top \dot{\mathbf{c}}(1) = 0 \tag{4.6}$$

The choice of the parameter interval is arbitrary, but of no importance. It is convenient, because Bézier curves are used for the description of the boundaries. The equations state, that the curve must interpolate the given

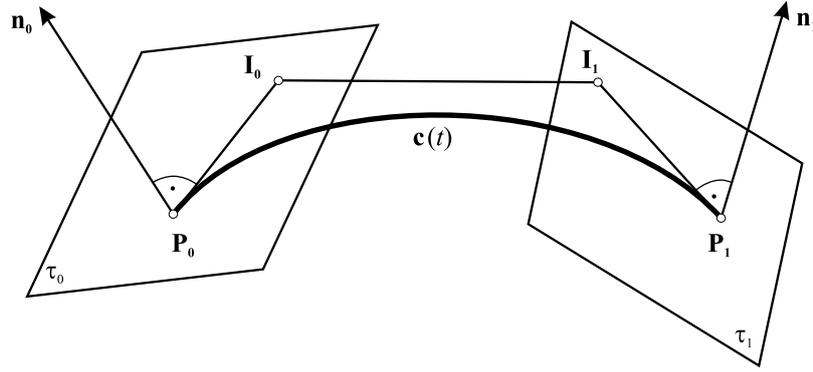


Figure 4.4: Boundary curve as Bézier curve with control polygon and end point interpolation conditions (point and tangent planes).

end points (Eqs. 4.3, 4.4) and the tangent vectors in the end points are orthogonal to the given normal vectors (Eqs. 4.5, 4.6).

The boundary curve  $c(t)$  can be written as a Bézier curve of degree 3 with its 4 control points. Because of the end point interpolating property of Bézier curves, the first and the last control point are  $P_0$  and  $P_1$  (Fig. 4.4). The two inner control points  $I_0$  and  $I_1$  can be used to give Eqs. 4.5 and 4.6 a new form, because the tangents in the end points of Bézier curves have the directions of the vectors from the outer to the adjacent inner control point.

$$\begin{aligned} (I_0 - P_0)^\top n_0 &= 0 \\ (I_1 - P_1)^\top n_1 &= 0 \end{aligned}$$

To summarize, the control polygon of the curve is the point sequence  $P_0, I_0, I_1, P_1$  (Fig. 4.4).

This leaves open two parameters for each inner control point, which can be considered as parameters of the curve tangents in the end points: its direction in the tangent plane and its length. For a Bézier curve of degree  $n$  the tangent to  $t = 0$  is  $n(I_0 - P_0)$ . Analogy applies for the other end point at  $t = 1$ .

**Tangent direction** The tangent direction is chosen as the orthogonal projection of the edge connecting  $P_0$  and  $P_1$  on the tangent plane (normal section). This guarantees, that the sequence of curves around one vertex is the same as the sequence of edges emanating from this vertex. In an orthogonal projection on the tangent plane the edges and the curve end point tangents have the same image.

A different possibility is to choose planar curves, but this does not guarantee, that the curves have the same sequence as the edges around one vertex as it is shown in [Kolb, 1995]. Furthermore, tests in [Pfeifer, 1997] have shown, that the results for the described patch are better, if the tangent direction is the projection of the edge.

Choosing the tangent direction defines two lines, each of which is incident with one of the inner control points. Of course, these lines lie in the tangent planes.

Problems arise, if the edge is orthogonal to the tangent plane. In this case the projection is a point and a tangent direction cannot be defined properly. This is, however, either a problem of the tangent plane estimation which failed, or a problem of the triangulation which samples the underlying surface to sparsely. In the studied examples this problem did not occur.

**Tangent length** As mentioned above, the first derivative of a Bézier curve in the start point is  $n(I_0 - P_0)$ , where  $n$  is the degree of the curve (in this case 3), and  $P_0$  and  $I_0$  are the first and the second control point, respectively. Analogous definitions hold true for the other end point. Choosing the length of the tangent in

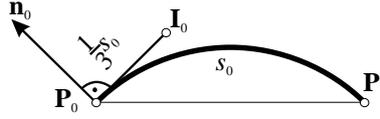


Figure 4.5: Definition of the tangent length in the left end point with a circular segment. The curve has degree 3.

both curve end points therefore defines the curve completely, if the end points themselves and the tangent directions in the end points have already been chosen.

If the tangent lengths are finally defined for a cubic Bézier curve, the inner control points of the curve must be chosen on the correct side of the projection from the edge onto the tangent plane. As mentioned above, in the studied examples this was not a problem. A different kind of problem can occur much easier. If the tangent lengths are too large, the curve can have a cusp or, if it is a planar curve, a loop. The curves used here are close to planar, and it can occur, that the projection of the curve onto an averaging plane has a loop.

Different methods for the choice of the tangent length have been proposed in the literature for this task. The easiest choice is  $\|\dot{\mathbf{c}}(0)\| = \|\dot{\mathbf{c}}(1)\| = \|\mathbf{P}_1 - \mathbf{P}_0\|$ . It leads to curves with strongly varying curvature. The method chosen here, was to set the tangent length in each of the end points equal to the length of the circle segment specified by the chord  $(\mathbf{P}_1 - \mathbf{P}_0)$  and the corresponding tangent direction in either end point (Fig. 4.5 shows this for the first end point,  $s_0$  is the length of the arc). A third possibility is to choose the tangent length in order to minimize the curvature. In this case not the curvature itself, but the integral over the square of the second derivative is minimized (compare Eq. 3.11 for the curve case).

Other methods are to interpolate not only the end points and tangent directions, but also the curvature in the end points. This is possible for planar curves [de Boor et al., 1987], but of course, the normal curvature in the tangent direction would have to be estimated first. If additional points were given on the surface, the tangent length could be estimated in order to achieve an approximation of these points by the curve [Hoschek, 1988].

The choice based upon the lengths of the circle segments provided the best results on test data and was therefore chosen for the following. If there is a right angle between the edge and the normal vector in one end point, the circle degenerates to a straight line. In this case the length of the edge is used as the length of the first derivative in the corresponding end point.

## Normal vector field

Ways to determine a net of curves have been described in the previous sections. This net is consistent in the vertices of the triangulation in the sense, that the tangents in the end points meeting in one vertex fit to a common tangent plane. Many approaches, split-domain as well as convex-combination, proceed by interpolating these curves and finding suitable patches which guarantee the G1 continuity across the patch borders. Patches in other approaches like the transfinite interpolant in [Nielson, 1987] are also capable of interpolating a field of normal vectors or cross boundary derivatives. This approach is taken here, too, though the interpolation constraint is loosened to an approximation task (see below). At this point of the surface reconstruction, a normal vector field along each boundary curve has to be estimated.

The method to determine a field of normal vectors  $\mathbf{n}(t)$  along the boundary curve  $\mathbf{c}(t)$  which is chosen in this work proceeds by blending the normal vectors in the end points. Given are the curve  $\mathbf{c}(t)$ , of course the curve end points  $\mathbf{P}_0$  and  $\mathbf{P}_1$  and the normal vectors in the end points  $\mathbf{n}_0$  and  $\mathbf{n}_1$ .

$$\begin{aligned} \mathbf{n}(t) &= w_0(t)\mathbf{n}_0 + w_1(t)\mathbf{n}_1 & (4.7) \\ w_0(0) &= 1, \quad w_0(1) = 0 \\ w_1(0) &= 0, \quad w_1(1) = 1 \end{aligned}$$

The functions  $w_0$  and  $w_1$  are defined over the interval  $[0,1]$ . The weight functions must be monotonous decreasing and increasing, respectively, which is fulfilled by the functions  $w_0(t) = (1 - t)^k$  and  $w_1(t) = t^k$ , with

$k = 1, 2, \dots$ . This approach does not guarantee, that  $\dot{\mathbf{c}}(t)^\top \mathbf{n}(t) = 0$ , thus the preliminary  $\mathbf{n}(t)$  is projected orthogonally into the plane perpendicular to  $\dot{\mathbf{c}}(t)$  to obtain the final field of normal vectors. In the following  $k = 2$  was chosen, which provokes, that  $\mathbf{n}(t)$  touches the constant functions  $\mathbf{n}_0$  and  $\mathbf{n}_1$  with order one at  $t = 0$  and  $t = 1$ , respectively.

A different approach would be to determine an interpolating function which interpolates the curve end points, the normal vectors in the end points, a fixed number of curve points and the tangents in these curve points. This method would guarantee, that the normal vectors, which are obtained by differentiating the interpolating surface, are perpendicular to the tangents right away. Of course, this holds only at the curve points which are interpolated. Another approach is to estimate cross boundary derivatives along the boundary curve and to construct the normal vector as cross product of the boundary tangent and the cross boundary derivative vector in each curve point. A solution similar to the estimation of the normal vector field is possible. In the end points cross boundary derivatives are defined. Along the curve these vectors are blended into each other. These vectors must be orthogonal to the corresponding vertex normal and not collinear with the tangents in the end point. Setting the cross boundary derivatives orthogonal to these two vectors allows a determination of the patches independently from each other. An advantage of this method is, that it is at least possible to interpolate the complete normal vector field, because the degree of the polynomial describing the cross boundary derivative is sufficiently low.

### 4.3.3 Insertion of patches

To maintain locality, the determination of the inner control points of adjacent patches should be independent from each other, and based on the boundary data alone (curves and normal vector fields). Continuous normal vector fields can in general not be interpolated with Bézier triangles, but the field can be interpolated or approximated at a number of positions. This leads to adjacent patches which have similar, but not exactly the same normal vector fields along their common boundary curve. With a threshold value it is judged, whether the continuity has been achieved satisfyingly, or not. What needs to be done if the criterion is not satisfied is described below.

#### Inner Bézier points and tangent planes along the boundary

For the construction of the patches the boundary curves have already been fixed. Additionally, a field of normal vectors along these curves is prescribed which shall be approximated. The free parameters of the patches are the inner control points. It is now necessary to derive the connection between the inner control points and the field of tangent planes along the boundary curves of a Bézier triangle.

It follows immediately from the tangent plane construction with the algorithm of de Casteljau that the tangent planes of a Bézier triangle along a boundary curve depend only on the corresponding boundary control points and on the points of the neighboring row. In the case of Bézier triangles of degree three, there is only one inner point, the neighboring row has three control points, but two of these are control points for the other boundary curves. All the tangent planes along the boundary are dependent on the choice of this inner point. In the case of degree four, always a pair of two inner control points have influence on the tangent plane along the boundary curve.

The functional dependency between the tangent plane at  $\mathbf{P}$ , a point on the boundary, and the inner points is easily derived (see also Appx. B). Here it will be demonstrated for degree three. Let  $\mathbf{P}$  be a point of the Bézier curve whose control points are  $\mathbf{P}_{300}, \mathbf{P}_{210}, \mathbf{P}_{120}, \mathbf{P}_{030}$ ; its curve parameter shall be  $t$ . Further, let  $\mathbf{Q}$  be the point with parameter  $t$  on the Bézier curve with control points  $\mathbf{P}_{201}, \mathbf{P}_{111}, \mathbf{P}_{021}$ . Then  $\mathbf{q} = \mathbf{Q} - \mathbf{P}$  is a tangent vector at  $\mathbf{P}$  in a direction transversal to the boundary curve. With  $\mathbf{n}$  as normal vector at  $\mathbf{P}$ , we therefore have

$$\mathbf{n}^\top \mathbf{q} = 0. \quad (4.8)$$

In terms of the unknown inner point  $\mathbf{P}_{111}$  and using the Bernstein polynomials for Bézier curves this can be written as

$$\begin{aligned} 0 &= \mathbf{n}^\top \mathbf{q} = \mathbf{n}^\top (\mathbf{Q} - \mathbf{P}) \\ 0 &= \mathbf{n}^\top ((1-t)^2 \mathbf{P}_{201} + 2t(1-t) \mathbf{P}_{111} + t^2 \mathbf{P}_{021} - \mathbf{P}), \\ 2t(1-t) \mathbf{n}^\top \mathbf{P}_{111} &= \mathbf{n}^\top (\mathbf{P} - (1-t)^2 \mathbf{P}_{201} - t^2 \mathbf{P}_{021}). \end{aligned} \quad (4.9)$$

Eq. 4.9 is a scalar product and therefore each boundary normal to be interpolated introduces one linear scalar equation for  $\mathbf{P}_{111}$ . Hence, with a cubic Bézier triangle three normal vectors can be interpolated. These normal vectors can be given on one boundary curve or distributed over the three boundaries. For the two other boundary curves similar formulae can be derived. Of course, the boundary point  $\mathbf{P}$  and the normal vector  $\mathbf{n}$  depend on the parameter  $t$ .

The way to extend Eq. 4.9 for Bézier triangles of degree four is obvious. A Bézier triangle of degree four has three inner points and thus nine normals at the edges can be interpolated. For completeness, the formula for a patches of degree four will be given. If the curve network as described in the previous section has curves of degree three, then the boundary curves have to be degree elevated to order four first. This is only a formal degree elevation, the curve obtains one more control point, but the shape of the curve remains unchanged. For the boundary curve between the control points  $\mathbf{P}_{400}$  and  $\mathbf{P}_{040}$  Eq. 4.9 takes the form:

$$3(1-t)^2 t \mathbf{n}^\top \mathbf{P}_{211} + 3(1-t)t^2 \mathbf{n}^\top \mathbf{P}_{121} = \mathbf{n}^\top (\mathbf{P} - (1-t)^3 \mathbf{P}_{301} - t^3 \mathbf{P}_{031})$$

### Approximation of normal vector fields

Instead of interpolating a number of given surface normals along the boundary, three for a patch of degree three, nine for a patch of degree four, it is also possible to approximate given normal vectors along the boundary with Eq. 4.9. If  $n$  normal vectors  $\mathbf{n}_i$  are given, distributed along the three boundary curves, and  $n$  vectors  $\mathbf{q}_i$  as defined above, the inner points are determined by minimizing:

$$F = \sum_{i=1}^n (\mathbf{q}_i^\top \mathbf{n}_i)^2 \quad (4.10)$$

This system is linear and no approximate values of the inner control points are required. It leads to a linear system of equations of the form:

$$\mathbf{A} \mathbf{x} = \mathbf{l}$$

In the matrix  $\mathbf{A}$ , which has  $n$  rows, the given normal vectors, multiplied with Bernstein polynomials, are contained. The vector  $\mathbf{x}$  contains the co-ordinates of the unknown inner point(s), it has three rows for a patch of degree three and nine rows for degree four. The vector  $\mathbf{l}$  contains the known control points of the boundary polygons multiplied with Bernstein polynomials. The minimization leads to the well-known system:

$$\mathbf{A}^\top \mathbf{A} \mathbf{x} = \mathbf{A}^\top \mathbf{l} \quad (4.11)$$

The number of prescribed normals per side for patches of degree four should be six. This result has been obtained from experiments on patch work over a torus. A torus is suitable, because it has hyperbolic, parabolic and elliptic regions. For fewer normals, the smoothness decreases notably, a higher number of normals does not increase the smoothness.

If the vectors  $\mathbf{n}_i$  are unit vectors and if all the vectors  $\mathbf{q}_i$  have approximately the same length, and if the angles between the  $\mathbf{n}_i$  and  $\mathbf{q}_i$  are close to right angles, then not only squares of scalar products but approximately the sum of squares of angles is minimized by solving 4.11. Otherwise, the angles are weighted with the lengths of  $\mathbf{q}_i$ , which are, of course, unknown at this time. A weight iteration could be used to minimize the sum of squared angles, but this has not been tried.

If  $\mathbf{A}$  has not full rank, the system is not solvable. For a patch of degree 3 this means, that the prescribed normal vectors must span the 3D space. In flat areas, the system becomes very instable, the normal vectors

show approximately in the same direction. It can occur that Bèzier triangles ‘tumble over’ and approximate the surface normal field from the wrong side. This can be seen in Fig. 4.6 in the upper part. The figure shows a small patch work of three patches with Bèzier triangles of degree four and their control nets. These patches have been determined in the way described in this chapter so far. At each boundary curve six normal vectors have been prescribed, distributed evenly in the parameter domain. At the inner control points the error ellipsoids from their determination are drawn (in green). In some inner points the ellipsoids are very narrow

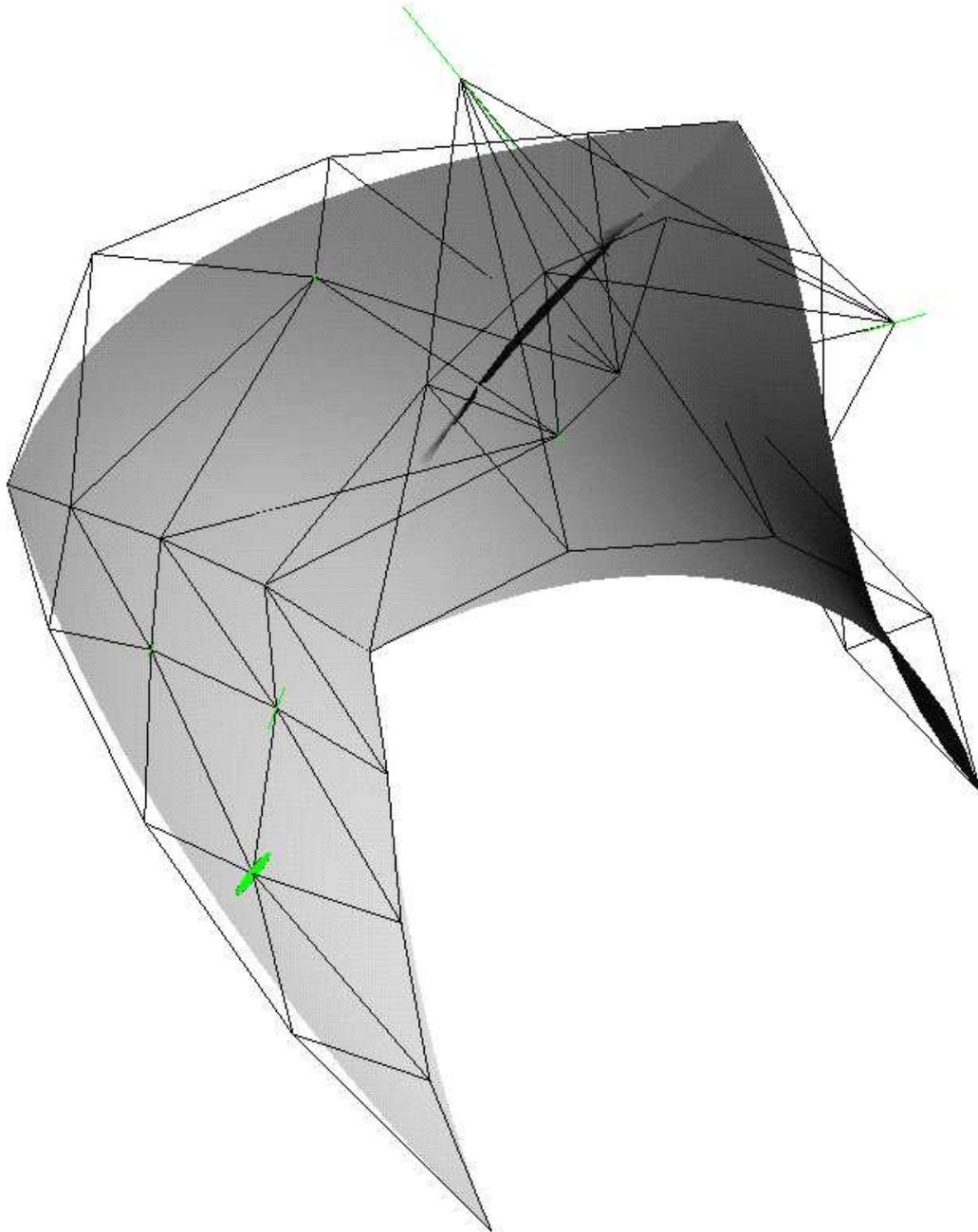


Figure 4.6: Patch work with three patches. The error ellipsoids are scaled by a factor  $10^{-3}$  (i.e. made smaller). At some inner points the error ellipsoids are too small in order to be seen. Along the left common boundary the continuity has been achieved quite well. At the right boundary curve the middle Bèzier patch tumbles over and approximates the prescribed tangent planes from the wrong side.

and therefore appear as ‘line segments’. All ellipsoids are oriented towards the boundary curves. They have been taken from the normal equations inverse  $\mathbf{Q}$  scaled with the unit weight error  $m_0$ .

$$\begin{aligned}\mathbf{Q} &= (\mathbf{A}^\top \mathbf{A})^{-1} \\ m_0 &= \sqrt{\frac{\mathbf{r}^\top \mathbf{r}}{n-u}} \\ \mathbf{r} &= \mathbf{A}\mathbf{x} - \mathbf{1} \quad \text{correction vector} \\ u &\dots \quad \text{number of unknowns, 3 or 9}\end{aligned}$$

#### 4.3.4 Insertion of patches and minimizing energy

The problems described in the previous section are obvious. The inner points are only determined weakly, if the prescribed normal vectors are too similar. The positions of the inner control points are determined strongly in the direction of the normals, but only weakly in the perpendicular direction. The system of equations becomes singular or at least very instable if the normals are too similar. The consequence is, that the inner points lie not within the ‘area’ circumscribed by the outer (boundary) control points. A method has to be found which guarantees, or at least enforces, this property. This can be achieved by minimizing not only Eq. 4.10 but by also including an energy functional for regularization purposes.

$$F_E = \alpha \left( \sum_{i=1}^n (\mathbf{n}_i^\top \mathbf{q}_i)^2 \right) + (1 - \alpha)E, \quad \text{with } \alpha \in [0, 1]. \quad (4.12)$$

The functional  $E$  can be the thin plate energy (Eq. 3.11) or the membrane energy (Eq. 3.12). The weight factor  $\alpha$  determines which weight shall be given to the minimization of the energy versus the minimization of the angles between the prescribed and the obtained normals.

Instead of applying functionals to continuous surfaces, they can also be applied to meshes, as it has been described in Sec. 3.4. In this case the mesh can be the control net, but also a subdivided version of the control net (see Fig. B.4). The control points of the subdivided nets depend only linear on the control points<sup>5</sup>. If  $E$  is a quadratic functional depending on the unknown inner control points (vector  $\mathbf{x}$ ) it can be written in the form

$$E(\mathbf{x}) = \mathbf{x}^\top \mathbf{E}_2 \mathbf{x} + \mathbf{e}_1^\top \mathbf{x} + e_0 \quad (4.13)$$

Inserting 4.13 in 4.12 and applying the minimization approach leads to

$$(\alpha \mathbf{A}^\top \mathbf{A} + (1 - \alpha) \mathbf{E}_2) \mathbf{x} = \alpha \mathbf{A}^\top \mathbf{1} - \frac{1 - \alpha}{2} \mathbf{e}_1 \quad (4.14)$$

which is still a linear system of equations with a unique minimum.

The choice of  $\alpha$  determines the regularization power. In this work the energy for meshes (Eq. 3.13) with equal weight for all edges ( $w_{ij} \equiv 1$ ) has been chosen. Experiments on the torus (see above) have shown that the number of subdivision steps before the application of the regularization has little influence. In Fig. 4.7 the same data as in Fig. 4.6 is shown with regularization (thin plate energy) after one subdivision step. As in the previous example the surface at the left boundary between the adjacent patches appears continuous. At the right common boundary curve the regularization shows its effects, the surface does not tumble over anymore. In the upper part of the boundary curve a lack of smoothness of the surface can be observed. Note, that this lack of smoothness has also been visible in the previous version, where the surface tumbled over additionally.

Another example can be seen in Fig. 4.8, where different regularization factors have been used for one example. Again, the thin plate energy was applied, but here it was applied directly to the control net. It shows a vertical wall with two adjacent horizontal areas. Eight normal vectors were prescribed per edge for the determination

<sup>5</sup>Of course, the locations of the subdivided control net points depend in a cubic or quartic way on the locations of the subdivision points in the parameter domain. These locations are fixed and the coefficients are therefore constants.

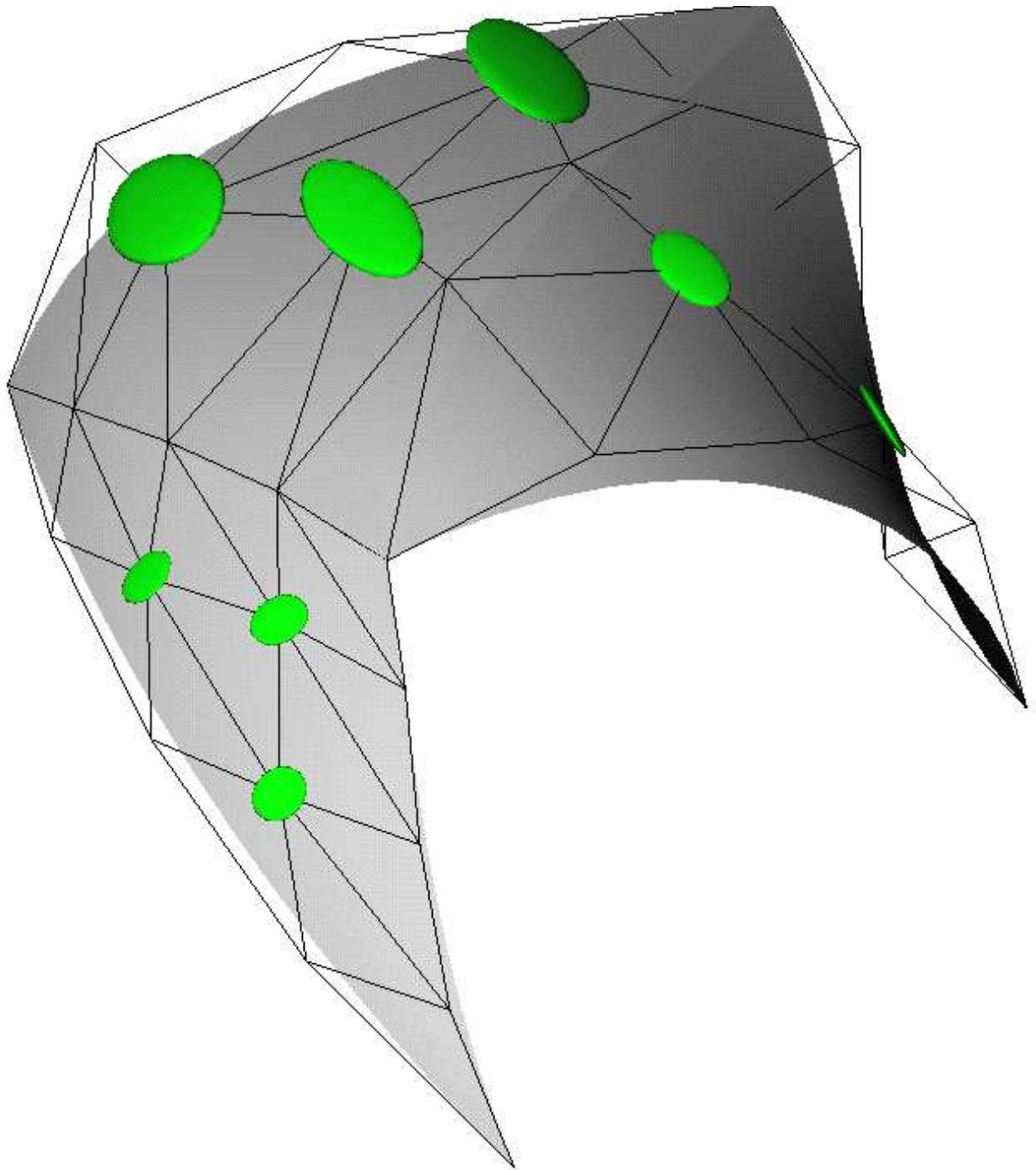


Figure 4.7: Patch work with three patches. The error ellipsoids are not scaled. For the determination regularization with  $\alpha = 0.95$  was applied. The control net was divided into four sub-nets (one subdivision step based on subdivision at the edge midpoints).

of the inner Beziér points. In the upper left image the triangulation is shown. From left to right and top to bottom the values for  $\alpha$  (Eq. 4.14) are: 1.0, 0.999, 0.95, 0.8, and 0.001. Not all control nets are drawn in order to give a better impression of the surface. Additionally, the given data is symmetric, so most of the missing control nets are symmetric to the drawn ones. In the upper right surface, the highest smoothness is obtained (no regularization), but again, the surface tumbles over along the upper edge. On the lower edge the smoothness is achieved as desired. Along the diagonal edges from the inner point with valency seven to the upper outer points the patches approximate the prescribed normal vector field, from the same side too, instead of joining from different sides. These effects are removed with a weak regularization of 0.1% ( $\alpha = 0.999$ ). However, most of the curvature can still be found close to the edges of the Beziér patches. With stronger regularization the smoothness is clearly diminished.

### 4.3.5 Additional splitting

If the smoothness is not achieved up to a user specified threshold value, the faces have to be split. This gives more degrees of freedom in order to allow a better approximation of the normal vector field. The advantage versus split domain schemes is, that splitting occurs only where it is necessary. One method is to introduce new points at all those edges (curves) where the criterion could not be fulfilled. This provokes a re-triangulation of the vertices, a new determination of normal vectors, curves and normal vector fields. The new edges only divide the existing triangles and the calculation only concerns a certain neighborhood of the edge.

A different approach is to apply one of the split domain schemes directly. In this case all the normal vector fields should be determined according to one of the construction principles of the split domain schemes [Mann et al., 1992]. The continuity could then be determined exactly at those edges. The free parameters of the split domain schemes could be set so that the previous patch is approximated.

Additional ideas and algorithms concerning the splitting are given in [Pfeifer, 1997].

### 4.3.6 Results

In this chapter a patch work approach with Beziér patches of degree four was proposed. Over the edges of the triangulation curves of polynomial degree three are estimated and interpolated by the patches. Approximate continuity is obtained by prescribing a field of normal vectors along those boundary curves which is approximated by the patches from either side. In the determination of the boundary curves and their normal vector fields geometric boundary conditions (e.g. given normal vectors, ...) can be considered. The described method produces surfaces which are, up to a small angle, smooth ( $\varepsilon G1$ ). In certain situations the smoothness cannot be achieved to a satisfying degree. In these cases a splitting of the patches is inevitable. Under certain conditions the patches approximate the normal vector field but the patches tumble over and reach the boundary curve from the wrong side. The proposed regularization technique (minimizing a functional of the control net, namely the thin plate energy) successfully prohibits the tumbling over of patches. However, the curvature tends to be concentrated along the boundaries. This suggests that other functionals (e.g. minimum variation of curvature (MVC), [Greiner, 1994]) may be more appropriate.

Breaklines which are given in the triangulation can be considered as a sequence of edges. Along these edges an interpolating cubic spline curve is determined. This line is already the final breakline of the patch work surface. In each breakline vertex two surface normal vectors must be estimated, one for either side of the line. This estimation is only performed with the points on the line itself and those in the appropriate region of the surface. Also two normal vector fields are determined, one for either side. The patches adjacent to the break line edge approximate the corresponding normal vector field. This guarantees, that the surface is continuous but not differentiable along the break line.

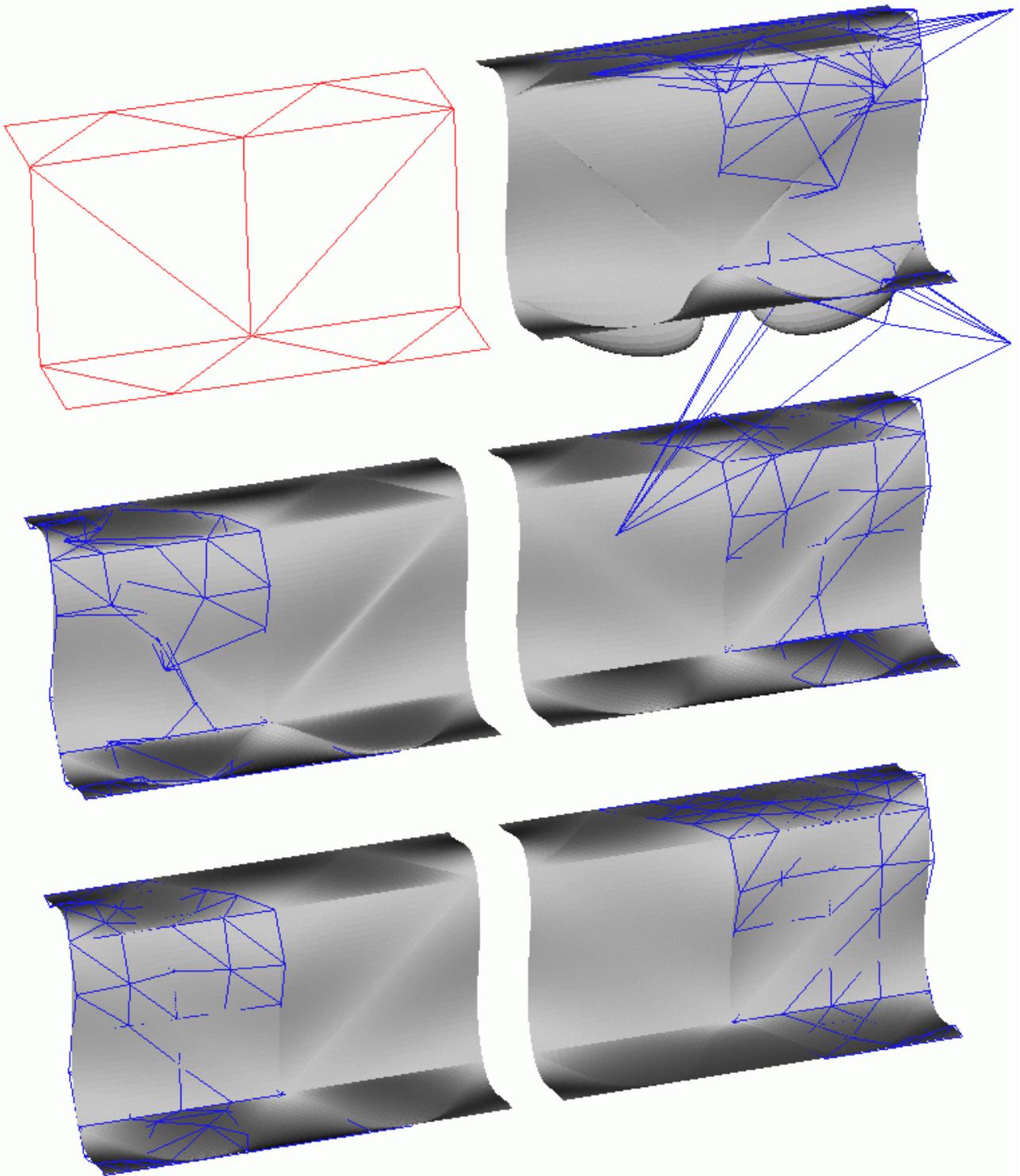


Figure 4.8: Different regularization factors for the determination of a patch work over a terrain step. In the upper left the given triangulation is shown. The value  $\alpha$  for the different surfaces is: 1.0, 0.999, 0.95, 0.8, and 0.001. In blue some of the control nets are shown.



# Chapter 5

## Subdivision

This chapter introduces the subdivision technique for the generation of smooth surfaces based on triangulations. Compared to the patch work approach with parametric patches (Chap. 4) this technique is younger. First a short introduction to subdivision will be given, followed by an overview of existing approaches which are – more or less – suitable for the modelling of topographic surfaces. In Sec. 5.3 a new technique will be presented which is capable of fulfilling geometric boundary conditions.

### 5.1 The subdivision paradigm

The idea behind subdivision is to compute a surface iteratively by refining a surface mesh. In subsequent steps more and more points and faces are generated yielding a mesh with growing density. The beginning is made with a mesh, which is for the given task a triangulation, but the mesh can also be composed of quadrilaterals or of arbitrary shaped cells. In each step new points are computed. For triangular meshes this is performed by the so-called *face split* where each face is split into four new faces. The new points are inserted topologically into (over) the existing edges. Alternatively, in the so-called  $\sqrt{3}$ -refinement, one point is inserted per patch. Each new point is connected to the vertices of its surrounding triangle and the edges which connect vertices of the previous step are swapped (e.g. [Ivrissimtzis and Seidel, 2002], where other refinement operators are described, too). For quadrilateral meshes also a *vertex split* is possible, where one vertex is replaced by four new vertices. The face split methods are also called primal subdivision schemes, whereas vertex split subdivision is also known as dual subdivision. An example of triangular subdivision can be seen in Fig. 5.1.

The mesh which describes the surfaces is refined in steps. This yields a surface representation at different (subdivision) levels, level zero corresponds to the original given triangulation. If the rules for the *refinement* are the same in each level, the scheme is called *stationary*. In the face splitting schemes the vertices of one level are grouped into *even* and *odd* vertices. Even vertices are inherited from the previous subdivision level, odd vertices are inserted newly into the mesh. In the triangular edge midpoint subdivision the odd vertices

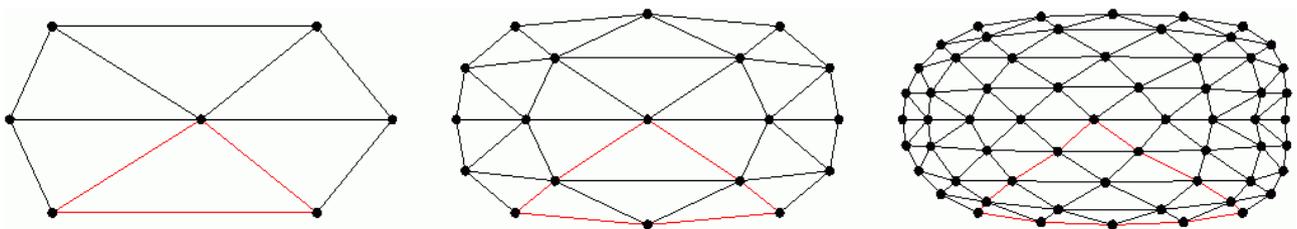


Figure 5.1: Triangular subdivision by face splitting. Over each edge a new vertex is generated and each triangle is split into four new triangles.

are inserted along the edges of the mesh of the preceding level. In this scheme, which will be applied in this work, the odd vertices have valency six if they are in the interior and four if they are on the boundary. Vertices with a valency unequal to six are called *extraordinary* vertices, all other vertices are *regular*. A net which is obtained after application of subdivision steps is said to have subdivision connectivity, which is necessary for multiresolution mesh editing<sup>1</sup> and wavelet analysis [Lounsbery et al., 1997].

Usually, the vertices of the refined mesh are affine combinations of the vertices in the previous level. The *subdivision mask* describes the vicinity of a point which is used to compute its position. At each vertex of this mask a weight factor is given. Usually, the subdivision mask is rather small, containing e.g. only the direct neighbors. If the rules for subdivision are suitable, the sequence of meshes converges towards a *limit surface*. Depending on the properties of the limit surface the schemes are called C1 or C2, for tangent plane and curvature continuous surfaces, respectively.

Subdivision schemes can be *interpolating* or *approximating*. In an interpolating subdivision scheme all computed vertices lie on the limit surface. In interpolating triangular face split subdivision schemes the even vertices at each level have the same positions as on the previous level. In such approximating schemes new positions are computed not only for the odd vertices but also for the even ones. In [Nasri and Sabin, 2002] modifications for approximating subdivision schemes are presented which lead to the interpolation of one prescribed point. Likewise, modifications for the interpolation of prescribed tangent planes are given. In 'subdivision surface fitting' (e.g. [Ma et al., 2002]) an initial control mesh is generated from the original mesh of points, so that the limit surface interpolates the given points. This involves the solution of a global system of equations.

The question of continuity is central to the analysis of a subdivision scheme. It has been observed (e.g. [Zorin and Schröder, 2000]) that there is generally a tradeoff between surface quality and interpolation constraint for approximating versus interpolating schemes. Requiring that the limit surface interpolates the points generated in each level leads to surfaces with oddly varying curvature.

Subdivision has many advantages which will be given in the following overview:

- Arbitrary topology can be handled by subdivision. Furthermore, the genus of the surface to be subdivided need not be known. Especially, the triangulation (i.e. a graph) need not be planar.
- Level of detail (LoD) is built into subdivision automatically. The desired level can be achieved by the number of subdivision steps. In *adaptive subdivision* the refinement is applied only to those parts of the mesh which should have a finer representation (e.g. because those parts are closer to a viewers eye, whereas the background is represented sufficiently in a coarser resolution).
- Subdivision provides a polygonal mesh, which is very useful for rendering or intersection algorithms. Parametric patches are usually converted to polygonal meshes for drawing, but for subdivision this is not necessary.
- The subdivision approach is very efficient, because only simple rules are applied (affine combinations). This does not only provide numerical stability, but it also leads to simple programs.
- The methods are local, because the subdivision masks are restricted in size. Changing a point of the original net influences the sequence of meshes and therefore the limit surface only in a restricted neighborhood.
- If the new points are affine combinations of the points on the previous level, then the limit surface is coupled in an affine invariant way to the original mesh.

In each step the number of points, edges and faces (triangles) is increased. If  $p$  is the number of points,  $t$  the number of triangles,  $e$  the number of edges,  $g$  the genus of the surface, and  $i$  the number of inner regions (i.e.

---

<sup>1</sup>There are also some approaches for multiresolution mesh editing which do not require subdivision connectivity, e.g. [Eck et al., 1995].

not necessarily triangular faces, which are not subdivided), than it holds<sup>2</sup>:

$$2 - 2g = (t + i + 1) + p - e$$

In triangular subdivision based on the edge midpoints the number of triangles  $t_s$  after one subdivision step is quadrupled ( $t_s = 4t$ ) and the number of edges  $e_s$  in the new graph is twice the number of old edges plus three times the number of faces ( $e_s = 2e + 3t$ ). Therefore, the new number of points  $p_s$  can be computed with:

$$p_s = 1 - 2g - t - i + 2e$$

## 5.2 Method overview

### Loop

The Loop scheme ([Loop, 1987] and [Zorin and Schröder, 2000]) is an approximating triangular subdivision scheme. The subdivision coefficients and masks are given in Appx. C.1 and an example can be seen in Fig. 5.2. As the scheme is approximating, it is not directly applicable to the posed problem, however, it produces nice surfaces. It is C1 everywhere and C2 at regular vertices. The mesh converges fast towards the limit surface. The boundary curves of the Loop scheme are B-splines of degree three. For regular triangular meshes the limit surface is composed of polynomial surface patches of degree four.

It is also possible to compute the location of a vertex on the limit surface (i.e. the location the vertex obtains after applying infinitely many subdivision steps) and e.g. surface tangents in such a point.

### Modified Butterfly

The Modified Butterfly [Zorin et al., 1996] is an interpolating triangular scheme. It is C1 at all vertices. The subdivision coefficients and masks are given in Appx. C.2 and an example can be seen in Fig. 5.3. Unlike the Loop scheme, the Butterfly subdivision limit surface is not composed of polynomial patches.

As mentioned before, this scheme is interpolating. This means, that all computed vertices lie on the limit surface. The subdivision mask is larger than for the Loop scheme (see Appx. C.2), and for this reason the computation of surface tangents to the limit surface is a little more expensive than for the Loop scheme.

### Variational Subdivision

In [Kobbelt, 1998] a variational approach to subdivision is presented for triangular meshes. In this context the subdivision is considered as a split operation, which increases the number of degrees of freedom. This is followed by a smoothing step. The new points are inserted at the midpoint of the edges, not only in the topological sense, but also in the geometrical. This mesh is smoothed by minimizing a functional which is an approximation of the sum of squares of principle curvatures. In the first subdivision steps the newly inserted points are allowed to move in order to minimize the functional. This leads to a sparse linear equation system for the complete mesh. In the following subdivision steps, all points except those of the original mesh are allowed to move in each smoothing phase. This is an interpolating scheme in the sense, that the original points are interpolated by the limit surface.

<sup>2</sup>This is a simple specialization of the Euler formula, which says that the genus of a closed polyhedron, i.e. a solid with  $f$  planar faces, is given by:  $2 - 2g = f + p - e$ . The terrain is usually not closed, thus a 'closing' face is necessary to turn it into a polyhedron, which contributes the '1' to the number of faces. The number of inner regions  $i$  stays constant during subdivision, each inner region can be closed by one face, too. Planar triangulations ( $g = 0$ ) of the terrain can be mapped onto a plane and the '1' in the formula stands for the 'closing' face which is bounded in the interior by the map of the outer edges and has no outer bound.

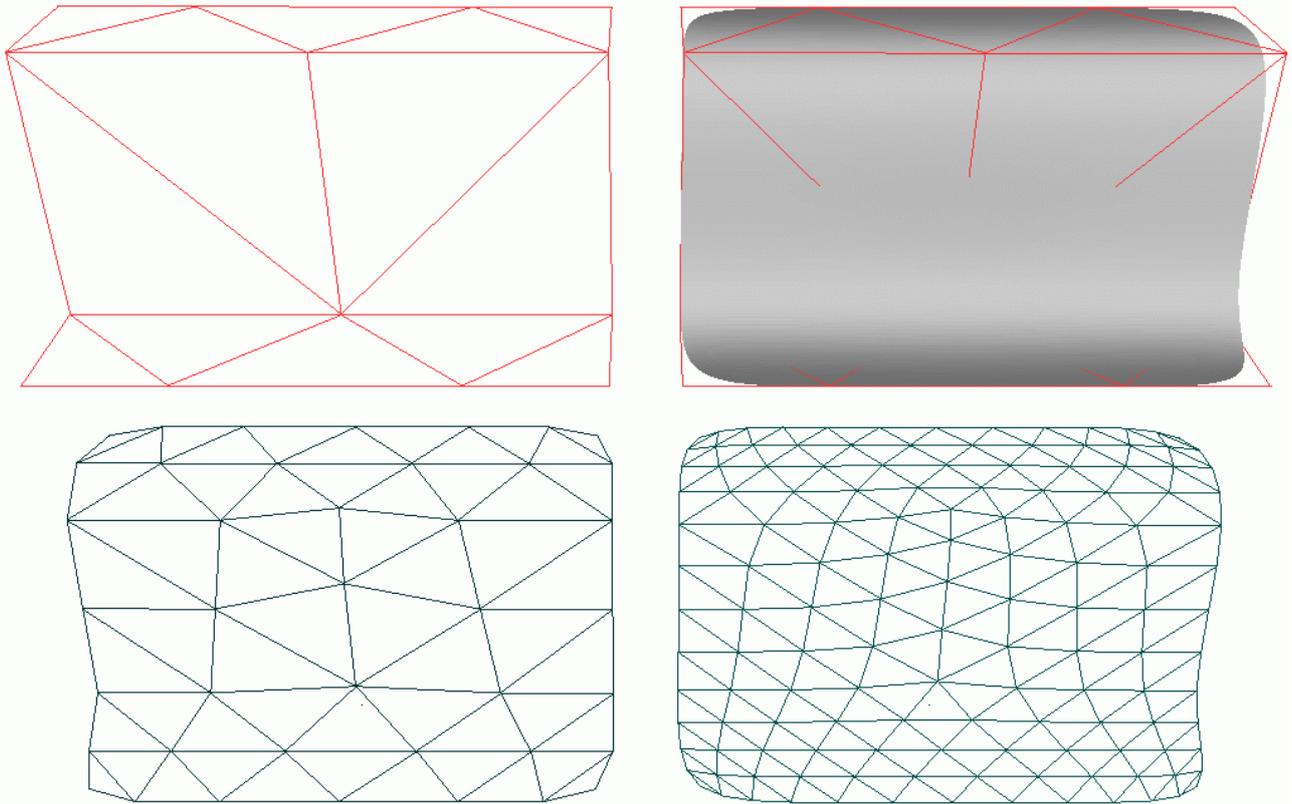


Figure 5.2: Subdivision with the Loop scheme, Upper left: given triangulation of two horizontal and one vertical surface part. Lower part: triangulation after one and two subdivision steps, respectively. Upper right: surface after five steps of subdivision together with the original triangulation.

## Curve network

An interpolating triangular scheme where the odd vertices are not affine combinations of the even vertices has been proposed in [Botsch, 1999]. The subdivision rules are of a constructive nature. First, in each vertex a normal vector is estimated. Next, planar cubic curves are determined for each edge with methods as in Sec. 4.3.2). The midpoint of each curve is taken as new point over the corresponding edge. With a suitable method for the estimation of the normal vectors this scheme is able to reproduce parabolic surfaces [Botsch, 1999].

## Short Comparison

In Figs. 5.2 and 5.3 the same triangulation is subdivided. It consists of 14 triangles and 14 vertices, only two of which are interior vertices. There is no regular vertex in the triangulation. As it can be seen, the newly inserted vertices are all regular, only those on the boundary have valence four. It is evident, that the surface with the Loop subdivision scheme is nicer, but it only approximates the original vertices. The Modified Butterfly scheme succeeds in interpolating the points and the surface is smooth, too. However, unwanted oscillations appear on the entire surface. As mentioned above, this tradeoff has also been observed in [Zorin and Schröder, 2000].

The Loop and the Modified Butterfly scheme have been extended to work near ‘crease edges’. At these edges the tangent plane continuity is not required. Additionally, the points along the crease should only depend on the neighboring crease points and not on other surface points. The formulae for Loop and Butterfly are also given in [Zorin and Schröder, 2000]. In photogrammetric terms these ‘creases’ are breaklines.

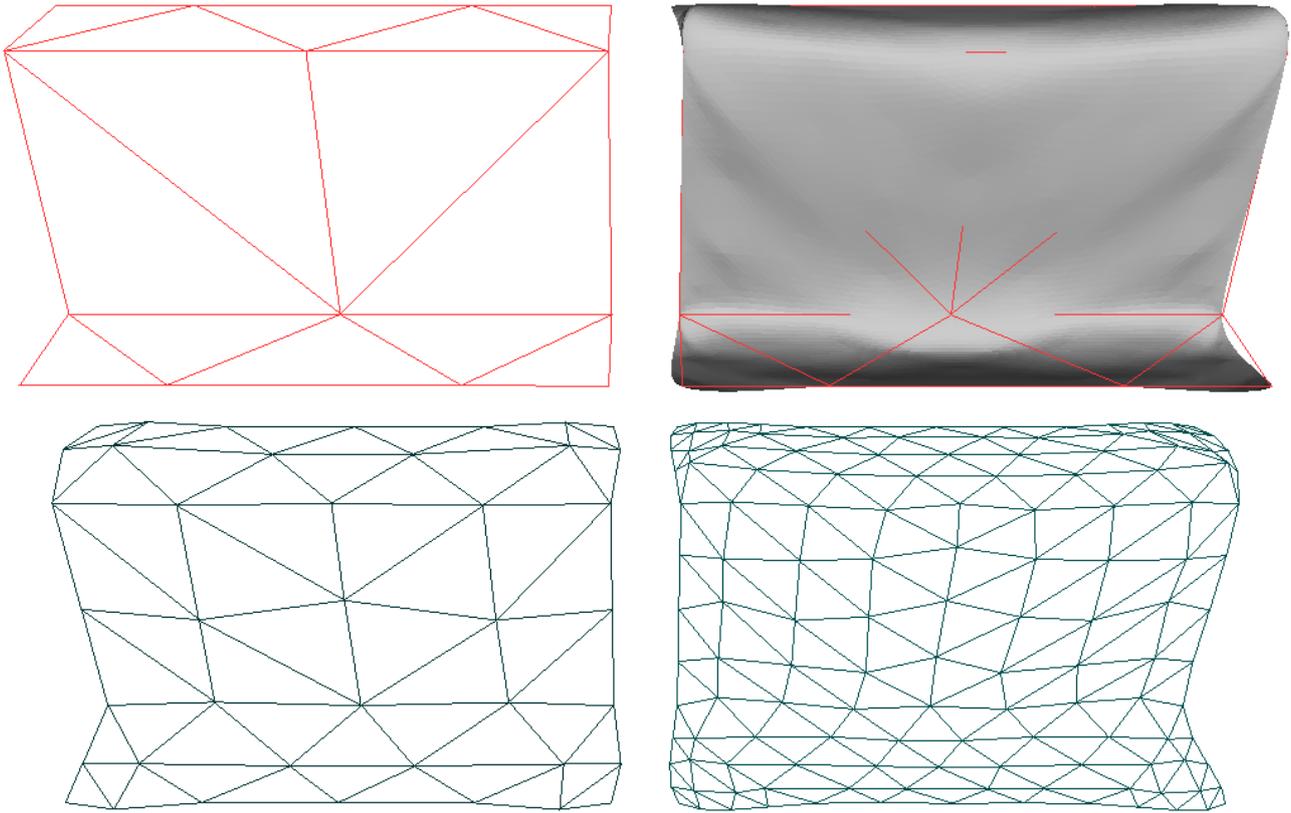


Figure 5.3: Subdivision with the Modified Butterfly scheme, Upper left: given triangulation. Lower part: triangulation after one and two subdivision steps, respectively. Upper right: surface after five steps of subdivision together with the original triangulation.

### 5.3 Subdivision by estimation of local surfaces

The above mentioned schemes are not capable of fulfilling the required geometric conditions as demanded in Sec. 2.4. For the first two schemes the rules near special points (and lines) would have to be altered in order to fulfill the requirements. As mentioned above, for breaklines and special points this has already been done. The incorporation of other special features (form lines, ...) would have to be developed. The third approach leads to a global system of equations. This does not only increase computation time, but also prohibits local updates of surface parts, e.g. when new measurements are available in a certain region.

The approach presented here assumes that there is an underlying surface which has been sampled by points. After triangulating the points, it proceeds by first determining in each vertex a local surface. This local surface is an approximation of the underlying surface in the specific vertex. It is therefore assumed that the local surface can be estimated from a vertex and its neighbors. In this estimation process the geometric boundary conditions (peak points, ...) can be considered. In order to compute a new point over the midpoint of an edge, a representative 'mid edge point' is computed on the local surface in either end point. These two points are averaged to obtain the location of the new point.

As approximating surfaces, surfaces of order two are suggested because they can adapt to the local curvature. As it will be demonstrated in the following section, tangent planes (order one approximations) are not sufficient. Higher order approximations, on the other hand, would increase the computation amount notably.

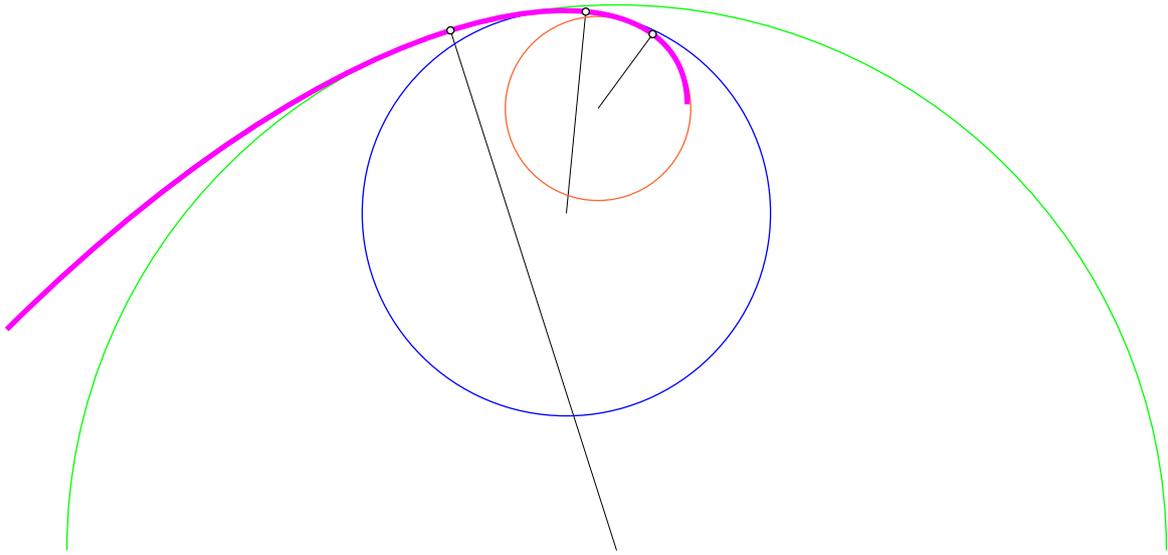


Figure 5.4: Curve (thick line) with osculating circles at three points. The radii from the circle centers to the points of second order contact are drawn as thin lines (in black).

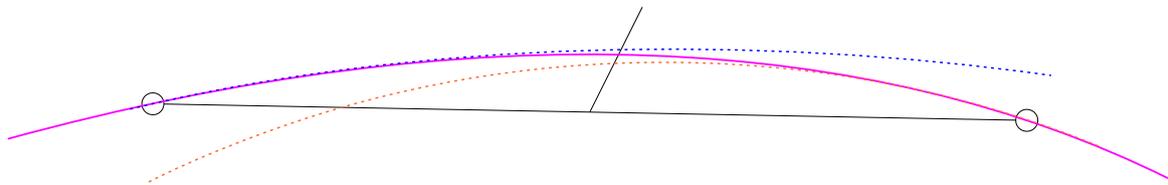


Figure 5.5: Detail with the two smaller circles of Fig. 5.4 (rotated). The dotted lines (in blue and orange) are osculating circles in the indicated points, which are connected by an edge. In this interval the curve runs in between those two circles.

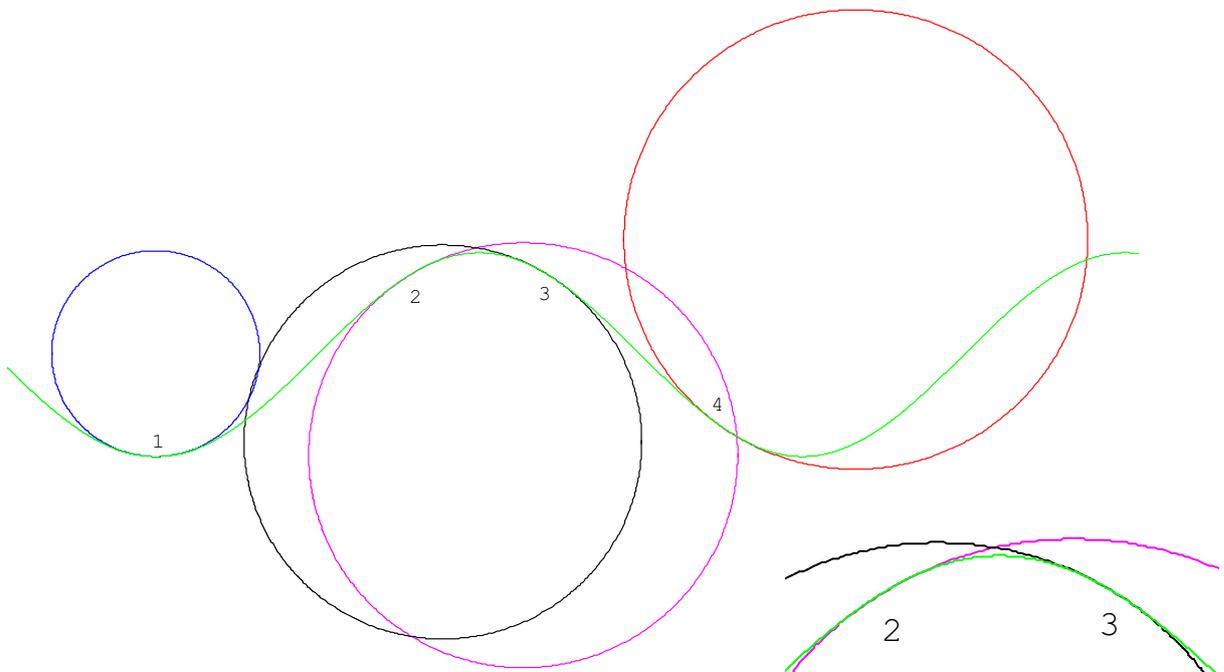


Figure 5.6: Osculating circles to a sinus-curve (bright line (in green)) in four points, with an enlarged detail in the lower right.

### 5.3.1 The curve case

In this section a motivation for this approach will be given and some properties of the scheme will be investigated for the curve case. First, properties of osculating curves will be discussed. This is followed by the explanation of the subdivision scheme for curves.

#### Osculating circles and parabolae

In Fig. 5.4 a curve is shown which has increasing curvature from the left to the right (thick line). Along this curve the osculating circle is shown at three positions. In the point of contact the circle touches the curve (i.e. circle and curve have the same tangent direction) and the curve also has the same curvature as the circle<sup>3</sup>. It would also be possible to draw osculating parabolae, ellipses, ... (generally: conics) to the curve. The vertex of such a conic section (not to be confused with the term vertex in the triangulation) does not necessarily have to be the point of contact. With other words, the direction of the axis is not determined by the osculation requirement.

As the curvatures of the circles are constant and that of the curve in Fig. 5.4 increases, the curve changes from the outward sides of the circles to the inward sides in the points of osculation. If the curvature is increasing continuously between two points of the curve, the curve will be within 'bounds' defined by the two osculation circles (Fig. 5.5). So, the second order approximation in one end point of the interval 'overestimates' the run of the surface and in the other end point it is 'underestimated'. It can also be said that one approximation overshoots in the interval and the other one undershoots<sup>4</sup>. Of course, the curvature may change the sign between two such points. In this case one circle is left of the curve and the other one is on the right hand side. However, these bounds are only well defined, if the curvature behaves monotonously. Otherwise, e.g. if the two points are before and after a point with a local maximum of curvature, both circles will overshoot in the interval. These situations are shown in Fig. 5.6. Between the points 2 and 3 a maximum in the curvature can be found. Thus, both of the osculating curves overshoot in this interval. Between the points 3 and 4 the sign of the curvature changes, but the curve is still 'bounded' by the circles in this interval. The interval between points 1 and 2 is too wide and the osculating circles do not 'bound' the curve anymore.

#### Subdivision rule

To curves the subdivision is applied analogously as to surfaces, as roughly sketched above. Given is a polygon which is iteratively refined by inserting a new point between each pair of adjacent existing points. One such interpolating scheme – the four point interpolatory subdivision scheme – is also mentioned in Appx. C.2. The scheme presented here proceeds by estimating local curves (e.g. parabolae) in each point of the polygon at a given level. These should approximate the curve in a certain neighborhood. To obtain a new point to be inserted over an edge the local curves in the edge end points are used. From each curve one point is computed which is – in some sense – close to the midpoint of the edge. These two points are averaged to form the point on the next subdivision level. One possibility is demonstrated in Fig. 5.5. A line incident to the edge midpoint and with a certain direction, e.g. the mean of the normals to the circle tangents in the vertices, is intersected with the two approximating curves. A different line for the intersections would be the line passing through the edge midpoint which is perpendicular to that edge. The intersections yield two points which are, as mentioned above, averaged to obtain the new point.

If a polygon is given which is sampled from an underlying curve, the above subdivision procedure will insert new points close to the 'true' (but unknown) curve points. A new point is within tight bounds prescribed by the second order approximations, which are estimated from the neighboring points. Therefore, a good (and reliable) estimation of the local curve is necessary. Furthermore, the sampling of the curve must avoid situations as those shown in Fig. 5.6 in the intervals [1,2] and [2,3].

<sup>3</sup>For a planar curve  $\mathbf{x}(t)$  the curvature  $\kappa$  is defined as  $\kappa(t) = \mathbf{x}'(t) \wedge \mathbf{x}''(t) / \|\mathbf{x}'(t)\|^3/2$ .

<sup>4</sup>This is also the reason, why an approximation with the tangent (planes) would not be sufficient. In the explained example, all tangents lie on the same side of the curve. A "cancellation" of overshooting and undershooting is therefore not possible. Only in an inflection point the tangent lies on both sides of the curve.

### 5.3.2 Surface subdivision with approximating surfaces

In this section the scheme will be described for surfaces. First, the term osculation will be defined in this context. This establishes analogy to the curve case and the rationale for the curve subdivision scheme can be extended to surfaces.

Two surfaces are osculating in one point, if parameterizations exist for these two surfaces which are identical up to the second order elements (quadratic elements). A well known example is the Taylor approximant of a surface given in parameter form. If the development is stopped after the second derivatives, an osculating surface is obtained in the point of development. In general, second order approximations can be achieved with quadrics, if they are sufficiently general. Though a cylinder (parabolic or elliptic) is a quadric it is not sufficiently general, because in the direction of the generators the curvature is always zero. Ellipsoids can only be used at elliptic surface points. Paraboloids, and one and two sheeted hyperboloids are sufficiently general to osculate a given surface in a certain point. Some details on quadrics can be found in Appx. A.2. Similarly, vector-valued second order polynomial surfaces are sufficiently general for a second order contact in a prescribed surface point<sup>5</sup>. Estimating such surfaces in the vertices can also be seen as estimating the second fundamental form in each vertex. The described subdivision rule is then a method of blending – on a point basis – the fundamental forms of adjacent points.

The principle for the subdivision of surfaces is analogous to the curve case. The polygon is replaced by a triangulation and the approximating conic is extended to be a paraboloid or a general quadric, or a vector-valued second order polynomial. The parameters of this subdivision algorithm are:

**Local surface type** Quadrics, and in particular paraboloids, but also vector-valued second order polynomials have been used as local surfaces for approximating the underlying (triangulated) surface. As mentioned above, the local surface should osculate the underlying surface. If a paraboloid or a general second order polynomial surface is used – as in Sec. 3.3.3 – a parameterization has to be computed first. Of course, higher order surfaces could be used, too.

**Local surface estimation** The local surface has to be estimated by some method for each vertex from its neighborhood. There are a number of parameters for this estimation.

1. The neighborhood can be chosen differently, e.g. first generation neighbors, star-shaped neighborhood, ... (cf. Sec. 3.1).
2. The neighbors can be weighted in the estimation process in order to give more weight to points nearer to the vertex.
3. The center vertex can be interpolated or approximated.

Additionally, rules are necessary if the number of neighbors is not sufficient for computing the surface. This is the case if the number of surface parameters (e.g. five for a paraboloid that interpolates the center vertex, if the direction of the axis is prescribed) is smaller than the number of neighbors. In the case of equality the surface can show strong oscillatory behavior. For these cases some form of regularization is required.

**Averaging** Local surfaces of adjacent points are averaged to form a new point. For this end one point is computed on either surface. If parameterized paraboloids are used the points can be obtained by evaluating the local surface at the parameterization of the edge midpoint. A different method is to intersect both surfaces with one line and take the mean of the intersection points. This line should pass through the edge midpoint and its direction could be the mean normal vector of the local surfaces in the two vertices.

**Interpolating or Approximating scheme** The local surfaces can be determined in a way that the corresponding vertex is interpolated or approximated. In the second case, the even points have to be updated, too, and moved to their positions on the local surface. Otherwise, an inconsistency is obtained, because the surfaces which were used for the odd points do not fit to the positions of the even points. If an interpolating surface is used, the even points remain at their positions. Like in the variational subdivision

---

<sup>5</sup>Paraboloids are a special form of both, quadrics and vector-valued second order polynomials.

scheme mentioned above, it is possible to determine interpolating local surface for the original vertices and approximating local surface for all inserted vertices. In this case only the original points keep their positions in the subdivision levels. This approach will be called ‘interpolating/approximating’.

The question whether an interpolating or an approximating scheme should be used can also be answered by looking at the given data. If it is assumed to be free of errors, an interpolating scheme is appropriate. In this case the neighborhood for the determination of the local surface could also be smaller. If the points are free of error, the second generation neighbors would not contribute essential new information to the estimation of the approximating surface.

Independently of using an interpolating or approximating scheme, the same limitations apply as before in the curve case. The sampling must be dense enough in order to allow a reliable determination of the approximating local surfaces. Two local surfaces of adjacent points must not be too far apart in order to bound the underlying surface in the region of the edge.

### Geometric boundary conditions

The estimation of the local surface can be restricted in different ways if necessary. If a vertex is known to be the top of a hill, it means that its tangent plane is horizontal. This can be considered in the estimation of the local surface. It could also be considered in the estimation of the local surfaces of the neighboring points. However, the neighboring local surfaces should represent the underlying surface in their surrounding as good as possible, and such a requirement would reduce the available degrees of freedom for this task. Therefore, it is suggested that the requirements are only considered in the points where they have been established. This will lead to a limit surface with a horizontal tangent plane in this point, but not necessarily the highest point in its surrounding<sup>6</sup>. Of course, with the co-ordinates of the newly computed points it could be checked, if the requirements are fulfilled or not. In the latter case this is an indication of a bad approximation or too sparse sampling of the surface.

Breaklines can be considered in this approach, too. The breakline itself is subdivided with a curve subdivision scheme in order to maintain its independency from the surrounding points. Alternatively, the breakline can be given as a cubic spline or a similar curve. In this case the knots should also be vertices of the triangulation and the inserted points obtain the co-ordinates of the spline points to the corresponding interval midpoint. This means, that the shape of the breakline only depends on the breakline points. Across the breakline the surface has a discontinuity in the first derivative. Therefore, two smooth local surfaces are determined, one for either side. Each one approximates only the points on the breakline and on one of the two sides. For the vertices in the vicinity of a breakline, the approximating surface is determined without the points on the other side of the breakline. When computing an odd points over an edge which connects a breakline point and a normal point, the appropriate surface of the two computed ones in the breakline point is taken.

In the following sections different possible realizations of the subdivision algorithm (e.g. type of local surface) will be compared. The result of these studies on typical sample data will be presented.

### 5.3.3 Paraboloids vs. general quadrics as local surfaces

Using quadrics has two advantages. First, quadrics allow a better fit (i.e. better than of second order, though a third order contact is – in general – impossible) than paraboloids, and second, quadrics are conveniently described as a set of points fulfilling one equation (zero set). Therefore, only the co-ordinates and no parameterization is required for the computation. The disadvantage of the general quadrics is that the neighborhood relations, which are prescribed by the triangulation are not necessarily respected by the quadric. If two sheeted hyperboloids or (narrow) ellipsoids are found as solutions for the approximating surface, it can be the case, that neighboring points come to lie on different sides of the quadric (e.g. different sheets, ..., cf. Fig. 3.7).

<sup>6</sup>This is a problem of the mathematical description, which describes the surface shape only in a very local way. The tangent plane shall be horizontal, but this does not say, if the point is a local maximum, a saddle point, or a minimum and it does not forbid other maxima and minima near by.

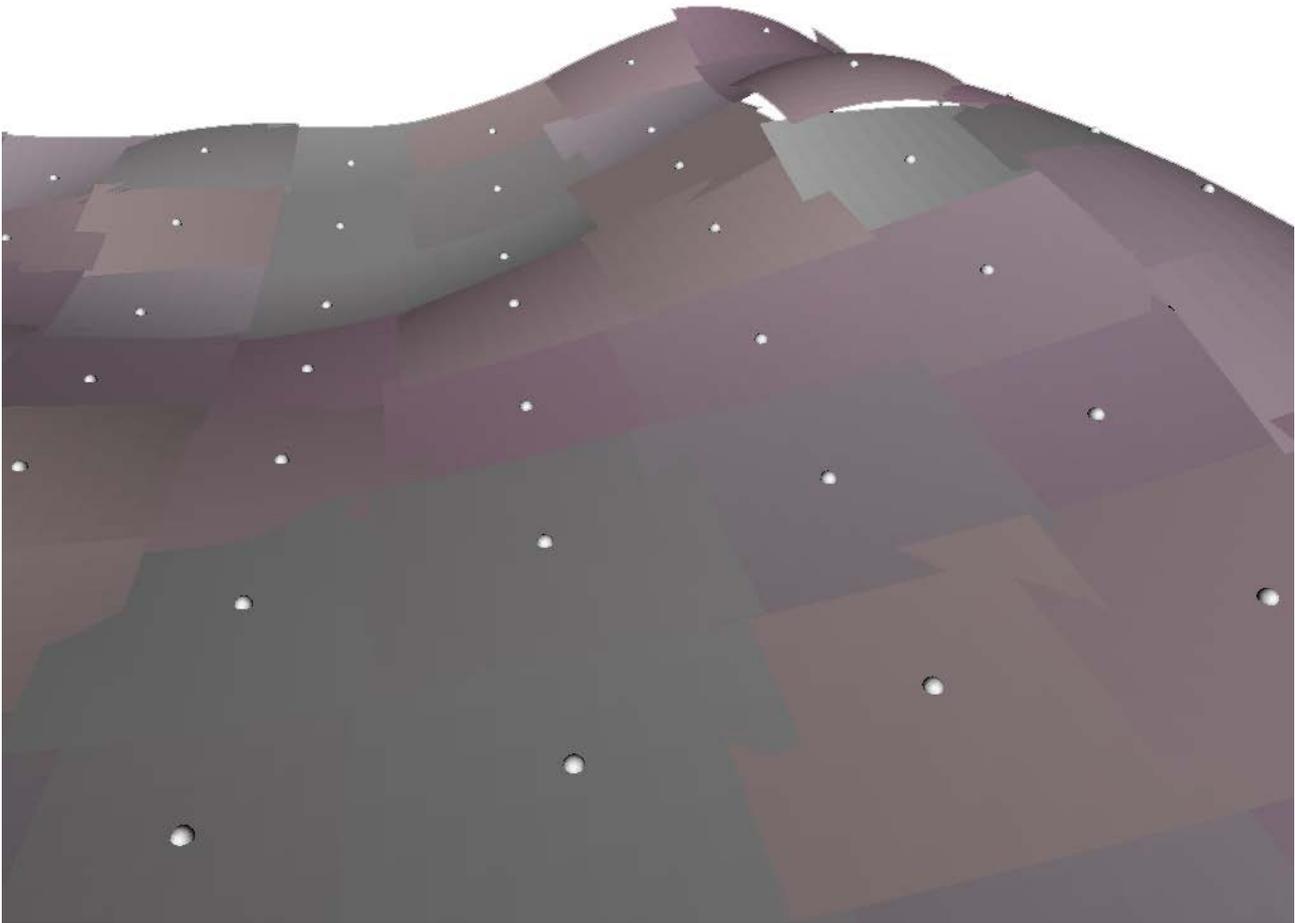


Figure 5.7: Estimated paraboloids for surface points. Each paraboloid has been given a color in order to make the paraboloids distinguishable. This figure is also shown in color as fig. E.4 on page 125.

Using a paraboloid in the parametric form this problem does not arise. First, because paraboloids have only one sheet, and second because the parameterization prescribes the neighborhood relations. Therefore, if the parameterization of the points which are to be approximated respects the neighborhood relations of the triangulation, the approximating paraboloid will do this, too. Of course, paraboloids can also be described as a set of points fulfilling the quadric equation and one additional side condition (a certain determinant must vanish). This would remove the need for a parameterization, but it cannot be guaranteed, that the neighborhood relations are respected. Therefore, the quadrics are less appropriate for the estimation of a local surface than the paraboloids in parametric form. Of course, in the latter case a parameterization of the points has to be computed first.

In Fig. 5.7 computed paraboloids are shown for a part of a surface. The original points are indicated as small spheres, the paraboloids are drawn around each vertex. In the shown example the vertices were given in a regular grid in the ground projection. The drawing extension of the paraboloids has been chosen in order to have a small overlap between neighboring surface approximations. In the overlap zone the new points are determined by averaging the two surfaces. In the foreground it can be seen that the paraboloids approximate the surface very well. There is hardly a discrepancy between adjacent paraboloids. Also the hyperbolic terrain area left of the ridge is described well. At the ridge one point stands out, because the local surface approximation is quite away from the neighboring approximations. This point is the peak of the ridge. At this point all neighboring points are below, therefore the estimated surface shows an elliptic behavior. The tangent plane to the local surface approximation is almost horizontal at this point, seen from above the surface is convex. In the neighboring points, it can be seen best left of the ridge, the type of the local surface approximations is hyper-

bolic or also elliptic, but concave. This explains the gap between the neighboring surface approximations. The new points which are inserted there will obtain positions in the gap between the two local surfaces.

For completeness, the complete algorithm for the calculation of the subdivision surface with interpolating quadrics will be described now. The described scheme is purely interpolating. For the computation of the local surface the neighbors up to the second generation have been used. In a regular vertex there are 6 neighbors and for the determination of the quadric which interpolates the center vertex at least 8 points are necessary. Therefore, at least the star shaped neighborhood is necessary. Additionally, the neighborhood can be restricted by breaklines. Thus, the neighbors up to the second generation were chosen. Only in very few cases, e.g. in vertices on the border or on a breakline, this neighborhood was not sufficiently large. In this case an adjusting plane was used instead of the quadric, but of course, more intelligent techniques could be used, too. However, the quadric subdivision scheme did not only work poor along the borders but also in the interior. In all cases, no weighting of the points was performed. To obtain an odd point, the quadrics at the two end points of an edge were intersected with a line which interpolated the edge midpoint. The direction of the line was the average direction of the quadric normals in the corresponding vertices. In cases where the line did not intersect one of the quadrics the odd point was set to the edge midpoint. If there is a real intersection between the quadric and the line, there are in general two intersections.

### 5.3.4 Paraboloids vs. second order polynomials as local surfaces

In this section two methods of local surface representations are compared: on one hand a paraboloid is used, and on the other a vector-valued second order polynomial surface.

If the density of sampled surface points is very high, the neighboring points of a given vertex  $\mathbf{P}$  will lie (almost) in this vertex's tangent plane  $\tau$ . In this paragraph it will be made clear that under these circumstances no large differences between the two methods can be expected. As the neighbors lie very close to  $\tau$ , a parameterization of the neighborhood by projection into  $\tau$  and by any of the methods of Sec. 3.2.4 will be (very) similar. These methods for local parameterization reproduce the triangulation itself, if the points are coplanar. In Appx. A.3 it is shown that a transformation of the local co-ordinate system in the determination of a vector-valued polynomial does not change the shape of the adjusted surface. Therefore, it is possible to find a rotation of the co-ordinate system  $(x, y, z \mapsto x_r, y_r, z_r)$ , so that two of three polynomials of the approximating surface are (almost) linear:  $x_r(u, v) \simeq u$ ,  $y_r(u, v) \simeq v$ . Under this rotation the co-ordinate directions of  $x_r$  and  $y_r$  will lie in (close to) the tangent plane  $\tau$ . In the adjustment these linear polynomials will be reproduced and the result for the third co-ordinate will be  $z_r(u, v) = au^2 + buv + cv^2 + du + ev + f$ . Exactly (more or less) the same will be the result for the paraboloid which is parameterized over the projection onto  $\tau$ .

Therefore, differences are to be expected if large angles between adjacent triangles occur. In this case the methods of parameterization (projection vs. Sec. 3.2.4) differ considerably. The example of the wall (Figs. 5.2 and 5.3) is used for comparison of the two local surface description methods (Fig. 5.8). In both cases the neighbors up to generation one and a half have been used and an interpolating/approximating scheme has been applied. The upper part of the figure shows the surface with paraboloids as local approximations and the lower part shows the same with second order polynomials as approximations. As it can be seen, the sampling is too low for both of the methods in order to produce a differentiable surface. At the inner points a small cusp remains. This cusp is obviously caused by poor surface approximations which did not reproduce the parabolic surface behavior<sup>7</sup>. However, the subdivision surface computed from the vector-valued polynomials is much smoother and also more fair than the other one. The cusp is smaller, too. In the upper surface, triangles of intermediate subdivision steps can still be seen in the limit surface.

From this example it can be concluded, that the use of second order polynomials produces fairer subdivision surfaces than the usage of paraboloids. The higher number in degrees of freedom allows a better approximation of the vertices. The computation efforts for the vector-valued surface and a paraboloid are comparable. The intersection is a little more complicated for the vector-valued polynomial than for the scalar-valued.

<sup>7</sup>Here the term 'parabolic behavior' refers to surface points where one of the principal curvatures is zero.

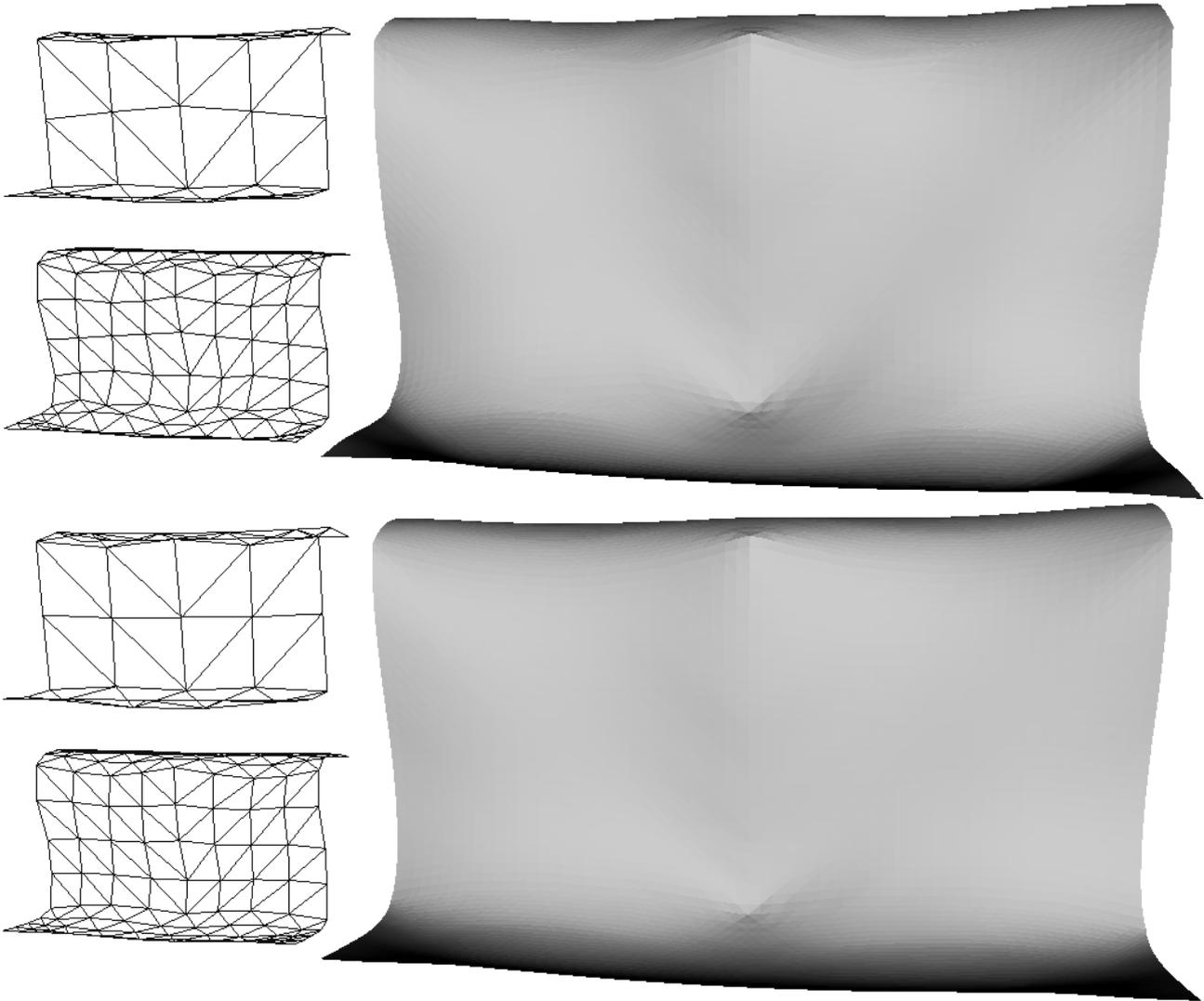


Figure 5.8: The wall is shown after the first, the second, and the fifth subdivision step. In the upper part the approximating local surfaces are paraboloids, in the lower parts they are second order polynomials.

### 5.3.5 Interpolation and Approximation

As it has been observed for the Loop and Modified Butterfly subdivision scheme, the approximating approaches generally produce nicer surfaces than the interpolating approaches. The interpolating approaches are especially bad on unfiltered data, i.e. data containing random measurement errors. In this case the surface has many faults. Better results have been achieved on real data after a filtering of the random measurement errors (cf. Sec. 3.6). If an interpolating scheme is applied, the triangulation is still visible in the final surface.

The approximating approach, on the other hand, diminishes details which are prescribed by the triangulation strongly. The surfaces are, however, visually smoother than in the interpolating schemes. The difference between a filtering of the triangulation and the filtering which is achieved by the application of an approximating scheme is that the filtering can be controlled much better with a qualified technique as kriging in the triangulation. These techniques allow the elimination of random errors without reducing the detail information too strong. By applying the approximating scheme, on the other hand, the filtering cannot be controlled easily. Of course, weighting can be used in the adjustment (i.e. in the determination of the local surfaces), but this does not provide the same control over the filtering. The approximating approach requires therefore, that the

sampling is denser and not only the relevant points, but more points are in the data set.

For the investigation terrain data with medium height variation has been used. Thus the paraboloids are sufficient as local surfaces.

The best results have been achieved with a mixture of interpolation and approximation as described above. If the sampling of the surface is performed sparsely, the few originally given points have to be interpolated. The newly inserted points are allowed to move 'freely' in order to increase the smoothness of the surface.

These findings can be seen in Fig. 5.9. These figures can also be seen enlarged in Appx. D. The upper left image shows the original data. It shows two streets which merge and undulated terrain in the foreground. As it can be seen, there are random measurement errors in the data which provoke thin steep triangles which have a gradient very different to the real terrain gradient at this spot (see also Sec. 3.6 and Fig. 3.9). These faults are removed in the filtered version which can be seen in the upper right image. All of the subdivision surfaces are shown after the third refinement.

The two images in the third row of Fig. 5.9 are based on the original triangulation (upper left). The left surface is obtained with a strictly interpolating subdivision scheme. For each vertex the local surface interpolates this point. This is valid on each level for the odd and the even points. As it can be seen, the faults are not only kept but even increased. The location of the original triangles can still be seen in the surface, but this is the case for all presented schemes with the pure interpolation property.

The right surface of the bottom row shows the surface obtained with an approximating scheme. The paraboloids do not only approximate the neighboring points of a vertex but also the vertex itself. This leads to an diminution of details. It can be seen best at the undulations in the foreground. The ridges have been levelled. However, the original triangles are no longer visible and the surface is altogether very smooth. The street which runs into the foreground contains some very thin triangles. At these, the surface still shows strong variation in (gaussian) curvature. A removal or levelling of these triangles would be necessary to obtain a high quality surface.

The left middle image shows a surface which is based on the original data and obtained with a subdivision scheme which is interpolating for the original points and approximating for the newly inserted points. Compared to the strictly interpolating approach only little improvements are obtained. This can be attributed to the thin triangles and the overall low sampling rate of terrain points.

Finally, the surface on the right side in the middle row is obtained with the same interpolating/approximating scheme as its left neighbor, but applied to the filtered triangulation. It can be seen that the surface at the faults has improved significantly. This stresses the need for a 'nice' triangulation as basis for the subdivision. Generally, and as mentioned above, the triangulation is very rough and so the subdivision surface is rough, too. The surface is less smooth than the lower right one (purely approximating on the original data), but contains more of the details.

From tests with this and other data sets the last method has provided the best results. The judgement has been based on visual inspection of the surface. The conclusion to draw from this section is to apply this interpolating/approximating scheme. The interpolating part ensures that the original measurements are considered correctly and the surface interpolates these points. If necessary, smoothing has to be performed in a separate, pre-processing step. Though it has not been applied here, long thin triangles should be removed first.

### 5.3.6 Averaging

Three methods of computing an odd point from the local surfaces in the even points have been investigated. All these methods use the edge midpoint in one way or the other. The methods are based on

1. the edge midpoints in the parameter space,
2. a line through the edge midpoint with the average direction of reference plane normals in the even vertices,



Figure 5.9: Subdivision with interpolating and approximating paraboloids. Upper left: original data, Upper right: original data after removal of random errors with kriging, Lower left: purely interpolating scheme on the original data, Lower right: purely approximating scheme on the original data, Middle left: scheme with interpolation for the original points and approximation for the newly inserted points on the original data, Middle right: interpolating/approximating scheme like on the left side but on the triangulation after removal of random noise.

3. a line through the edge midpoint with the average direction of the local surface normals in the even vertices.

Schematically the three methods are shown in Fig. 5.10.

The first method is the easiest to compute. The edge midpoint is located in the parameter domain of either local surface. It is the midpoint between the center vertex and the corresponding neighbor. For both surfaces the point is computed and the average is taken.

The second method is only applicable, if the parameterization plane has a geometric meaning. This is, for example, the case, if the parameterization is performed by projecting the points orthogonally onto a plane which is determined from the center vertex and its neighbors in an adjustment. The plane is, in this case, an approximation of the tangent plane. The two plane normal vectors of the even vertices are taken and averaged. A line passing through the edge midpoint in the triangulation and with this direction is intersected with the local surfaces. The two points are averaged. This method also offers a possibility to perform a check on the local surfaces. If the two vectors are too divergent (e.g. they enclose an angle larger than  $25^\circ$ ), the estimation of the odd vertex is not reliable. In such a case the edge midpoint itself can be taken as the odd vertex. The idea behind this approach is to detect automatically the rough regions in the data, where the estimation of the approximating surface does not work properly anymore.

The third method is very similar to the second one. The difference is, that the local surface normals in the even vertices are used for the computation of the direction of the intersecting line. This is a better approximation of the surface normal vector than in the second method. Additionally, it is also applicable, if the parameter plane has no geometrical meaning. The detection of rough areas can be performed in the same way.

In the tests, the third method has provided the best results for the visual appearance of the surface. If the two points which are computed on either surface have a relatively large distance from each other (compared to the edge length), the averaged point can be very far away from both surfaces. Therefore, those methods which intersect the two local surfaces with one line only provide better results. In this case the averaged point is always between the two surfaces. In Fig. 5.11 surfaces obtained with the three methods of averaging can be seen. As it can be seen, the differences in the surface occur where the surface is described badly by the triangulation (long thin triangles). These figures can also be seen enlarged in Appx. D.

The conclusion to be drawn from this section is that two neighboring paraboloids should be intersected with one line to obtain the two points which are averaged. This ensures that the point lies between the two paraboloids. The method based on the surface normals in the even points is applicable for the second order polynomials too, whereas the second method is not.

### 5.3.7 Roughness detection

In the above, one method to detect the roughness of the area automatically has been presented. If the normal vectors of the local surfaces in the end points of an edge are too divergent, then the terrain can be considered to be rough. In this case the dangers of obtaining surfaces with unwanted undulations is very high. One counter measure is to take the edge midpoint instead of the computed averaged point. While this measure is possible – and to some extent also successful – it is not very elegant. It would be better to make a slower transition from the averaged points of the two local surfaces to the edge midpoint. Depending on the angle between the two normals a weight could be determined. The subdivision point could then be a weighted average of the edge midpoint and the averaged surface point.

Other detectors for roughness are possible, too. One is to base the classification if a local surface is rough, or not, on the a posteriori unit weight error of the adjustment in the local surface determination. This did not prove particularly useful, because in some symmetric situations the unit weight error can be very small, although the surface is quite rough. One example is a vertex which resembles a mountain peak and the neighboring points lie approximately on the same height and have the same distance from the peak. In this case a paraboloid with vertical axis can interpolate the data very well, and a small adjustment error will be computed. However, if the terrain around the peak is very steep and therefore rough – or not – is not detected by this adjustment.

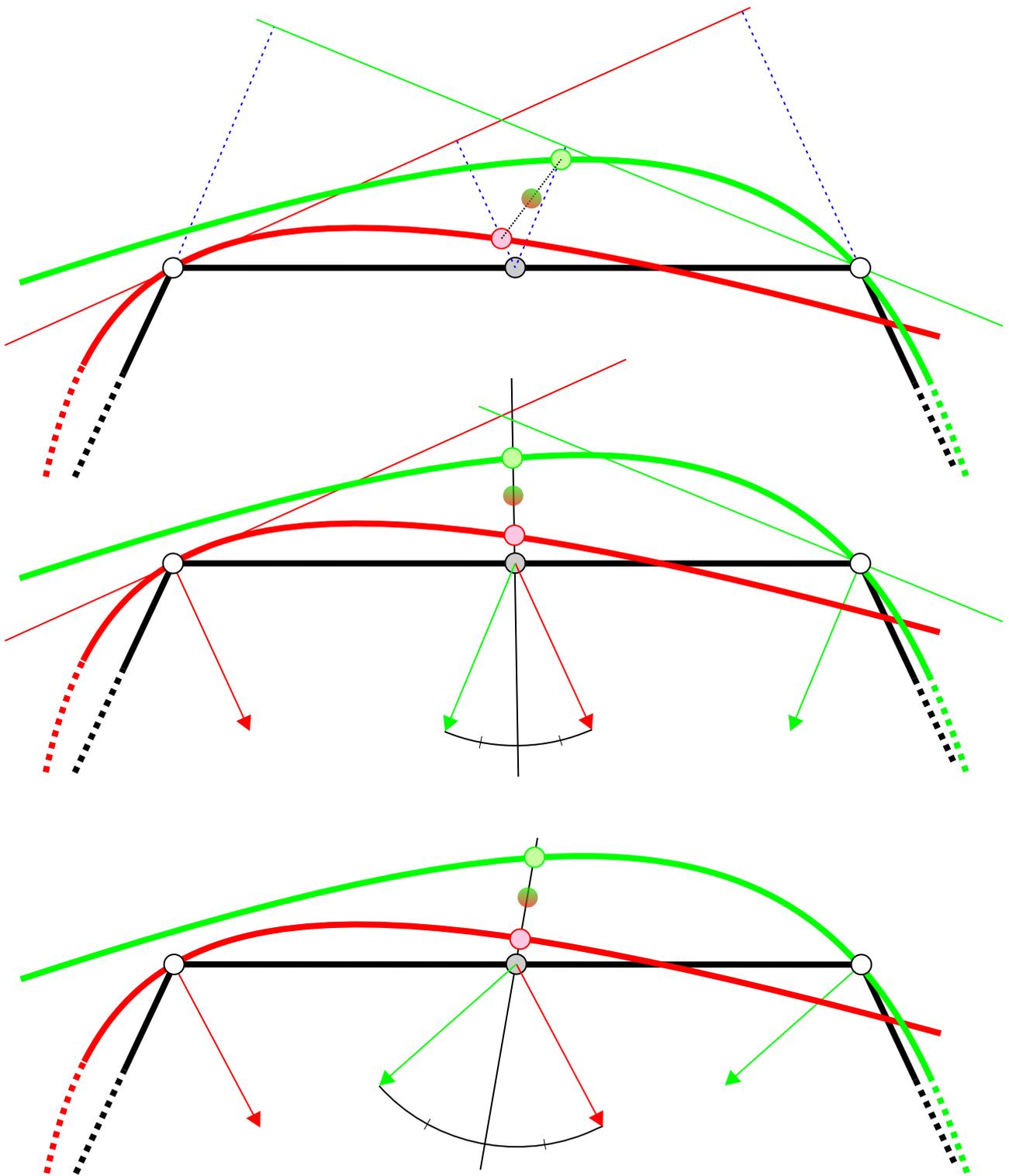


Figure 5.10: Three averaging methods to compute an odd point from the local surfaces in the end points of an edge. From top to bottom: use edge midpoint in the respective parameter space (dotted lines), line through edge midpoint with average direction of the reference plane normals in the even points, and line with average direction of the local surface normals in the even points.

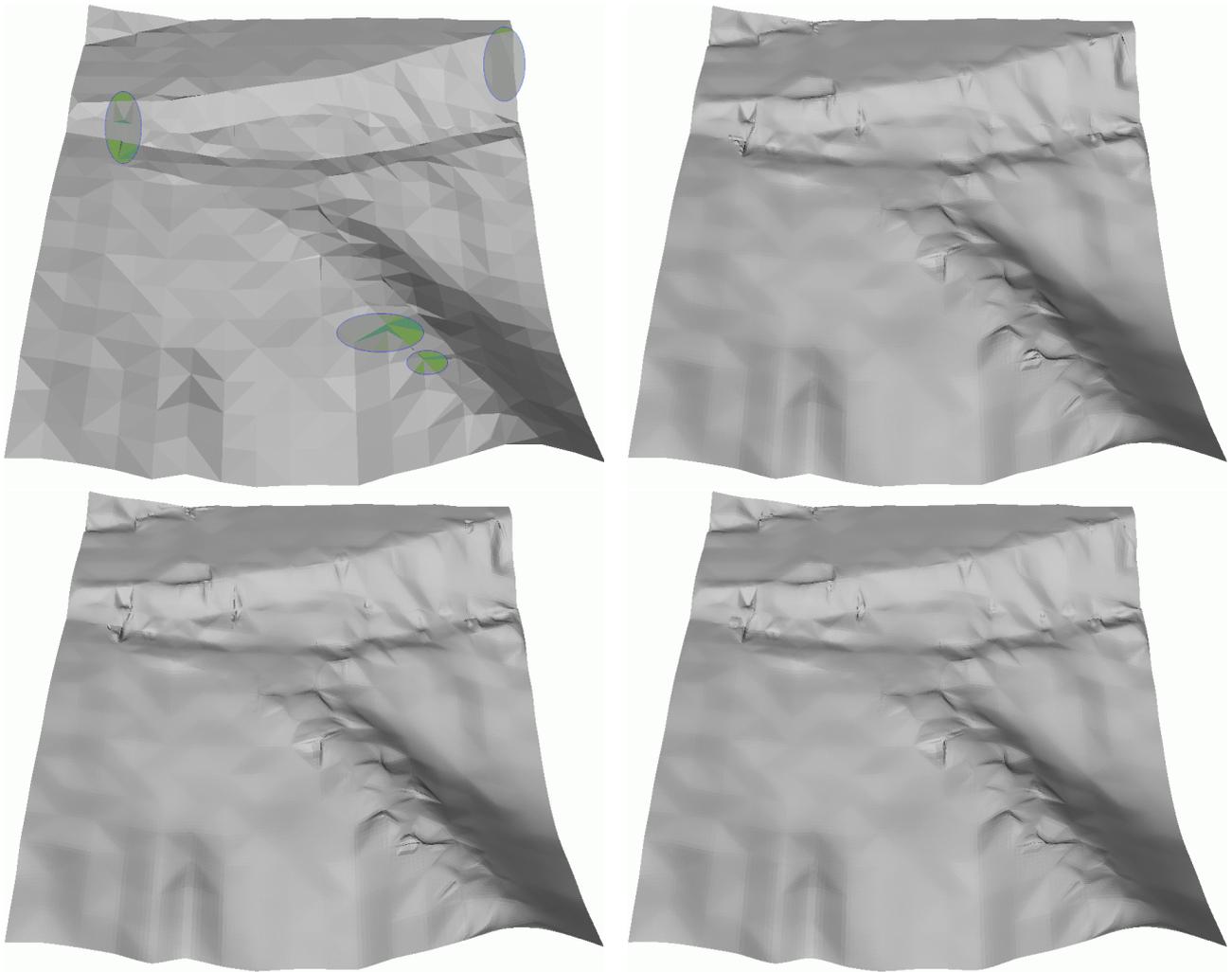


Figure 5.11: The three averaging methods applied to one example data set. At the top left the original data is shown and those areas where differences in the surfaces can be found are indicated. From top right to bottom right: based on the edge midpoint in the respective parameterization, based on a line with the average direction of the reference plane normals, and based on the average direction of the surface normals in the edge end points. The subdivision method is the interpolating/approximating scheme, the points have not been filtered before.

A further possibility is to check the distance from the averaged point to the edge midpoint. If it is too far away – relative to the edge length – this is also an indication of a poor sampling of the surface.

The drawback of these methods is, that the roughness is consider only very locally, i.e. independently in each point.

### 5.3.8 Results

In all test data sets the best results have been achieved with the interpolating/approximating scheme. The neighborhood should be one and a half generation. This ensures that in most cases enough points are available and the estimation of the local surface is over determined. In this way, wild oscillations as known from polynomial interpolation can be avoided. If fewer points are available, regularization is necessary. For real

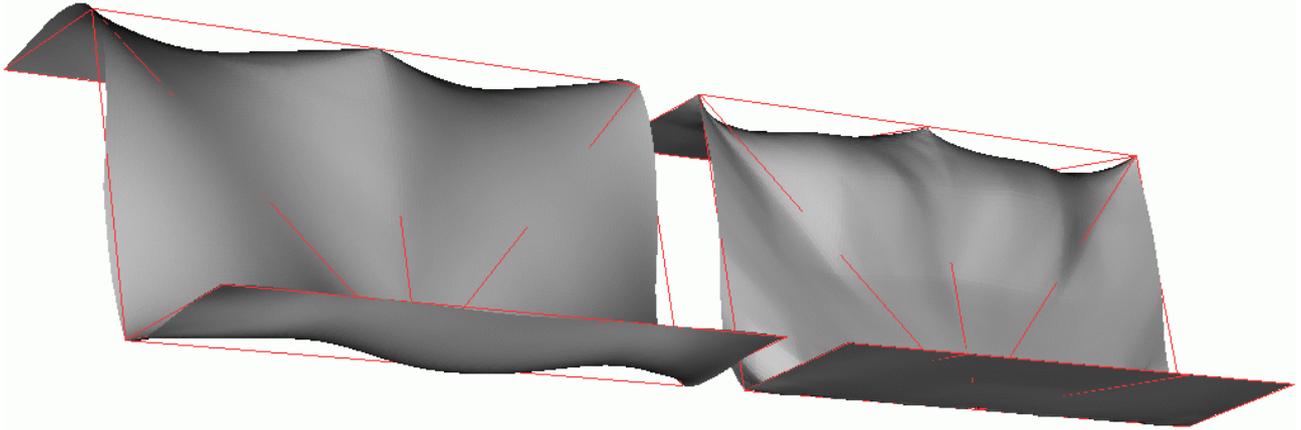


Figure 5.12: The interpolating/approximating scheme with paraboloids, left with a neighborhood of one and a half generations, right with the first generation only.

data, a filtering (smoothing) of the triangulation is necessary. Otherwise triangles with completely wrong normal vectors (i.e. different to the real terrain normal) can be found in the data. The computation of an odd point from the local surfaces in the adjacent even points should be performed by averaging the two representative mid edge points which are obtained by intersecting the local surfaces with a line through the edge midpoint and with the direction of the average of the surface normals in the corresponding local surfaces. If the terrain is very rough, a vector-valued second order polynomial is the better choice for the local surfaces than a paraboloid.

Though it is not shown here, a weighting of the points does not show much influence on the final surface. More examples would have to be computed to allow a more precise statement on this matter. Similarly, no systematic comparison on the impact of the parameterization method for the vector-valued polynomial surfaces was performed.

The main problem in this approach is to find good surface approximations. Often the paraboloids are not sufficiently flexible for approximating the local shape of the surface. As mentioned above, this can also be considered as an under-sampling of the surface. Using higher order local surfaces, means also using more neighbor points in order to determine the parameters. The gained flexibility would then be used – to some extent – in order to interpolate those points which are further away from the center vertex.

The number of neighbors used for the estimation of the local surface has also an impact on the shape of the subdivision surface. If the same local surface, e.g. a paraboloid, is used once with the generation one neighbors and once with the star shaped neighborhood, then the local surfaces tend to have smaller curvatures for the bigger neighborhoods. This can be seen in Fig. 5.12.

The question of continuity has not been answered in a mathematical way. In some examples it was not obtained to a satisfying degree. However, a method to produce a smooth surface which considers the photogrammetric requirements is the following. First, a number of subdivision steps with the described algorithm is performed. At and near the special regions (peaks, ...) the surface will show the desired forms. If the interpolating/approximating scheme is applied, the neighboring points of the original points will lie close to the tangent plane at these points. Then a known  $C1$  subdivision scheme (Loop or Modified Butterfly) can be applied to ensure mathematically the desired smoothness. Even if an approximating scheme is applied, the limit positions of the original points will not change significantly. Of course, this is a question of the number of subdivision steps with the first algorithm. At breaklines the extensions of the schemes at creases has to be used.

An example is shown in Fig. 5.13, where the triangulation of the vertical wall is subdivided three times with the local surfaces and then two more times with the Modified Butterfly scheme (upper surface) and the Loop

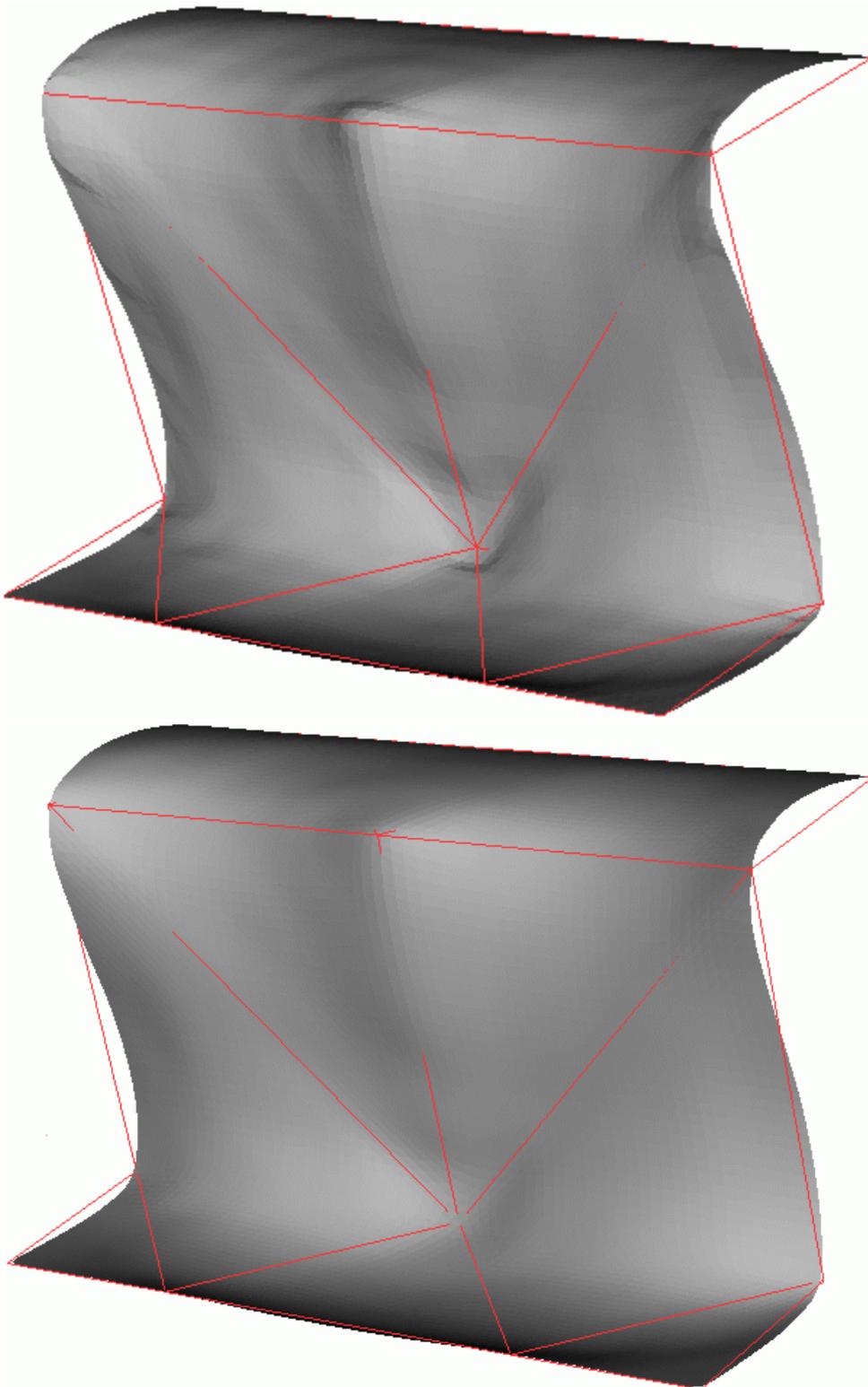


Figure 5.13: The vertical wall subdivided with the local interpolating/approximating scheme with vector-valued surfaces first (three refinement steps) and then with the Butterfly scheme (upper surface) and the Loop scheme (lower surface).

scheme (lower surface), respectively. The first image shows some unwanted oscillations, again, but the smoothness at the inner points is guaranteed. The lower surface is much fairer but only a quasi-interpolation is achieved. Of course, if the sampling was denser, then the limit positions would lie even closer to the original points.

Finally, it shall be repeated that the sampling density must be high enough in order to estimate the curvature (the local surface) reliably.

# Chapter 6

## Examples

In this chapter examples for terrain models reconstructed from different data sources are presented. Some of the examples are synthetic in order to investigate special aspects. For each data set different approaches will be compared. The comparison is performed primarily by visual judgement of the surface quality. Generally, there is no master model which can be considered to be superior to the others in advance.

### 6.1 Vertical Wall

This data has already been used in Secs. 4.3.4, 5.2 and 5.3.4. It shows a vertical wall which is bordered by two horizontal areas. The triangulation is shown in Fig. 6.1, left. To the right a parameterization of the triangulation is shown which is simply a development of the triangles. This is possible, because the example is a ruled surface.

In Fig. 6.2, upper left (cf. Fig. 5.2), the surface obtained from the triangulation by Loop subdivision is shown. As the scheme is approximating the given points are, in general, not interpolated. If the density of the mea-

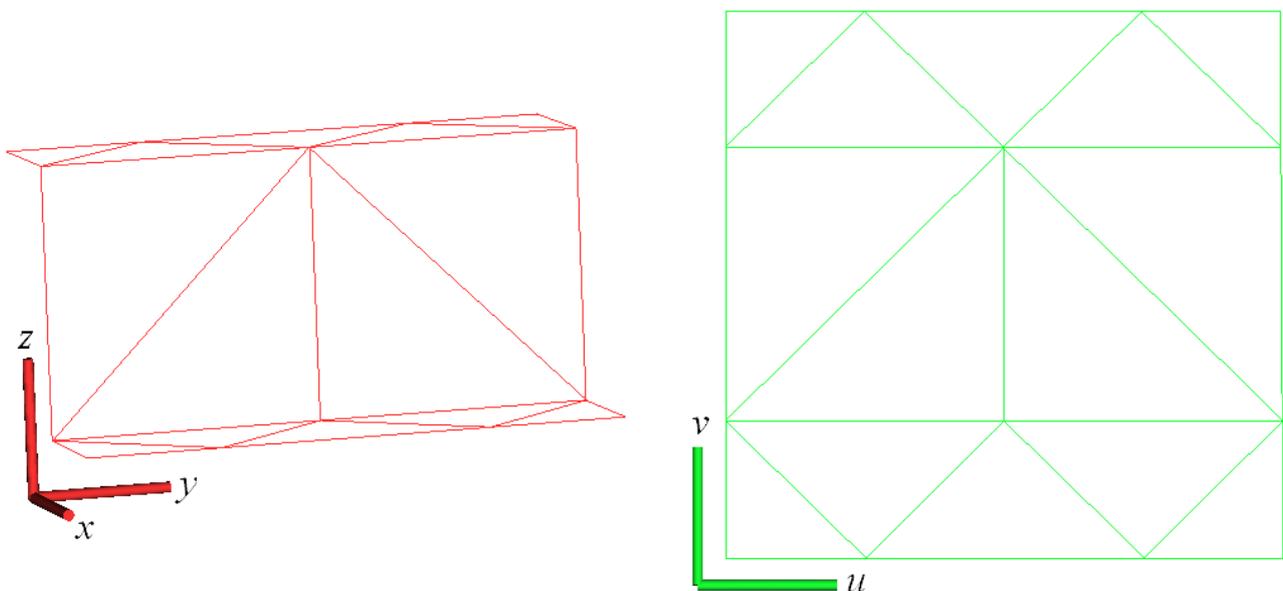


Figure 6.1: A vertical wall and its development, i.e. a parameterization.

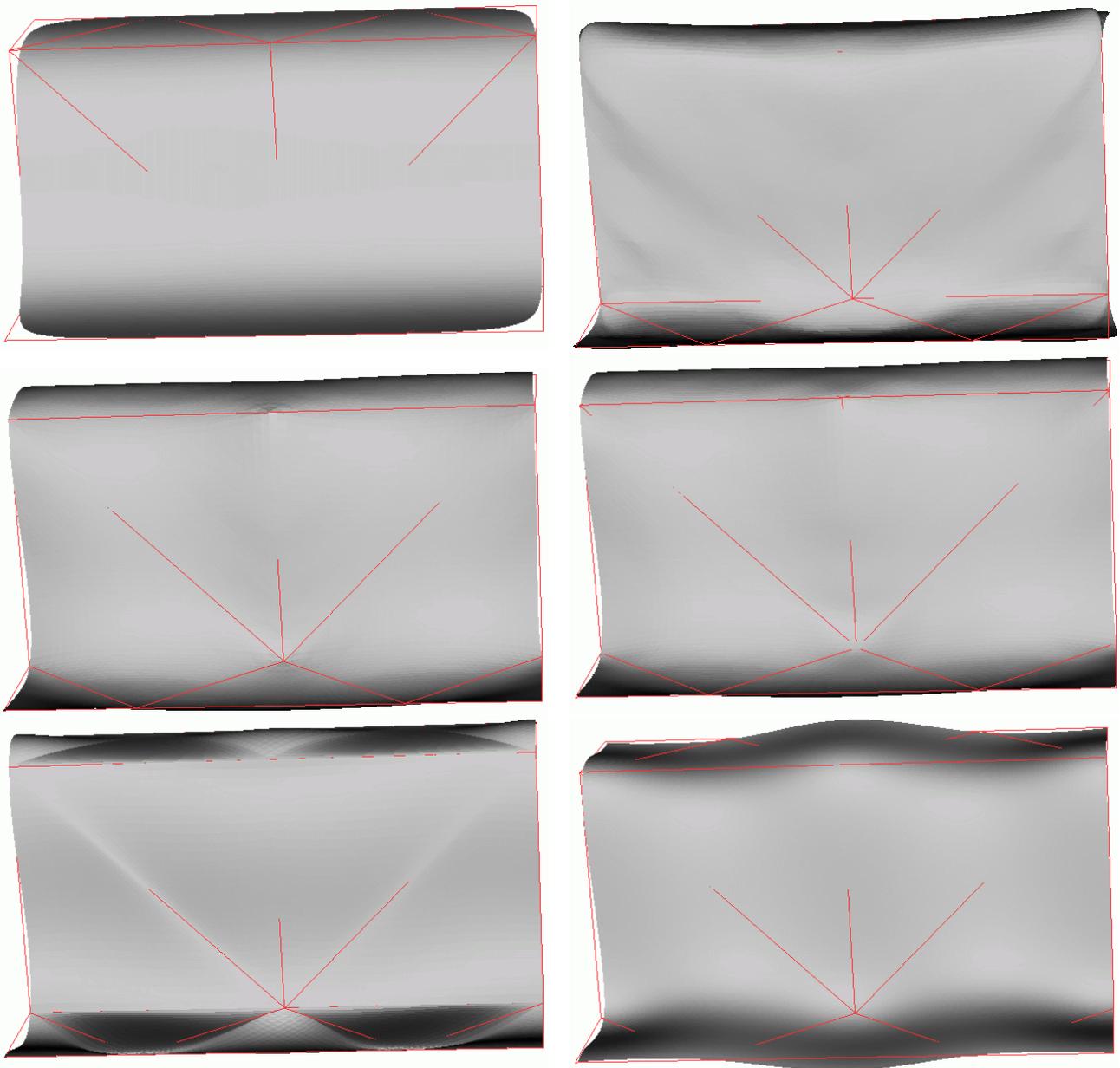


Figure 6.2: Terrain model of a wall, reconstructed with different algorithms, all shown with the same perspective. From left to right and top to bottom: subdivision surface with Loop refinement; with Butterfly refinement; with refinement by estimation of vector-valued second order polynomial surface; with three subdivision steps with the local surfaces first, followed by refinements with the Loop scheme; patch work surface with polynomial patches of degree four; surface by computing an  $x$ -,  $y$ -, and  $z$ -surface over a parameterization of the data.

sured points is very high, this may possibly be accepted. Except for the extraordinary vertices, this scheme is curvature continuous. At these the surface is 'only' tangent plane continuous. In Fig. 6.2, upper right (cf. Fig. 5.3), the surface under Modified Butterfly refinement is shown. The surface interpolates the given points but has unnatural wiggles. Those at the vertical terrain part have the same direction as the triangulation edges.

In contrast to this, the next surface (cf. Fig. 5.8) is overall fairer but has cusps at the two interior vertices. It was computed as described on p. 65 with subdivision by the estimation of local vector-valued second order polynomial surfaces. This surface interpolates the given points. The fourth image (middle right, cf. Fig. 5.13) shows a subdivision surface where the first three refinements were performed with the local surfaces and the following refinements with the Loop scheme. This has the advantage that the geometric boundary conditions and the interpolation requirement are considered approximatively, depending on the number of subdivision steps with local surfaces, and that the smoothness is guaranteed for the limit surface. Note, that this approach is not interpolating, but the distances from the surface to the given vertices is very small.

In Fig. 6.2, lower left (cf. Fig. 4.8) the patch work approach from Chap. 4 is used, with regularization. The smoothness could not be achieved, additional splitting would be necessary. For the last surface a 3D kriging approach was used. Over the parameterization (Fig. 6.1, right) three surfaces, one for each co-ordinate direction, were computed. The (simple<sup>1</sup>) kriging system was solved with SCOP [I.P.F., 2001]. The given points are not interpolated exactly, because a small filter value was used in the kriging matrix. The strength of the filtering can be prescribed in the units of measurement.

The only surface reconstruction approach which reproduced the parabolic behavior of the surface (zero gaussian curvature) is the Loop subdivision surface, which is approximating. The interpolation of a vector-valued surface over the parameterization (generally obtained by solving a global equation system) is not only left-right symmetric, but also symmetric to a horizontal axis because the edges of the triangulation are not of importance in the parameter plane. These special reconstruction features (e.g. parabolic behavior) do not only depend on the given data. If the given data is not aligned as symmetric as in this example, such features are usually not reconstructed.

---

<sup>1</sup>cf. footnote on p. 15

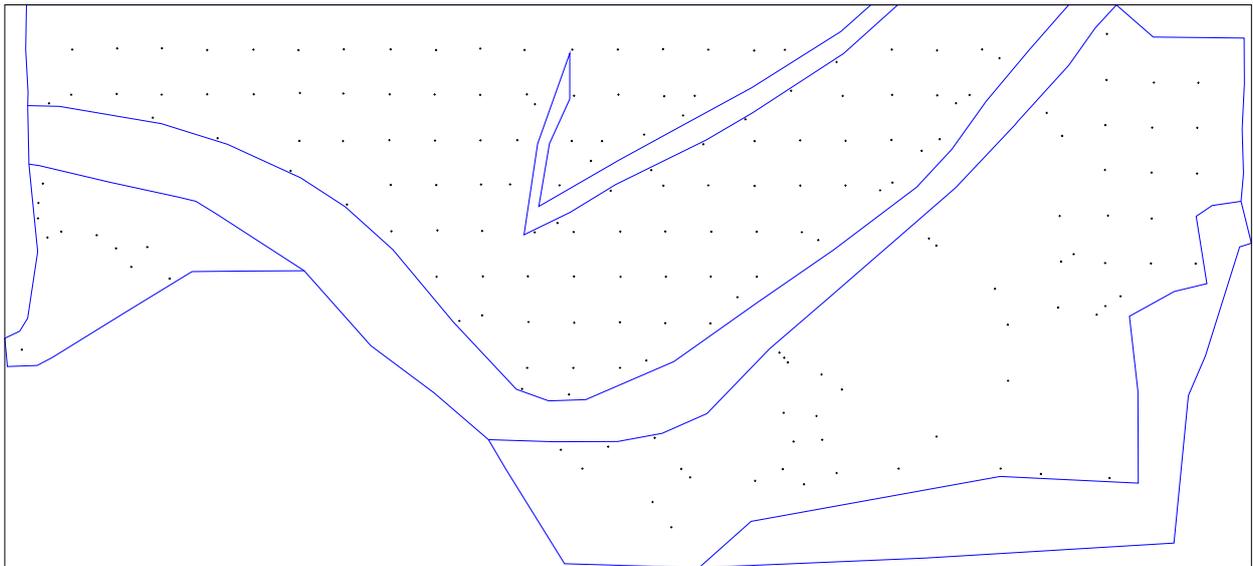


Figure 6.3: Data set “Elev”: small dots indicate point measurements, the inner lines are breaklines, the others delineate the border of the area.

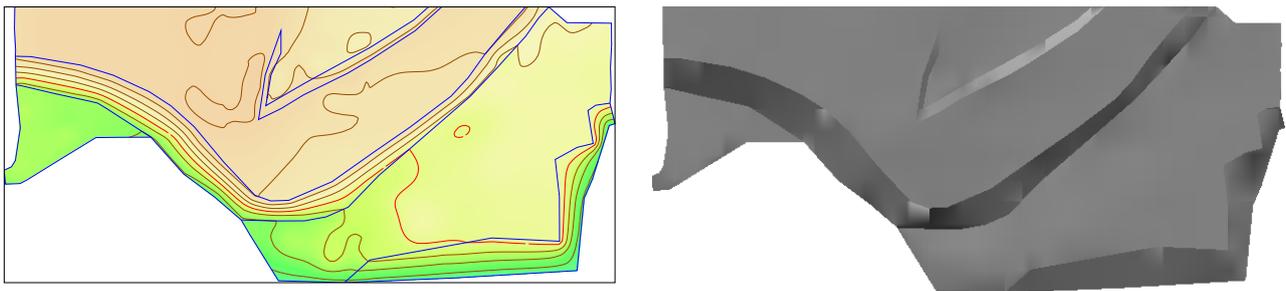


Figure 6.4: Left a z-coding and the contour lines of the terrain model are shown, right a shading of the surface can be seen. The model was derived with kriging and stored in a hybrid grid which contains the breaklines.

## 6.2 Data set “Elev”

This data<sup>2</sup> was obtained by measurements in the terrain. It includes breaklines and can be parameterized over the ground plan. As it is shown in Fig. 6.3, the distribution of data is inhomogeneous. There are many flat surface parts. This kind of data is very well suited for an approach, where the  $z$ -values are a function of the  $xy$ -co-ordinates. In a hybrid grid model this kind of terrain can be represented very well. A qualified gridding technique (e.g. kriging) allows the elimination of random measurement errors. For larger areas (i.e. more points) the solution of global equation systems has to be circumvented. The images in Fig. 6.4 have been produced with SCOP++ ([Dorffner et al., 1999]) applying the described method. As it can be seen, the surface is very smooth.

In Fig. 6.5 a terrain model to the same input data computed by subdivision is shown. After the triangulation of the data, the random errors have been filtered with kriging. In each point the neighbors up to the second generation were used for this filtering, parameterized over their adjusting plane. The interpolating/approximating subdivision algorithm based on paraboloids was applied, the surface is shown after four subdivision steps.

<sup>2</sup>by courtesy of INPHO GmbH. Stuttgart

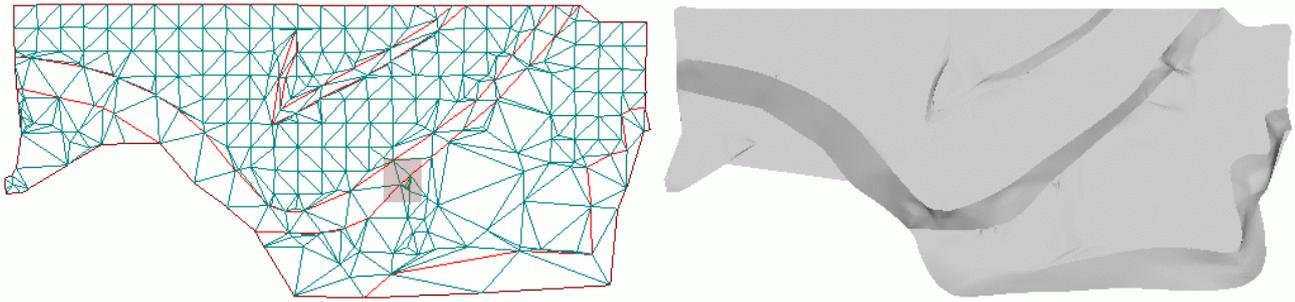


Figure 6.5: Terrain model computed with a subdivision technique based on the averaging of local approximating surfaces. Left, the original triangulation is shown. The marked area is shown in an enlargement in Fig. 6.6.

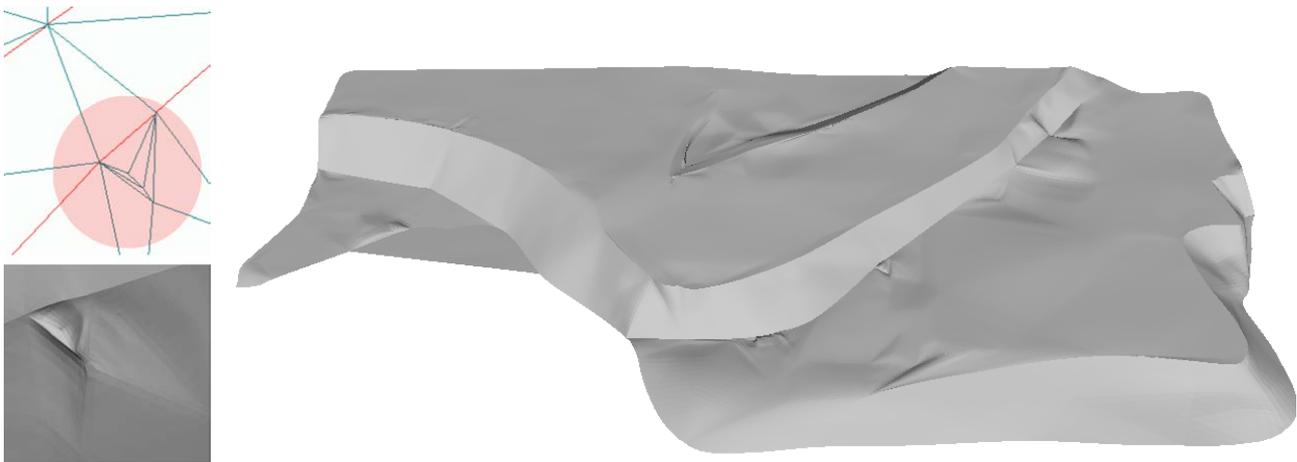


Figure 6.6: Perspective view of the subdivision surface. Left a part of the triangulation is shown, where one point has only valency three. This causes also a fault in the computed surface.

The breaklines were considered as described in Sec. 5.3.2, in a breakline point two local surfaces were computed, one for either side. Along the breakline the Four point interpolatory subdivision scheme was used (Appx. C.2). In the right part of the surface a fault can be seen (cf. Fig. 6.6). It is caused by a point which has only valency three<sup>3</sup> in the triangulation, and three aligned points which lie very close to each other. For a nicer surface the triangulation would have to be free of such artifacts (Sec. 3.5, mesh improvement). Overall, the shape of the surface is satisfying.

In such a ‘simple’ case, the ‘traditional’ 2.5D approach – which has continuously been improved over years – provides a solution which is at least as good as the subdivision approach.

---

<sup>3</sup>i.e. three emanating edges

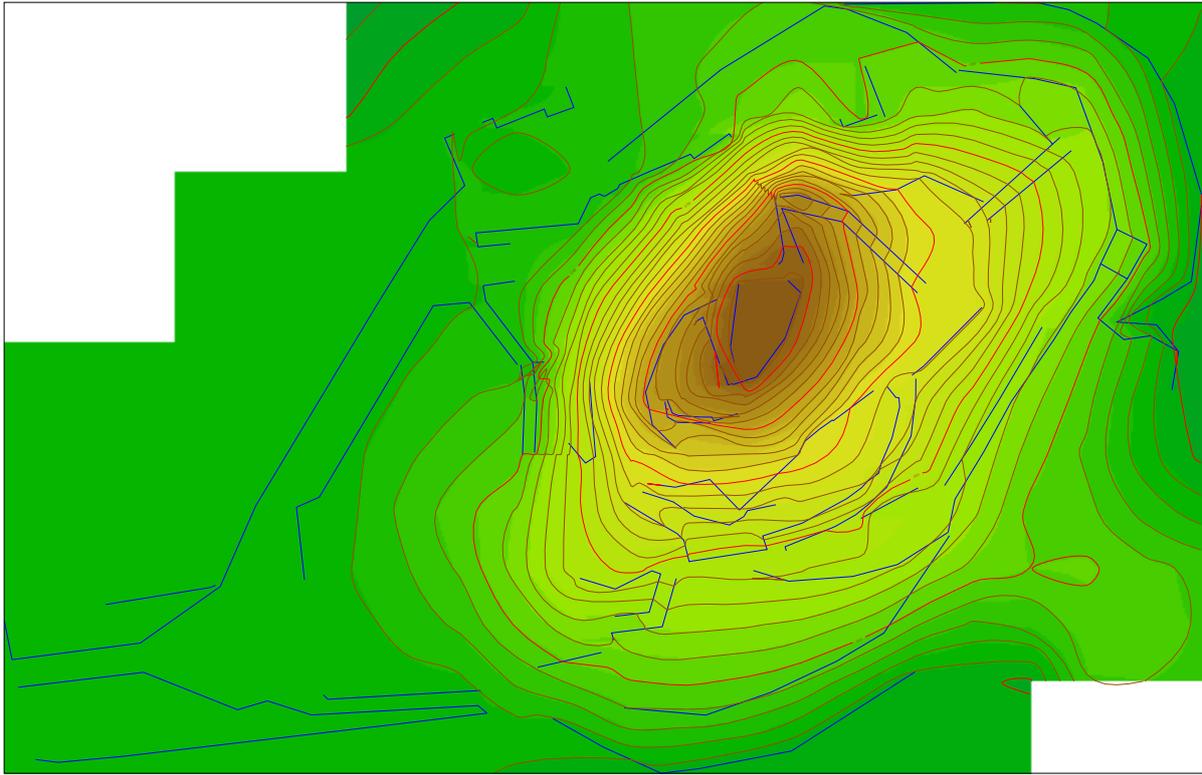


Figure 6.7: Terrain model computed only from breaklines. The contour lines and the z-coding were computed in SCOP++.

### 6.3 Breaklines only

This example<sup>4</sup> contains only breaklines, it is a real world example, too. In each of the given points two or more different slopes meet. Fig. 6.7 shows the measured data as dark lines. Additionally, the contour lines and a z-coding, both computed with SCOP++, are overlaid in the figure. A perspective view of this data is shown in Fig. 6.8.

This can be compared with the model computed with the subdivision approach which has also been used for the “Elev” example (averaging of paraboloids in each vertex). The breakline points were not filtered before the subdivision. Between the breaklines there is only moderate curvature of the terrain, only in some areas strong curvature without breaklines can be found. The final subdivided surface is shown in Fig. 6.9. If no roughness detection (Sec. 5.3.7) is performed, the computed surface has self intersections as shown in Fig. 6.11, left. Note, that self intersections generally can occur with any 3D surface reconstruction techniques, even if the triangulation itself has no self intersections. The roughness detection was performed in a very simple way in this case. If the subdivision points are too far away from the edge midpoint (in this case a quarter of the edge length), then the edge midpoint itself is used instead of the averaged point. As mentioned in Sec. 5.3.7, there are more clever techniques than this one.

In both examples some effects can be seen which cannot be attributed to the breaklines alone. In the left background of Fig. 6.8, the 2.5D approach, an unnatural edge which crosses the given breaklines can be seen. This is caused by the ‘localization’ of the global approach<sup>5</sup>. Additionally, the surface has some bumps which cannot

<sup>4</sup>by courtesy of Vermessungsbüro Polly, Neunkirchen–Wr. Neustadt–Gloggnitz (surveying bureau Polly)

<sup>5</sup>In SCOP terms this is the border of a computing unit.

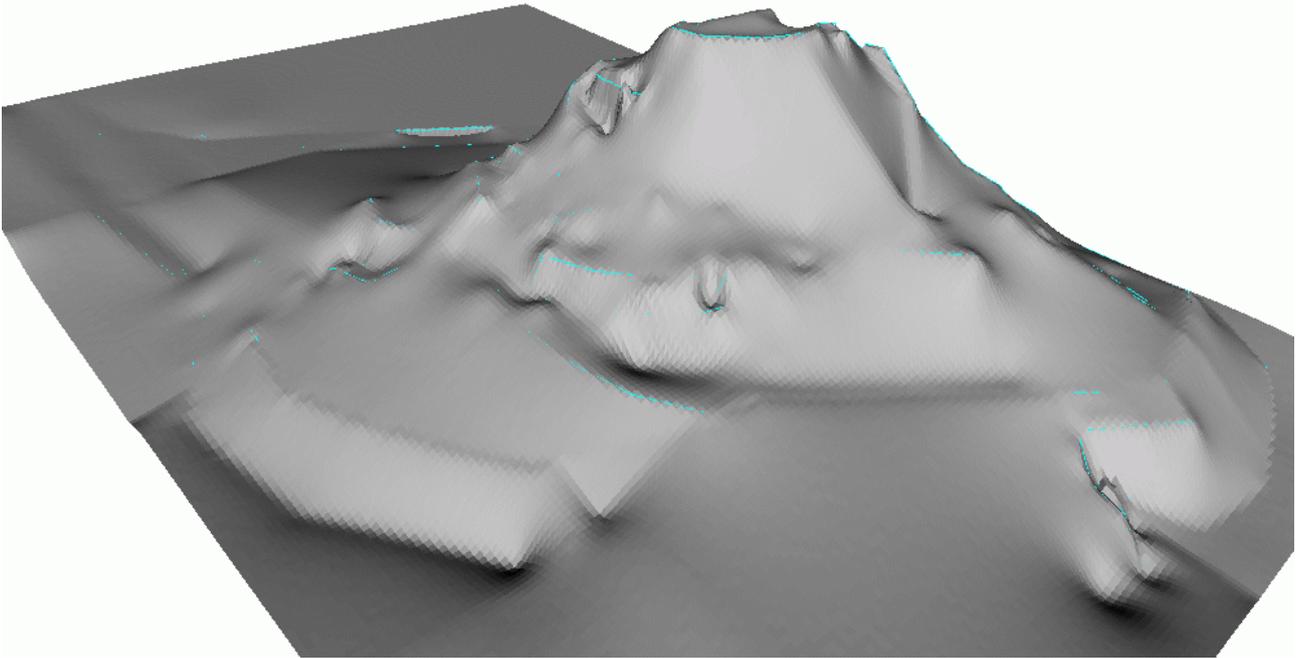


Figure 6.8: Terrain model from Fig. 6.7 in a perspective view. Because the hybrid grid model was exported as a pure grid model the breaklines are blurred. In order to improve the visual impression and the orientation, the original breaklines are drawn, too.

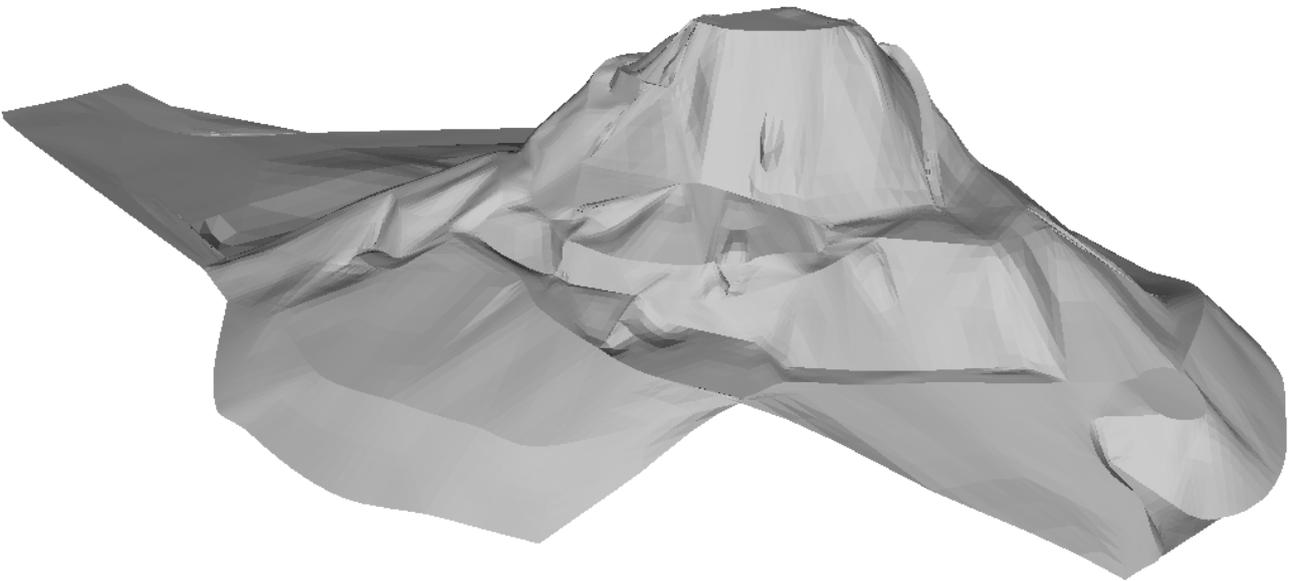


Figure 6.9: Terrain model only from breaklines with a subdivision approach. A criterion for detecting rough areas was applied, and tension was introduced in these areas automatically.

be found in the given data. This can also be seen in the contour line plot of Fig. 6.7. Unwanted oscillations can also be found in the surface of Fig. 6.9, the subdivision approach with the averaging of local surfaces. Stronger tension during the averaging of the local surface (roughness detection) would be necessary in order to avoid those. The problem is, that in some other areas the smoothness can be decreased by such a procedure.

This data was also used to compute a different model. The breakline points were considered to be ordinary

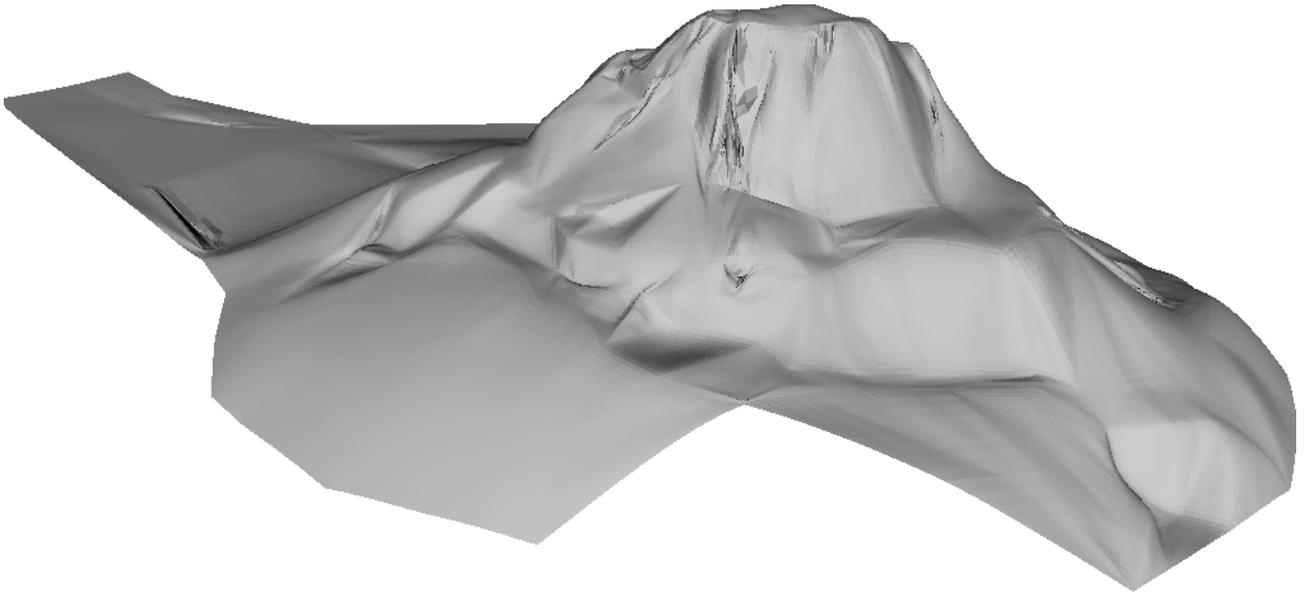


Figure 6.10: The “Breaklines only” example, but the breakline points are considered as normal surface points.

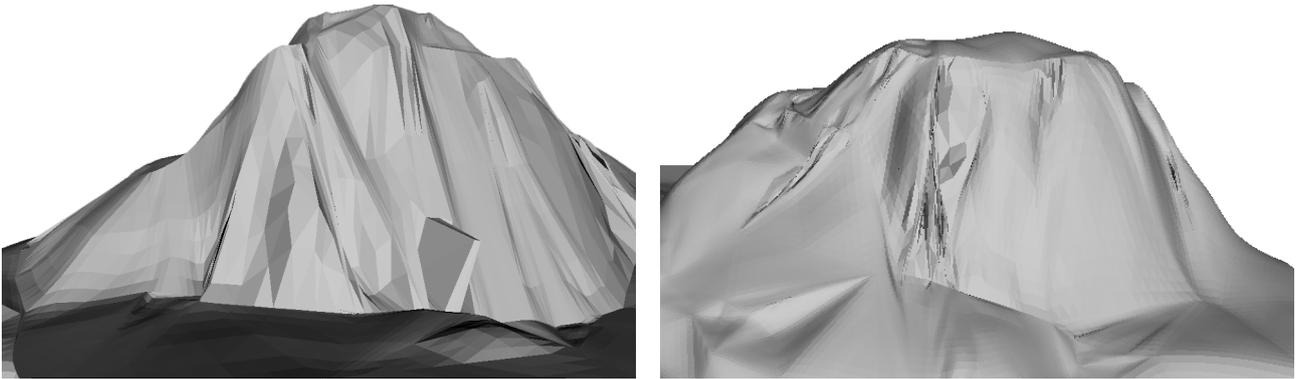


Figure 6.11: Without an automatic detection of the roughness of the terrain self intersections can occur. Left: surface computed from the breaklines with the same approach as for Fig. 6.9, but without roughness detection. The backside of the mountain (compared to the other figures) is shown. Right: surface computed with breakline points considered as normal points and inappropriate roughness detection. This is a detail of Fig. 6.10.

terrain points. The same triangulation as above was used but the breakline feature was removed from the affected edges. Therefore, the model should be smooth everywhere. The surface with the above subdivision approach is shown in Fig. 6.10. In most areas the smoothness is achieved as desired, though the original triangulation is partly still visible in the final surface. One slope down from the top plateau of the terrain has many self intersections as shown in Fig. 6.11, right.

The subdivision approach is applicable for this kind of data, but tension has to be used in order to avoid self intersecting surfaces which do not occur in natural terrain.

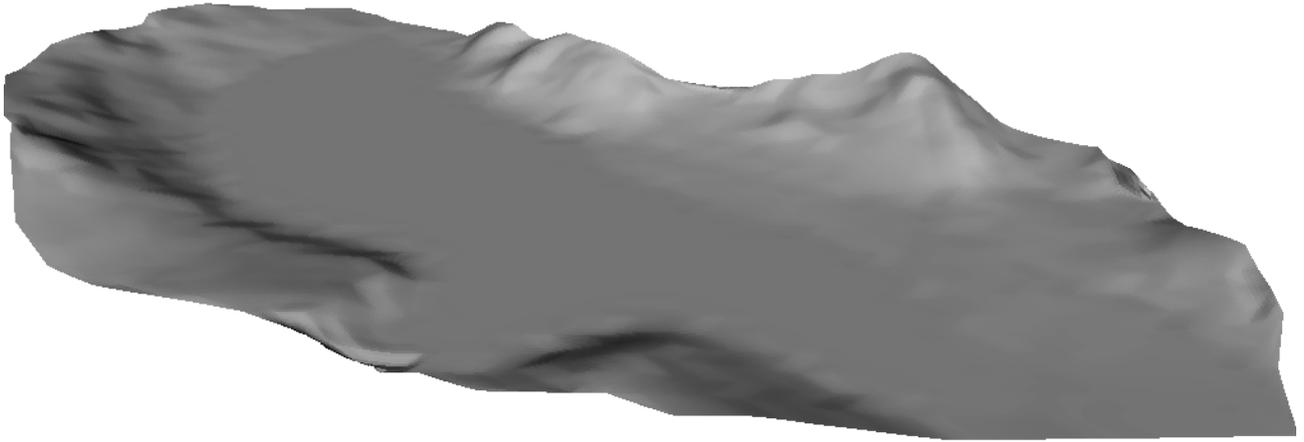


Figure 6.12: Gently sloping terrain surface reconstructed with an interpolating/approximating subdivision approach based on paraboloids as local surfaces.

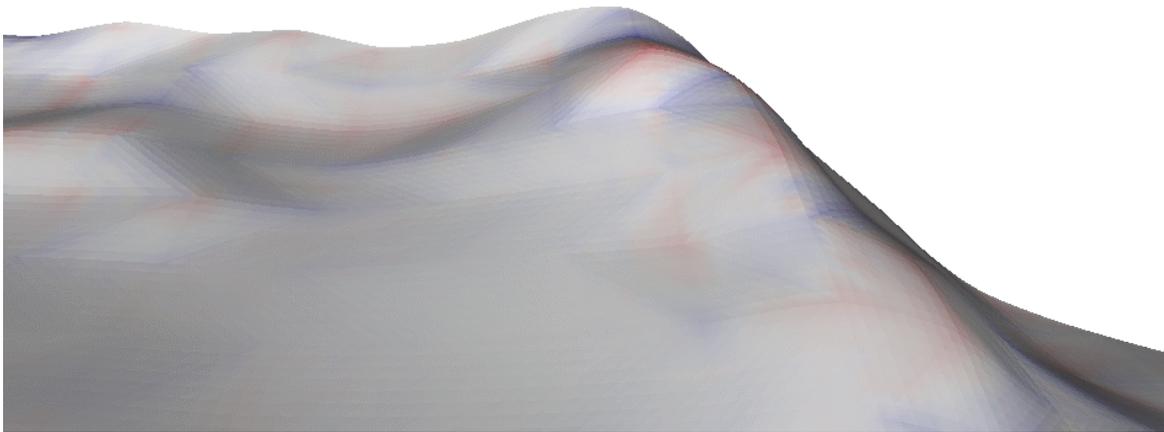


Figure 6.13: Detail of the "Albis" surface computed with a subdivision approach. Red and blue areas stand for positive and negative, respectively, gaussian curvature. A similar perspective on the same terrain is also shown in Fig. 5.7. This figure is also shown in color as fig. E.5 on page 125.

## 6.4 Data set "Albis"

The data of this example<sup>6</sup> is given in a regular grid in the ground projection. It is gently sloping terrain which has been measured by photogrammetry methods. It is obvious that a surface reconstruction method like kriging will produce a very nice result in this case, but as it can be seen in Fig. 6.12 also the subdivision approach with the local surfaces is very suitable for the reconstruction of such a surface.

In Fig. 6.13 the ridge on the right side in the back is shown in an enlargement. Red are those areas with positive gaussian curvature (elliptic points) and blue areas have negative gaussian curvature (hyperbolic points). A stronger color indicates a higher curvature. In this figure the original triangulation can still be seen and it also shows that at some triangles the curvature is concentrated along the borders.

---

<sup>6</sup>by courtesy of 'swisstopo' (Swiss Federal Office of Topography)

## 6.5 Bridge

This example is a synthetic example, again. It shows a bridge which stretches over a valley. All together 20 points and 32 triangles describe the terrain and man-made features in this case. This example is included in order to demonstrate the ability of the presented approaches to model terrain which is topologically not equivalent to a sphere but has higher genus, in this case genus one. A parameterization in the  $uv$ -domain over an open disc is not possible any more<sup>7</sup>. Thus only approaches over triangulations are presented here. In Fig. 6.14 a wire frame model and a face model are shown.

The given data has been subdivided with different subdivision approaches. In Fig. 6.15 the surfaces obtained with the Loop and the Modified Butterfly scheme are shown. It can be seen that the Loop subdivision, but also the Butterfly subdivision make the bridge thinner. The left surface runs high above the given points. Of course, in a real world example the sampling of points on the terrain would be denser, especially if measured by photogrammetric means and not by tacheometry. The right surface has unmotivated bumps.

In Fig. 6.16 the surfaces are shown after their sixth refinement with subdivision based on the local approximating surfaces. In both cases an interpolating/approximating approach is used. In the left example the surfaces are vector-valued second order polynomial and the neighborhood for their estimation is one and a half generations. On the right side, the surface is subdivided with paraboloids as local surfaces. The neighborhood reaches only to the first generation which explains why the valley is not dragged upwards. Additionally, three closed breaklines were introduced in this example, each consisting of four edges. Two can be found on either side of the bridge and one circumscribes the bottom side of the bridge. The lower edges of the bridge are therefore marked twice as breaklines. Not the Four point interpolatory subdivision scheme for curves was used in this example, but simple edge midpoint subdivision. This is the reason why the edges of the bridge are the same in the triangulation and the subdivision surface. With the Four point interpolatory subdivision the breaklines would not have sharp corners but they would be round. Again, it can be noticed that the subdivision with vector-valued polynomial surfaces has smoother limit surfaces than with the paraboloids, especially if the angles between adjacent triangles are large.

This example demonstrates the importance of correctly modelling straight edges and breaklines, if they are found in the topography.

---

<sup>7</sup>Of course, a torus, which is equivalent to the presented example, can be parameterized over the region  $(u, v) \in [0, 2\pi) \times [0, 2\pi)$ , but the important aspect is that the parameter domain is cyclically continued in itself and therefore not an open disc anymore.

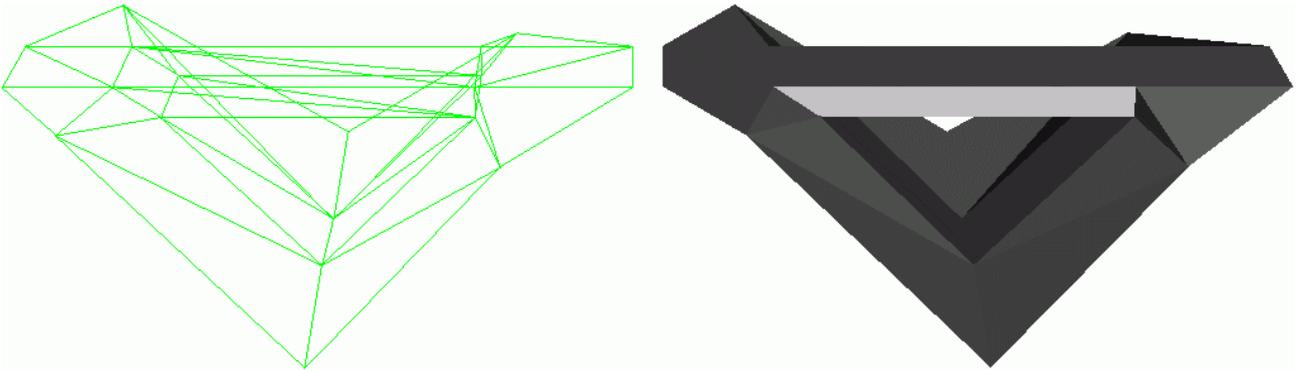


Figure 6.14: Triangulation of a bridge and the valley it spans. Left a wire frame is shown, right the faces can be seen. Note that the data is not completely left-right symmetric.

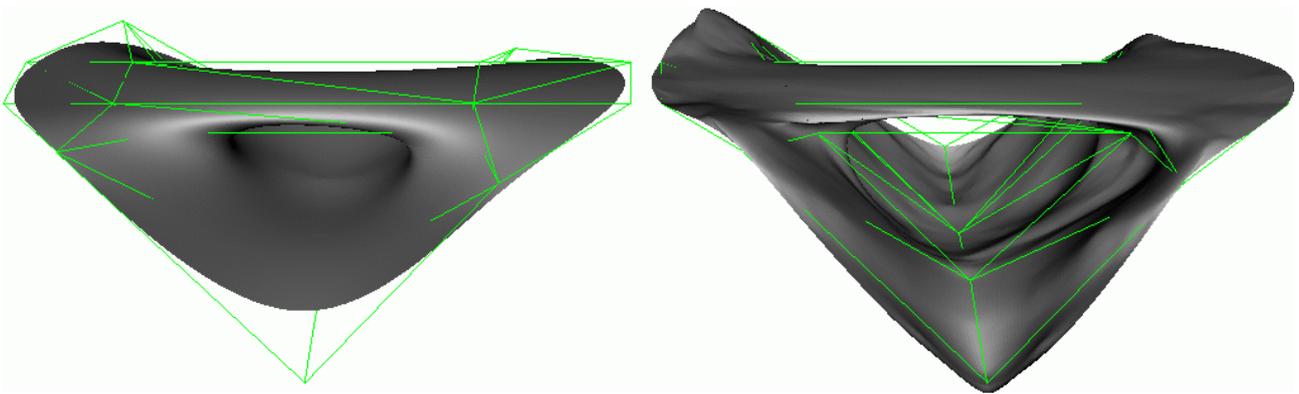


Figure 6.15: The surfaces show the triangulation of the bridge and the valley after six refinement steps with the Loop scheme (left) and the Modified Butterfly scheme (right).

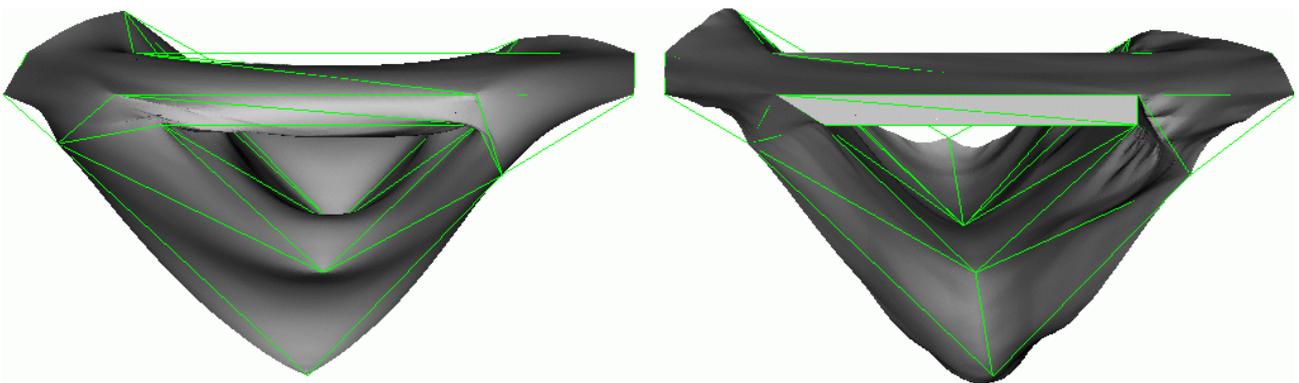


Figure 6.16: Left, the subdivision surface with vector-valued second order polynomials as local surfaces is shown. Right, the subdivision surface with paraboloids as local surfaces is shown, additionally, breaklines on the bridge were included.



## Chapter 7

# Conclusions and Perspectives

In this work, many different modelling techniques for the reconstruction of 3D terrain<sup>1</sup> have been presented. Obviously, they all have their advantages and drawbacks, but to some extent they aim at different goals. Smoothness, fairness and computation time are only three examples for such (dis)advantages and goals. However, terrain measurements are not used to reconstruct a model for itself, but in order to apply this model. Eventually, these applications should be used to define the required model properties. In Chap. 2, for example, the significance of smoothness for terrain models was pointed out. The decision for certain model properties also has an influence on the appropriateness of certain reconstruction techniques. However, the applications in terrain modelling are very versatile which suggests to apply a modelling technique which suits most needs as good as possible.

In a 3D model the well known applications for terrain models have some new properties. Though the main subject of this work is the reconstruction technique, it is worthy to take a glance at those applications under the perspective of a 3D terrain model (Sec. 7.1). In another section an enclave technique is described which allows to use 2.5D approaches in most areas and 3D approaches in the ‘real 3D’ parts.

Above all, algorithms for the reduction of data to a certain size or within a certain tolerance or mean error are necessary, too. This is not so much necessary because of methodological reasons, but for interfacing to other terrain model application programs. If the data is held simultaneously at different levels of resolution, each application can be solved on an appropriate level, not necessarily the finest one.

### 7.1 Applications

In this work several approaches have been presented for the definition of a topographic surface in 3D space without the restriction that the surface is the graph of a bivariate function. The patchwork method and the subdivision method have been presented in more detail and one method has been proposed for each type which fulfills the requirements in topographic surface modelling. These approaches require that the measured points and lines are triangulated. Another method is also based on a triangulation, but a parameterization of the mesh is computed first. Next, a vector-valued function is computed over the parameterization. It cannot be said, that one method is superior to all other methods. In the 2.5D case there is a long ongoing argument whether triangulations of the measured points are to be favored to grid models or vice versa. For the 3D case a triangulation appears – to some extent – inevitable, but this does not prescribe a certain surface reconstruction technique. The aspect of the triangulation common to all 3D modelling techniques is to determine the neighborhood relations. In the 2.5D case the neighborhood relations are much simpler, and thus, a triangulation is not always necessary. Other techniques, like quad-trees and other tessellations of the plane, can be applied, too. The triangulation is only one possible tessellation.

---

<sup>1</sup>Ordinary, commercially available DTM packages, are not completely 3D, because they do not allow the modelling of overhangs and caves. To point out these limitations, those approaches are often called 2.5D terrain models.

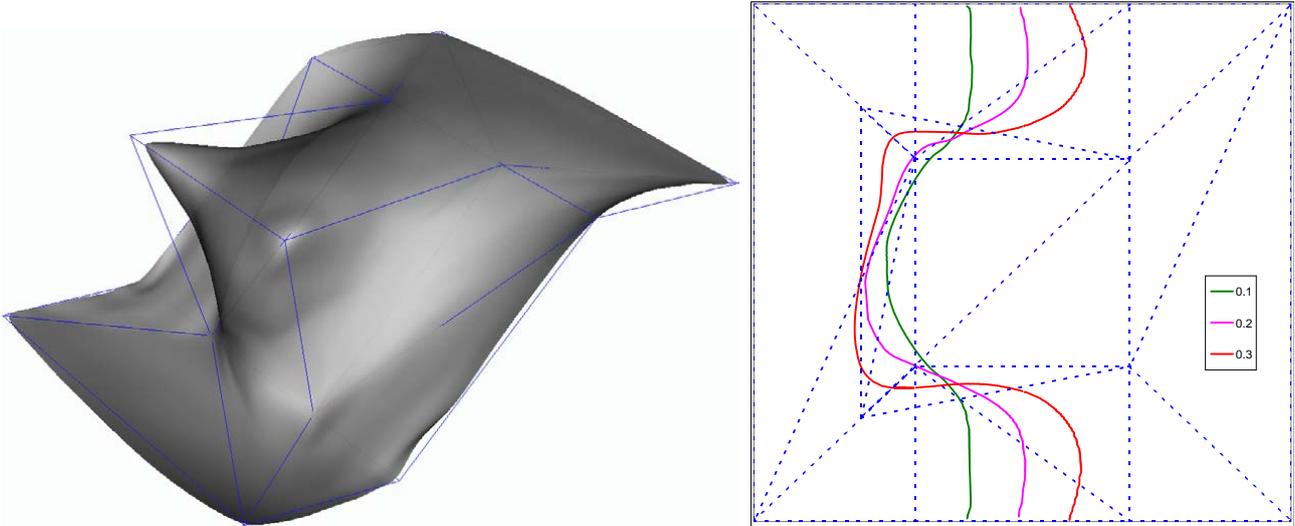


Figure 7.1: Perspective view of an overhang, the given points are in the  $z$ -range  $[0.0,0.4]$ . To the right the contour lines for the levels 0.1, 0.2, and 0.3 are shown. The surface is computed over the triangulation with a subdivision approach. The first three refinements are performed with the interpolating/approximating local surfaces (vector-valued) as described in Sec. 5.3.2, the following ones with the Loop scheme.

The definition of the existing applications in the new 3D frame and the development of new applications will be at least as important as the modelling technique itself. In the following, a short list will be given on how some of the existing applications of elevation models are defined in the new environment. This list does not claim to cover all possible applications of topographic surface models.

**Contour lines** are the intersections of the surface with planes with the common normal vector pointing to the zenith. In the 2.5D case these intersections tessellated the ground plan (i.e. the parameter domain) into adjacent regions of height intervals. In the 3D case this is no longer valid. The contour lines can have intersections, if they are projected onto the plane  $z = 0$  (Fig. 7.1). Additionally, contour lines alone provide less possibilities to reconstruct the surface than in the 2.5D case.

**Shading,  $z$ -coding and slope model** are terrain models where every surface point has an additional property attached: a grey value in the case of a shading (as shown in the previous chapters), which is determined by the aspect of the surface and the position of an imaginary sun, a color value which resembles the height and the gradient vector, respectively. In the 2.5D case this property could be viewed in the ground plan (the parameter domain). This is no longer possible in the 3D case. The property remains attached to the surface point or – if a global parameterization has been computed – to the parameter position in an  $(u, v)$ -plane.

If a projection into  $z = 0$  is required, certain surface parts have to be favored to others. One possibility is to take always the upper most surface, the part which is visible from a position above the highest surface point. Alternatively, the lowest could be taken or – after some methods which remain to be defined – a medium layer.

**Volume computations** are possible like before. The terrain is considered as the boundary of a solid body. A lower surface (usually  $z = \text{const}$ ) and an area (i.e. a curve in this lower surface) are defined. This curve defines a generalized cylinder with a certain direction of the generators (usually  $(0\ 0\ 1)^T$ ). This cylindrical surface bounds the volume in lateral direction, the lower surface builds the basis and the terrain surface closes the volume. The volume of the so defined body can be computed.

This can also be seen as the volume of a solid which is the intersection of two other solids. The terrain model is extended downwards (i.e. to the center of the earth) in order to generate the first solid. The other solid must define the ‘lower’ bound.

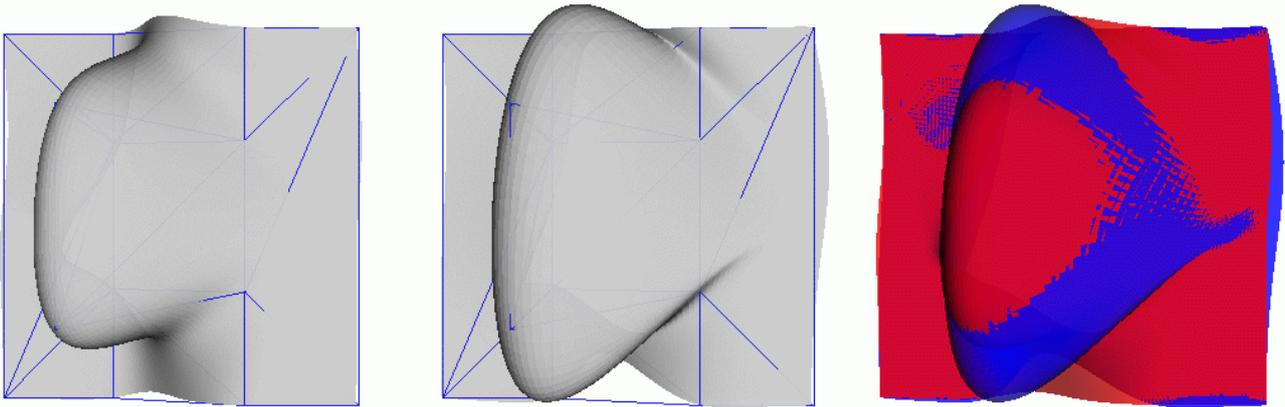


Figure 7.2: Left and middle: two different models describing an overhang, viewed from above. Right, the two models are superimposed. How to define the difference between those two models? The two models have been obtained by computing a vector-valued surface, stored in a grid model, over the parameterizations of the overhang triangulation shown in Fig. 3.4. This figure is also shown in color as fig. E.7 on page 126.

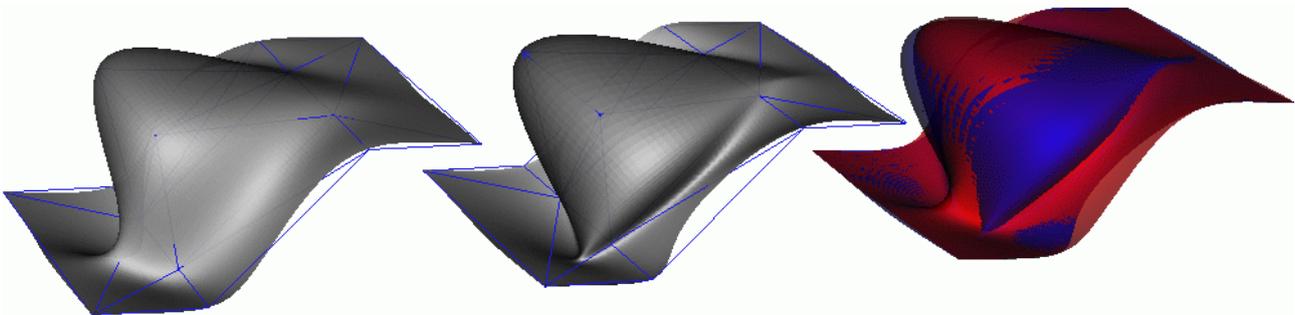


Figure 7.3: A different view on the data shown in Fig. 7.2. This figure is also shown in color as fig. E.8 on page 126.

**Difference models** are only defined, if both models have a common parameterization. In a difference model it is stored *where* in a *first model* and *how much* has been added to form a *second model*. In a 2.5D environment the common parameterization is always given. If two triangulated surfaces (possibly with parametric patches or refined with subdivision) are given which are sufficiently similar, it will also be possible to compute a difference model based on distances from one model to the other. If it is possible to compute the difference, then the orthogonal distance becomes a surface property of one of the models. The crucial question is whether it is possible to find a one-to-one correspondence between the points of the two surface (see Figs. 7.2 and 7.3).

If, on the other hand, the difference is computed over a parameterization in the  $(u, v)$ -plane some homologous points will have to be defined on both surfaces which map to the same point in parameter space. However, the difference vectors (difference in  $x$ -,  $y$ - and  $z$ -direction) will in general not be perpendicular to either surface.

**Perspective views** are well defined. Also the draping of texture (from a photograph or a satellite scene) over the model is well defined. Of course, the visibility analysis is more complicated than in the 2.5D case. For visualization purposes the quality of the model may be lower. If the viewer is prevented from zooming in too strong and if the point density is sufficiently high, then a ‘water proof’ model (i.e. a model without holes, caused e.g. by erroneous triangulations) is not necessary for visualization purposes.

**Local horizons** are only defined for points with an outward surface normal vector with positive  $z$ -value, otherwise the horizon can rise beyond the zenith. A different definition, based on the complete sphere and

not only on the northern hemisphere (zenith directed hemisphere) will be necessary. A new definition of a 'local local' horizon, based in a point and computed relative to its tangent plane is possible, too.

**Water shed analysis** differs from the 2.5D case by the fact that the water does not always flow to a neighboring point (area element, ...) but it can also drip (or fall, like in a cascade) from one surface point to a remote surface point which has the same planimetric co-ordinates but a lower height.

**Profiles** are well defined, too, but they cannot be described as graphs of univariate functions. A profile through the terrain in a vertical plane will be a general planar space curve. However, the intersection of the terrain with this plane can yield more than one curve, even if the genus of the surface is zero.

**Conversion to 2.5D model** but also simplification of the model (e.g. removal of all overhangs, filling of all caves, removal of all handles, so that the surface has genus zero) are additional applications. After conversion to 2.5D all the well-known applications can be performed in their present state..

**True ortho photos** are special cases of perspective views. The viewing rays become parallel and the projection plane is  $z = 0$ . The accuracy requirements refer to point accuracy as well as to topological correctness. For small scale applications, the modelling of smaller bridges may be omitted.

A so-called photo(realistic) model of an object (e.g. a building) is a CAD model of this object with the texture from photos draped over it. In a 3D terrain model the difference between ortho photo and photo model is abolished.

**Measurement** of distances between two surface points is no longer well defined, if only the ground plan co-ordinates are provided. To avoid ambiguities all three co-ordinates of the points must be specified.

## 7.2 Enclaves

The introduction to this work included a statement like: "...in a 3D surface model no co-ordinate direction must be favored...". Hill tops are modelled with horizontal tangent planes, but this is merely one special case of the general prescription of two linear independent tangential vectors in a surface point. Looking at the real terrain, this statement has to be relativized. Of course, there are overhangs, and there are almost vertical walls which are poorly represented in a 2.5D modelling approach, and finally, if houses shall be modelled in the same data structure, there are vertical walls<sup>2</sup>. The total area of these features is, however, small compared to the area which can be modelled as a scalar-valued bivariate function (2.5D approach), at least in natural regions. Furthermore, in a 3D approach unwanted 'micro overhangs' can occur, either because of the triangulation algorithm, or due to the surface computation. They originate in unfavorable data distributions and random measurement errors. Obviously, such errors cannot occur with 2.5D modelling techniques. This suggests to apply a vector-valued approach only there where it is necessary, also within one terrain model.

This is the so-called enclave technique. It means, that different model types are used for different parts of the terrain. For some algorithms and applications (e.g. ortho photo and contour line computation, perspective views with z-buffering) the mixing of results, derived from the 'different models within one terrain' is simple. The results are independent from each other, also along the borders of different regions. This is valid, if each area can be handled independently from the others, but other algorithms (e.g. visibility analysis) become more complicated, because the different data structures have to be inter-mixed in one way or the other. However, the enclave technique itself generates problems because the continuity conditions at the boundaries are not easy to handle. To achieve continuity (C0) is - within certain bounds - not difficult, because the problem posed is often an interpolation problem. Interpolating the same data therefore leads to the C0-continuity - at least at the measured points itself. Between the border points the surface definitions could vary. This would lead for example to contour lines which are set apart a small distance along the model borders. Much more complicated is to establish continuity of the first derivatives (i.e. the tangent planes). One possibility is to prescribe first derivatives at the boundary, another one is to demand, that the boundaries are always breaklines. The first

---

<sup>2</sup>A more suitable data structure may be to model the houses independently of the terrain in order to have house objects which can be manipulated independently of the terrain.

approach fits nicely into the patch-work technique. In general, both choices appear unnatural, the continuity should ideally be generated within the process of deriving the models. This would require a mutual derivation of the models.

There is also an economical motivation for the enclave technique. In airborne laser scanning projects, (for example,) only very few areas cannot be modelled by existing 2.5D approaches (either TIN or grid). These projects typically reach extends of many square kilometers (e.g. 100) over which millions of points are measured. Additionally, the economical interest in the real 3D areas (caves, walls) is comparably low. Furthermore, the 3D algorithms are typically more expensive than their 2.5D versions. Often, the strategy is to forget about the 3D-areas and find some provisional solutions with much manual interactive work on a trial-and-error basis for each problem. The solutions are then tailored to each problem, and in general this solution is very unsatisfactory. However, most of the area should be processed with a 2.5D approach.

## DTM derivation enclaves

In this paragraph an enclave technique will be described for the derivation of a terrain model. The complete terrain is parameterized over the  $uv$ -domain. In 'simple' areas the surface is described as  $x = u, y = v, z = f(u, v)$ , whereas 'real' 3D areas require a description of the type  $x(u, v), y(u, v), z(u, v)$ . For these areas a parameterization has to be computed, methods are specified in Sec. 3.2.3. The borders of these areas have to reach into the 'simple' terrain areas, and if their ground projection is convex, then this ground projection can be used as the border of the parameterization. This guarantees that the referenced parameterization methods have a solution. In order to achieve continuity (C0, C1, ...) an overlap between 'simple' and 'real 3D' areas is necessary.

Among the 3D surface reconstruction methods there are local methods for which a linear parameterization of the points ( $u = x, v = y$ ) results in a linear parameterized surface ( $x = x(u, v) = u, y = y(u, v) = v, z = f(u, v)$ ). This behavior can be found e.g. for affine invariant polynomial patches. If such a surface reconstruction is used in the 3D areas and its scalar-valued counter part in the simple areas, and if the overlap is sufficiently large, then exact smoothness can be achieved. Otherwise, the smoothness is obtained only approximately, and weighting functions may be necessary for the transition of one surface description into the other. One universal grid structure may be used to represent the complete surface. This would be a 3D grid defined over the  $uv$ -domain.

## DTM application enclaves

Differing from the DTM derivation enclaves which had alternatives in the model computation, the DTM application enclaves which will be described now have alternatives in the computation of the DTM products (contour lines, ...). For mere completeness, the whole process (i.e. DTM derivation first and then the applications) is described in the following.

1. Triangulate the data in 3D. If available, additional information on the measurement technique (e.g. line of sight method, ...) can be used to simplify the problem and speed up the computation. More general algorithms have also been presented in Sec. 2.3.

Filter the data in order to eliminate random measurement errors and establish the required smoothness of the terrain surface. For this end, information on the measurement process is necessary, too. This filtering should be applied locally, taking a sufficiently large neighborhood into account. This also requires knowledge on the type of terrain. Often, especially if the neighborhood relations have already been derived, this can be determined from the data itself. One example for this is the estimation of the covariance function of the surface as given in [Kraus, 2000].

2. Apply methods to generate tangent plane continuity across the edges if this is necessary (chapters 4 and 5). These methods have to work in 3D in order to avoid boundary problems. The necessity of tangent plane continuity is a question of the application and the roughness of the existing terrain triangulation

(angles between triangle normals). Because a terrain model is used and often re-used for many different applications, generating a smooth surface should always be considered. Applying subdivision has the advantage of keeping the data structure simple and allows interfacing to different programs without model conversion.

3. Find out, which areas can be treated as 2.5D areas and which areas are completely 3D. Again, a 2.5D area is such an area, where only one height  $z$  is defined for a position in the ground plan  $(x, y)$ . This is an information which can be stored for each triangle, possibly as one flag in the data structure. Of course, marking more triangles as 3D does not generate wrong results, but the opposite may do so. This solution would still take the special property of the third co-ordinate, it is (more or less) the direction of gravity which is essential for the forming of the terrain, into consideration.

No algorithms for this task have been presented in this work. Depending on the data structure, simple volumetric algorithms – which analyze the points in big voxels or octrees – could provide a fast solution. This algorithm would also be necessary for the detection of 2.5D and 3D areas for the DTM derivation enclaves.

4. Apply 2.5D algorithms in those areas where no triangle is marked as belonging to a 3D area. This provides fast routines for those areas, where no overhangs and caves exist. A perspective view can for example be generated by drawing from the background to the foreground, if it is known that the data is only 2.5D. Water shed analysis is easier likewise, because the water always flows to a neighboring triangle, if the data is only 2.5D.

Of course, this approach is feasible only, if large connected areas with the 2.5D flag set exist. It is needless to say that this approach cannot be applied to triangulations alone, but also to e.g. grid models. Furthermore, the two enclave techniques can be applied together. It is clear that the enclave technique also has a substantial drawback, the processing becomes more complicated, because some distinction between complex and simple terrain is necessary in different situations.

### 7.3 Concluding remark

As mentioned at the beginning of this chapter, several techniques with different advantages and drawbacks have been presented.

One of the methods was to compute a parameterization for the complete area and determine a vector-valued surface by some of the well-known methods (e.g. kriging) over the parameter domain. The result is best stored in a hybrid grid which guarantees the correct treatment of breaklines and special surface points like peaks. While in all the examples this approach produced the nicest surfaces, there are two main disadvantages. First, a global equation system has to be solved in order to compute the parameterization, at least with the referenced methods. Second, these parameterization methods work only for surfaces which are of genus zero.

The patch work approach is another method to compute a surface over a given triangulation which may also be of arbitrary topological type. With different parametric patches it is possible to compute a smooth surface. A disadvantage of this method is that the patches themselves are polynomials of higher order (three upwards), some of those patches are rational polynomials, which makes algorithms applied to these patches more expensive. The developed approach is suitable for topographic terrain modelling. Not in all cases the desired smoothness is obtained right away which makes a splitting of patches necessary.

The subdivision method does not have the disadvantages listed above. The limit surfaces of the subdivision approaches are smooth but after any subdivision step the surface itself is composed of triangles. For triangular networks many algorithms do already exist (e.g. mesh thinning, ...). The combination of subdivision based on local approximating surfaces for the first few subdivision steps and application of a refinement with guaranteed smoothness for the following steps provided the best results. Additionally, the consideration of the requirements in topographic surface modelling is possible. This method is therefore proposed for the reconstruction of general purpose 3D terrain models. For real data an improvement of the triangulation (i.e. elimination of very close points, ...) and the filtering of the random measurement errors is necessary before the application of the subdivision algorithm.

## **Last sentence**

As pointed out, the third dimension plays an important role in the modelling of the terrain. However, as author of this thesis I should not swing into airy three dimensional heights but come down to an earth surface which is mainly (treated) two and a half dimensional and acknowledge that an operable and truly three dimensional terrain model for photogrammetric and topographic purposes has not become readily available yet.



# Bibliography

- [Amenta et al., 1998] Amenta, N., Bern, M., and Kamvysselis, M. (1998). A new Voronoi-based surface reconstruction algorithm. In *Siggraph '98 Conference Proceedings*, pages 415–421.
- [Bajaj, 1990] Bajaj, C. (1990). Surface fitting using implicit algebraic surface patches. Technical Report CSD-TR-1001, Purdue University, Department of Computer Sciences.
- [Benkő et al., 2001] Benkő, P., Martin, R. R., and Várady, T. (2001). Algorithms for reverse engineering boundary representation models. *Computer Aided Design*, 33(11):839–851.
- [Bern et al., 1993] Bern, M., Edelsbrunner, H., Eppstein, D., Mitchell, S., and Tan, T. S. (1993). Edge insertion for optimal triangulations. *Discrete & Computational Geometry*, 10:47–65.
- [Beyer, 2000] Beyer, G. (2000). A grid-based edge model for the wavelet transformation of hybrid DTM. In *International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, B4*, pages 39–42, Amsterdam, Netherlands.
- [Bol, 1967] Bol, G. (1950, 1954, 1967). *Projektive Differentialgeometrie I, II, III*. Vandenhoeck & Ruprecht, Göttingen.
- [Borgefors, 1986] Borgefors, G. (1986). Distance transformation in digital images. *Computer Vision, Graphics and Image Processing*, 34(3):344–371.
- [Botsch, 1999] Botsch, M. (1999). 3-D Gesichtsmodellierung zur Operationsplanung. Diplomarbeit, Universität Erlanger-Nürnberg.
- [Briese et al., 2001] Briese, C., Belada, P., and Pfeifer, N. (2001). Digitale Geländemodelle im Stadtgebiet aus Laser-Sanner-Daten. *VGI, Österreichische Zeitschrift für Vermessung & Geoinformation*, 89(2):83–91.
- [Carrara et al., 1997] Carrara, A., Bitelli, G., and Carla, R. (1997). Comparison of techniques for generating digital terrain models from contour lines. *International Journal of Geographical Information Science*, 11:451–473.
- [Chen et al., 1994a] Chen, X., Ikeda, K., Yamakita, K., and Nasu, M. (1994a). Raster algorithms for generating delauney tetrahedral tessellation. In *International Archives of Photogrammetry and Remote Sensing, Vol. XXX, 3/1*, pages 124–131, Munich, Germany.
- [Chen et al., 1994b] Chen, X., Ikeda, K., Yamakita, K., and Nasu, M. (1994b). Three-dimensional modeling of gis based on delauney tetrahedral tessellations. In *International Archives of Photogrammetry and Remote Sensing, Vol. XXX, 3/1*, pages 132–139, Munich, Germany.
- [Choi et al., 1988] Choi, B. K., Shin, H. Y., Yoon, Y. I., and Lee, J. W. (1988). Triangulation of scattered data in 3d space. *Computer Aided Design*, 20(5):239–248.
- [Clark Labs, 2002] Clark Labs (2002). [www.clarklabs.org/](http://www.clarklabs.org/). Homepage of Idrisi.
- [Crippa et al., 1998] Crippa, B., Maliverni, E. S., and Tucci, G. (1998). Complex surface representations. In *International Archives of Photogrammetry and Remote Sensing, Vol. XXXII, 6/W4*, pages 125–133, Perugia, Italy.

- [de Boor et al., 1987] de Boor, C., Höllig, K., and Sabin, M. (1987). High accuracy geometric Hermite interpolation. *Computer Aided Geometric Design*, 5:269–278.
- [DeRose and Mann, 1992] DeRose, T. and Mann, S. (1992). An approximately  $G^1$  cubic surface interpolant. In Lyche, T. and Schumaker, L., editors, *Mathematical Methods in Computer Aided Geometric Design II*, pages 185–196. Academic Press.
- [DeRose, 1990] DeRose, T. D. (1990). Necessary and Sufficient Conditions for Tangent Plane Continuity of Bézier Surfaces. *Computer Aided Geometric Design*, 7:165–179.
- [Desbrun et al., 1999] Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH 99 Conference Proceedings*, pages 317–324.
- [Dickerson et al., 1997] Dickerson, M., Drysdale, R., McElfresh, S., and Welzl, E. (1997). Fast greedy triangulation algorithms. *Computational Geometry: Theory and Applications*, 8:67–86.
- [Dorffner et al., 1999] Dorffner, L., Mandelburger, G., Molnar, L., Wintner, J., and Wöhrer, B. (1999). Geländemodelltechnologien - Forschung und Weiterentwicklung am I.P.F. In *Tagungsband der 10. Internationale Geodätische Woche in Obergurgl*, pages 31–44.
- [Dyn et al., 1987] Dyn, N., Gregory, J., and Levin, D. (1987). A four-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design*, 4:257–268.
- [Dyn et al., 1990] Dyn, N., Levin, D., and Rippa, S. (1990). Data dependent triangulations for piecewise linear interpolation. *IMA Journal of Numerical Analysis*, 10:137–154.
- [Ebner and Reiß, 1978] Ebner, H. and Reiß, P. (1978). Height interpolation by the method of finite elements. *Nachrichten aus dem Karten- und Vermessungswesen*, pages 79–94.
- [Eck et al., 1995] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W. (1995). Multiresolution Analysis of Arbitrary Meshes. In *SIGGRAPH 95 Conference Proceedings*, volume 29.
- [Farin, 1982] Farin, G. (1982). A construction for visual  $C^1$  continuity of polynomial surface patches. *Computer Graphics and Image Processing*, 20:272–282.
- [Farin, 1986] Farin, G. (1986). Triangular Bernstein–Bézier patches. *Computer Aided Geometric Design*, 3:83–128.
- [Farin, 2002] Farin, G. (2002). *Curves and Surfaces for CAD*. Morgan Kaufmann, 5 edition.
- [Floater, 1997] Floater, M. S. (1997). Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14:231–250.
- [Floater and Reimers, 2001] Floater, M. S. and Reimers, M. (2001). Meshless parameterization and surface reconstruction. *Computer Aided Geometric Design*, 18:77–92.
- [Foley and Opitz, 1992] Foley, T. A. and Opitz, K. (1992). Hybrid cubic bézier triangle patches. In Lyche, T. and Schumaker, L. L., editors, *Mathematical Methods in Computer Aided Geometric Design II*, pages 275–286. Academic Press.
- [Forkert et al., 1995] Forkert, G., Halmer, A., Kager, H., and Kraus, K. (1995). Konsequente ausgleichung dreidimensionaler digitalisierter liniennetze. *Zeitschrift für Vermessungswesen*, 120(2):59 – 67.
- [Garland and Heckbert, 1997] Garland, M. and Heckbert, P. S. (1997). Surface simplification using quadric error metrics. In *SIGGRAPH 97 Conference Proceedings*, pages 209–216. <http://www.cs.cmu.edu/garland/quadrics/>.
- [Gonçalves et al., 2002] Gonçalves, G., Julien, P., Riazanoff, S., and Cervelle, B. (2002). Preserving cartographic quality in DTM interpolation from contour lines. *ISPRS Journal of Photogrammetry & Remote Sensing*, 56(3):210–220.

- [Gotsman et al., 2002] Gotsman, C., Gumhold, S., and Kobbelt, L. (2002). Simplification and compression of 3d meshes. In A. Iske, E. Quak, M. F., editor, *Tutorials on Multiresolution in Geometric Modelling (Munich Summer School Lecture Notes)*. Springer.
- [GRASS GIS, 2002] GRASS GIS (2002). [www3.baylor.edu/grass/](http://www3.baylor.edu/grass/). Homepage of Grass GIS.
- [Greiner, 1994] Greiner, G. (1994). Variational Design and Fairing of Spline Surfaces. In Dæhlen, M. and Kjelldhal, L., editors, *EUROGRAPHICS '94*, volume 13, pages 143–154. Blackwell Publishers.
- [Hagen and Pottmann, 1989] Hagen, H. and Pottmann, H. (1989). Curvature continuous triangular interpolants. In Lyche, T. and Schumaker, L. L., editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 373–384. Academic Press.
- [Hahmann and Bonneau, 2000] Hahmann, S. and Bonneau, G.-P. (2000). Triangular  $g^1$  interpolation by 4-splitting domain triangles. *Computer Aided Geometric Design*, 17:731–757.
- [Hahmann and Bonneau, 2002] Hahmann, S. and Bonneau, G.-P. (2002). Parametric surfaces over arbitrary triangulations. *IEEE Transactions on Visualization and Computer Graphics*. to appear.
- [Hardy, 1971] Hardy, R. L. (1971). Multiquadric Equations of Topography and other irregular Surfaces. *Geophysical Research*, 76:1905–1915.
- [Hartley and Zisserman, 2001] Hartley, R. and Zisserman, A. (2001). *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK.
- [Heckbert and Garland, 1997] Heckbert, P. S. and Garland, M. (1997). Survey of polygonal surface simplification algorithms. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- [Heitzinger, 1999] Heitzinger, D. (1999). *Wissensbasierte 3D-Oberflächenrekonstruktion*. PhD thesis, Vienna University of Technology, Vienna, Austria.
- [Heitzinger et al., 1996] Heitzinger, D., Halmer, A., and Kager, H. (1996). 3D-Surface modelling with Basic Topological Elements. In *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXI, 4, Vienna, Austria.
- [Heitzinger and Kager, 1999] Heitzinger, D. and Kager, H. (1999). Hochwertige Geländemodelle aus Höhenlinien durch wissensbasierte Klassifikation von Problemgebieten. *Photogrammetrie-Fernerkundung-Geoinformation*, 1:29–40.
- [Hoppe et al., 1993] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1993). Mesh optimization. In *SIGGRAPH 93 Conference Proceedings*, pages 19–26. <http://research.microsoft.com/hoppe/>.
- [Hormann, 2001] Hormann, K. (2001). *Theory and Applications of Parameterizing Triangulations*. PhD thesis, University Erlangen-Nürnberg.
- [Hoschek, 1988] Hoschek, J. (1988). Intrinsic parametrization for approximation. *Computer Aided Geometric Design*, 5:27–31.
- [Hoschek and Lasser, 1993] Hoschek, J. and Lasser, D. (1993). *Fundamentals of Computer Aided Geometric Design*. A K Peters Ltd.
- [I.P.F., 2001] I.P.F. (2001). <http://www.ipf.tuwien.ac.at/>→products→scop++. Institut of Photogrammetry and Remote Sensing, Vienna University of Technology.
- [Ivrissimtzis and Seidel, 2002] Ivrissimtzis, I. and Seidel, H.-P. (2002). Polyhedra operators for mesh refinement. In *Geometric Modeling and Processing, Theory and Applications*, pages 132–137, Wako, Saitama, Japan.
- [Jensen, 1987] Jensen, T. (1987). Assembling triangular and rectangular patches and multivariate splines. In Farin, G., editor, *Geometric Modeling: Algorithms and New Trends*, pages 203–220. SIAM Publications.

- [Journel and Huijbregts, 1978] Journel, A. G. and Huijbregts, C. J. (1978). *Mining Geostatistics*. Academic Press, New York.
- [Kager, 2002] Kager, H. (2002). Adjustment of algebraic surfaces by least squared distances. In *International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, B3*, pages 472–479, Amsterdam, Netherlands.
- [Kobbelt, 1995] Kobbelt, L. (1995). *Iterative Erzeugung glatter Interpolanten*. Shaker.
- [Kobbelt, 1998] Kobbelt, L. (1998). Variational design with parametric meshes of arbitrary topology. In *Creating fair and shape preserving curves and surfaces*. Teubner.
- [Kobbelt et al., 1998a] Kobbelt, L., Campagna, S., and Seidel, H.-P. (1998a). A general framework for mesh decimation. In *Graphics Interface '98 Proceedings*, pages 43–50.
- [Kobbelt et al., 1998b] Kobbelt, L., Campagna, S., Vorsatz, J., and Seidel, H.-P. (1998b). Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH 98 Conference Proceedings*, volume 32.
- [Kolb, 1995] Kolb, A. (1995). *Optimierungsansätze bei der Interpolation verteilter Daten*. PhD thesis, Universität Erlangen-Nürnberg.
- [Kolb et al., 1995] Kolb, A., Pottmann, H., and Seidel, H.-P. (1995). Fair Surface Reconstruction Using Quadratic Functionals. In Post, F. and Göbel, M., editors, *EUROGRAPHICS '95*, volume 14, pages 469–479. Blackwell Publishers.
- [Kós, 2001] Kós, G. (2001). An algorithm to triangulate surfaces in 3d using unorganised point clouds. *Computing Suppl.*, 14:219–232.
- [Kraus, 1997] Kraus, K. (1997). Eine neue Methode zur Interpolation und Filterung von Daten mit schiefer Fehlerverteilung. *Österreichische Zeitschrift für Vermessung & Geoinformation*, 1:25–30.
- [Kraus, 2000] Kraus, K. (2000). *Photogrammetrie, Band 3, Topographische Informationssysteme*. Dümmler. An english edition by Taylor and Francis (translator: H. Rüter) is in preparation.
- [Kraus and Mikhail, 1972] Kraus, K. and Mikhail, E. M. (1972). Linear least-squares interpolation. *Photogrammetric Engineering*, 38:1016–1029.
- [Kraus and Pfeifer, 2001] Kraus, K. and Pfeifer, N. (2001). Advanced DTM generation from lidar data. In *International Archives of Photogrammetry and Remote Sensing, Vol XXXIV, 3/W4*, Annapolis, MD, USA.
- [Lancaster and Salkauskas, 1986] Lancaster, P. and Salkauskas, K. (1986). *Curve and surface fitting, An Introduction*. Academic Press.
- [Lee et al., 1998] Lee, A. W. F., Sweldens, W., Schröder, P., Cowsar, L., and Dobkin, D. (1998). Maps: Multiresolution adaptive parameterization of surfaces. In *SIGGRAPH 98 Conference Proceedings*, volume 32, pages 95–104.
- [Linsen and Prautzsch, 2001] Linsen, L. and Prautzsch, H. (2001). Global versus local triangulations. In *Eurographics 2001 Proceedings*.
- [Loop, 1987] Loop, C. (1987). Smooth subdivision surfaces baes on triangles. Master's thesis, University of Utah, Department of Mathematics.
- [Loop, 1994] Loop, C. (1994). A  $g^1$  triangular spline surface of arbitrary topological type. *Computer Aided Geometric Design*, 11:303–330.
- [Lounsbery et al., 1997] Lounsbery, M., DeRose, T. D., and Warren, J. (1997). Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1).
- [Ma et al., 2002] Ma, W., Ma, X., Tso, S.-K., and Pan, Z. (2002). Subdivision surface fitting from a dense triangle mesh. In *Geometric Modeling and Processing, Theory and Applications*, pages 94–103, Wako, Saitama, Japan.

- [Mann et al., 1992] Mann, S., Loop, C., Lounsbery, M., Meyers, D., Painter, J., DeRose, T., and Sloan, K. (1992). A Survey of Parametric Scattered Data Fitting Using Triangular Interpolants. In Hagen, H., editor, *Curve and Surface Design*. SIAM Publications.
- [Mäntylä, 1988] Mäntylä, M. (1988). *An Introduction to Solid Modeling*. Computer Science Press, Rockville, Maryland.
- [Mencl, 1995] Mencl, R. (1995). A graph-based approach to surface reconstruction. *Computer Graphics Forum*, 14(3):445–456.
- [Meyer et al., 2002] Meyer, M., Desbrun, M., Schröder, P., and Barr, A. H. (2002). *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*. Springer.
- [Mikhail, 1976] Mikhail, E. (1976). *Observations And Least Squares*. IEP-A Dun-Donnelley, New York.
- [Mokhtarian et al., 2001] Mokhtarian, F., Khalili, N., and Yuen, P. (2001). Curvature computation on free-form 3-d meshes at multiple scales. *Computer Vision and Image Understanding*, 83:118–139.
- [Nasri and Sabin, 2002] Nasri, A. H. and Sabin, M. A. (2002). Taxonomy of interpolation constraints on recursive subdivision surfaces. *The Visual Computer*, 18(5-6):382–403.
- [Nielson, 1987] Nielson, G. M. (1987). A visually continuous Transfinite triangular Interpolant. In Farin, G., editor, *Geometric Modeling: Applications and new trends*, pages 235–245. SIAM Publications.
- [Opitz and Pottmann, 1994] Opitz, K. and Pottmann, H. (1994). Computing Shortest Paths on Polyhedra: Applications in geometric modeling and scientific visualization. *International Journal of Computational Geometry & Applications*, 4(2):165–178.
- [Peng et al., 1996] Peng, W., Pilouk, M., and Tempfli, K. (1996). Generalizing relief representations using digitized contours. In *International Archives of Photogrammetry and Remote Sensing, Vol. XXXI, B4*, pages 649–654, Vienna, Austria.
- [Peters, 1991] Peters, J. (1991). Smooth interpolation of a mesh of curves. *Constr. Approx.*, 7:221–247.
- [Pfeifer, 1997] Pfeifer, N. (1997). 3-dimensionale Oberflächenmodelle auf Basis einer Triangulierung — Flächenrepräsentation. Diplomarbeit, Technische Universität Wien.
- [Pfeifer and Briese, 2001] Pfeifer, N. and Briese, C. (2001). Airborne laser scanning and derivation of digital terrain models. In Grün and Kahmen, editors, *Fifth Conference on Optical 3-D Measurement Techniques*, pages 80–87, Vienna, Austria.
- [Pfeifer and Pottmann, 1996] Pfeifer, N. and Pottmann, H. (1996). Surface models on the basis of a triangular mesh – surface reconstruction. In *International Archives of Photogrammetry and Remote Sensing, Vol. XXXI, B3*, pages 638–643, Vienna, Austria.
- [Piper, 1987] Piper, B. (1987). Visually smooth interpolation with triangular Bézier patches. In Farin, G., editor, *Geometric Modeling: Algorithms and New Trends*, pages 221–233. SIAM Publications.
- [Preusser, 1990] Preusser, A. (1990).  $C^1$ - and  $C^2$ -interpolation on triangles with quintic and nonic bivariate polynomials. *Transactions on Mathematical Software*, 16(3):253–257.
- [Samet, 1989] Samet, H. (1989). *The Design and Analysis of Spatial Data Structures*. Addison Wesley, Reading, Massachusetts.
- [Schlüter, 1999] Schlüter, M. (1999). *Von der 2 1/2D- zur 3D-Flächenrekonstruktion für die photogrammetrische Rekonstruktion im Objektraum*. Number 506 in C. Verlag der Bayerischen Akademie der Wissenschaften, München, Doktorarbeit, Technische Universität Darmstadt.
- [Shirman and Séquin, 1987] Shirman, L. A. and Séquin, C. H. (1987). Local surface interpolation with Bézier patches. *Computer Aided Geometric Design*, 4:279–295.

- [Singer, 1995] Singer, P. (1995). *Die Berechnung von Minimalflächen, Seifenblasen, Membrane und Pneus aus geodätischer Sicht*. Number 448 in C. Verlag der Bayerischen Akademie der Wissenschaften, München, Doktorarbeit, Universität Stuttgart.
- [Stollnitz et al., 1996] Stollnitz, E. J., DeRose, T. D., and Salesin, D. H. (1996). *Wavelets for Computer Graphics*. Morgan Kaufmann, San Francisco.
- [Takagi and Shibasaki, 1996] Takagi, M. and Shibasaki, R. (1996). An interpolation method for continental DEM generation using small scale contour maps. In *International Archives of Photogrammetry and Remote Sensing, Vol. XXXI, B4*, pages 847–852, Vienna, Austria.
- [TopoSys, 2002] TopoSys (2002). [www.toposys.de](http://www.toposys.de). Homepage of TopoSys.
- [Tutte, 1963] Tutte, W. T. (1963). How to draw a planar graph. *Proc. London Math. Soc.*, 13:304–320.
- [Verbree and van Oosterom, 2001] Verbree, E. and van Oosterom, P. (2001). Scanline forced delauney tens for surface representation. In *International Archives of Photogrammetry and Remote Sensing, Vol. XXXIV, 3/W4*, pages 45–51, Annapolis, Maryland.
- [Vosselman, 2000] Vosselman, G. (2000). Slope based filtering of laser altimetry data. In *International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, B3*, pages 935–942, Amsterdam, Netherlands.
- [Várady, 1987] Várady, T. (1987). Survey and new results in n-sided patch generation. In Martin, R., editor, *The Mathematics of Surfaces II. (Proc. of the 2nd IMA Conf.)*. Oxford University Press.
- [Waterloo Hydrogeologic, 2002] Waterloo Hydrogeologic, I. (2002). [www.flowpath.com](http://www.flowpath.com) →software→surfer. Homepage of Surfer.
- [Wilson and Beineke, 1979] Wilson, R. J. and Beineke, L. W., editors (1979). *Applications of Graph Theory*. Academic Press, London, UK.
- [Zorin and Schröder, 2000] Zorin, D. and Schröder, P. (2000). Subdivision for modeling and animation. SIGGRAPH 2000 Course Notes. <http://www.multires.caltech.edu/pubs/pubs.htm>.
- [Zorin et al., 1996] Zorin, D., Schröder, P., and Sweldens, W. (1996). Interpolating subdivision for meshes with arbitrary topology. In *SIGGRAPH 96 Conference Proceedings*, pages 189–192.

# Appendix A

## Approximating surfaces

In this section the determination of approximating surfaces to a given set of points is treated. It is by no means complete but restricted to planes, quadrics and paraboloids, which are special cases of quadrics. In photogrammetry and surveying these kinds of problems are known under the term adjustment. Adjustment calculus is performed by minimization of a sum of errors or residuals, usually a sum of squared errors (least squares adjustment). The errors are considered to be errors which were made when observing (measuring) a certain quantity. The solving of adjustment problems includes regression calculus, where the solution is found by solving a linear system of equations. One example is the fitting of a line (regression line) to data points, when a functional dependency from the observed function values to the parameter positions exist. The parameter positions are assumed to be known exactly, an error can only occur in the quantity of the observation. In this case the distances to be minimized are measured in the direction of the observation axis. In more general cases the system has to be linearized first and is solved in an iterative fashion. So, regression is one way of adjustment, but often the adjustment result can be found as the solution of an eigenvalue-eigenvector problem. An example is finding an adjusting line where the distance from the line is measured orthogonal to the line, and not, as in the previous example, in a predefined direction. General background on adjustment calculation in photogrammetry and surveying can be found in [Mikhail, 1976].

### A.1 Adjusting plane

Given are  $n$  points  $\mathbf{P}_i$  and weights  $w_i$ . The task is to find a plane  $\tau$  for which the sum  $\Delta$  of the squares of the weighted orthogonal distances  $d_i$  becomes a minimum. The plane  $\tau$  is described in Hesse's normal form with normal vector  $\mathbf{n}$  and constant  $c$ .

$$\begin{aligned}\mathbf{P}_i &= (x_i \ y_i \ z_i)^\top \\ \tau &: \mathbf{n}^\top \mathbf{P} + c = 0 \\ \mathbf{n} &= (n_x \ n_y \ n_z)^\top, \text{ with } \|\mathbf{n}\| = 1 \\ d_i &= \mathbf{n}^\top \mathbf{P}_i + c \\ w_i &> 0 \\ \Delta &= \sum_{i=1}^n (\mathbf{n}^\top \mathbf{P}_i + c)^2 w_i \\ \mathbf{C} &= \frac{\sum_{i=1}^n \mathbf{P}_i w_i}{\sum_{i=1}^n w_i} \quad (\text{Center of Gravity})\end{aligned}\tag{A.1}$$

In this notation  $\mathbf{n}$  and  $c$  have to be determined, so that  $\Delta$  becomes a minimum. Rearranging Eq. A.1 and omitting the summation bounds leads to

$$\Delta = \sum (\mathbf{n}^\top \mathbf{P}_i)^2 w_i + c^2 \sum w_i + 2c \mathbf{n}^\top \sum \mathbf{P}_i w_i$$

Minimizing  $\Delta$  under the assumption of a predefined direction  $\tilde{\mathbf{n}}$  by setting the partial derivative  $\frac{\partial}{\partial c}$  to zero leads to

$$\begin{aligned} 2c \sum w_i + 2\tilde{\mathbf{n}}^\top \sum \mathbf{P}_i w_i &= 0 \\ c &= \frac{2\tilde{\mathbf{n}}^\top \mathbf{C} \sum w_i}{2 \sum w_i} \end{aligned}$$

Of course, this leads to the expected result, that the adjusting plane runs through the center of gravity (also barycenter) and therefore  $c = \mathbf{n}^\top \mathbf{C}$ . It is therefore sufficient to minimize  $\Delta$  for the points  $\bar{\mathbf{P}}_i$  which have been reduced to the center of gravity. The adjusting plane contains the origin of this local co-ordinate system and only  $\mathbf{n}$  is unknown. For its determination the side condition  $\|\mathbf{n}\| = 1$  has to be considered, which is achieved by minimizing the Lagrange function  $\Omega$ . Introducing matrices instead of sums the formulae for determining  $\mathbf{n}$  are:

$$\begin{aligned} \bar{\mathbf{P}}_i &= \mathbf{P}_i - \mathbf{C} = (\bar{x}_i \ \bar{y}_i \ \bar{z}_i)^\top \\ \mathbf{A} &= \begin{pmatrix} \bar{x}_1 & \bar{y}_1 & \bar{z}_1 \\ \vdots & & \vdots \\ \bar{x}_n & \bar{y}_n & \bar{z}_n \end{pmatrix} \\ \mathbf{W} &= \text{diag}(w_i) \\ \Delta &= \mathbf{n}^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} \mathbf{n} \\ \Omega &= \Delta - \lambda (\mathbf{n}^\top \mathbf{n} - 1) \end{aligned} \tag{A.2}$$

Eq. A.2 is easily verified, by recalling that the product of one line of  $\mathbf{A}$  with  $\mathbf{n}$  denotes the distance from the corresponding point to the plane with normal vector  $\mathbf{n}$ . Setting the derivative after the Lagrange factor  $\lambda$  to zero yields the side condition and setting the derivative after  $\mathbf{n}$  to zero leads to an eigenvalue problem.

$$\begin{aligned} \frac{\partial \Omega}{\partial \mathbf{n}} &= 2\mathbf{A}^\top \mathbf{W} \mathbf{A} \mathbf{n} - 2\lambda \mathbf{n} \\ \mathbf{A}^\top \mathbf{W} \mathbf{A} \mathbf{n} &= \lambda \mathbf{n} \\ (\mathbf{A}^\top \mathbf{W} \mathbf{A} - \lambda \mathbf{E}) \mathbf{n} &= \mathbf{0} \end{aligned} \tag{A.3}$$

Here,  $\mathbf{E}$  is a unit matrix of dimension 3 and  $\mathbf{0}$  the zero vector of dimension 3. The three eigenvectors  $\mathbf{n}_1$ ,  $\mathbf{n}_2$  and  $\mathbf{n}_3$  with their eigenvalues  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are the solutions of the system. Considering

$$\mathbf{n}_j^\top \mathbf{n}_j - 1 = 0, \quad j = 1, \dots, 3$$

leads to

$$\begin{aligned} \Delta = \Omega &= \mathbf{n}^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} \mathbf{n} = \mathbf{n}^\top \lambda \mathbf{n} \\ \Delta_j &= \lambda_j \end{aligned}$$

which means that the eigenvector to the smallest eigenvalue (say  $\lambda_s$ ) is the solution of the problem. The r.m.s.e. of the adjustment is

$$\sigma = \sqrt{\frac{\lambda_s}{n-3}}$$

The condition that the weights must be positive can be loosened a bit. At least three not collinear points must have positive weights, the other points may have weight zero, too.

## A.2 Adjusting quadric

A quadric (also quadratic surface) is a second-order algebraic surface. A quadric  $Q(x, y, z)$ <sup>1</sup> is the set of zeros (i.e. set of points fulfilling the equation)

$$\mathbf{P}^\top \mathbf{Q} \mathbf{P} + \mathbf{b}^\top \mathbf{P} + c = 0 \quad (\text{A.4})$$

with  $\mathbf{Q}$  a symmetric real-valued  $3 \times 3$  matrix,  $\mathbf{b} \in \mathbb{R}^3$  and  $c \in \mathbb{R}$ . It can be re-written:

$$(p_x \ p_y \ p_z \ 1) \begin{pmatrix} q_{11} & q_{12}/2 & q_{13}/2 & b_x/2 \\ & q_{22} & q_{23}/2 & b_y/2 \\ & & q_{33} & b_z/2 \\ \text{symm.} & & & c \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = 0 \quad (\text{A.5})$$

$$\tilde{\mathbf{P}}^\top \tilde{\mathbf{Q}} \tilde{\mathbf{P}} = 0$$

The values of  $\tilde{\mathbf{Q}}$  are only determined up to a scale factor. The tangent plane in a point  $\mathbf{P}$  has the normal vector  $\tilde{\mathbf{Q}} \tilde{\mathbf{P}}$ . For an arbitrary point  $\tilde{\mathbf{P}}_A$ , the scalar  $\tilde{\mathbf{P}}_A^\top \tilde{\mathbf{Q}} \tilde{\mathbf{P}}_A$  is the algebraic distance from the point to the quadric.

The adjustment problem can be defined as follows. Given is a set of  $n$  points  $\mathbf{P}_i = (x_i \ y_i \ z_i)^\top$ , the adjusting quadric is the quadric which minimizes:

$$\Delta = \sum_{i=1}^n (\tilde{\mathbf{P}}_i^\top \tilde{\mathbf{Q}} \tilde{\mathbf{P}}_i)^2 \quad (\text{A.6})$$

Here, no weighting is considered, but the extension for a weighted adjustment are straight forward. As mentioned before,  $\tilde{\mathbf{Q}}$  includes a scale factor. Thus, the trivial solution  $q_{ij} = 0$ ,  $i, j = 1, \dots, 4$  always exists. Eq. A.5 can be rewritten:

$$\begin{aligned} \mathbf{X}^\top \mathbf{q} &= 0 \\ \mathbf{X}^\top &= (x^2 \ y^2 \ z^2 \ xy \ yz \ zx \ x \ y \ z \ 1) \\ \mathbf{q}^\top &= (q_{11} \ q_{22} \ q_{33} \ q_{12} \ q_{23} \ q_{13} \ q_{14} \ q_{24} \ q_{34} \ q_{44}) \end{aligned}$$

So, neglecting the summation bounds, Eq. A.6 becomes

$$\begin{aligned} \Delta &= \sum (\mathbf{X}_i^\top \mathbf{q})^2 \\ \mathbf{A} &= \begin{pmatrix} x_1^2 & y_1^2 & z_1^2 & x_1 y_1 & y_1 z_1 & z_1 x_1 & x_1 & y_1 & z_1 & 1 \\ \vdots & & & & & & & & & \vdots \\ x_n^2 & y_n^2 & z_n^2 & x_n y_n & y_n z_n & z_n x_n & x_n & y_n & z_n & 1 \end{pmatrix} \\ \Delta &= \mathbf{q}^\top \mathbf{A}^\top \mathbf{A} \mathbf{q} \end{aligned} \quad (\text{A.7})$$

With the side condition  $\|\mathbf{q}\| = 1$  the sum of squares of algebraic distances  $\Delta$  can be minimized. The side condition is necessary to preclude the trivial solution. Additionally, the system is only determined up to a scale factor, and this side condition fixes it. The minimization is performed by the Lagrangian technique and leads to an eigenvalue problem.

$$\begin{aligned} \Omega &= \mathbf{q}^\top \mathbf{A}^\top \mathbf{A} \mathbf{q} - \lambda (\|\mathbf{q}\| - 1) \\ \frac{\partial \Omega}{\partial \mathbf{q}} &= 2\mathbf{A}^\top \mathbf{A} \mathbf{q} - 2\lambda \mathbf{q} = 0 \\ (\mathbf{A}^\top \mathbf{A} - \lambda \mathbf{E}) \mathbf{q} &= \mathbf{0} \end{aligned} \quad (\text{A.8})$$

The eigenvectors to the eigenvalues of Eq. A.8 contain the factors of the quadric. The eigenvector to the smallest eigenvalue has the smallest sum of squares of algebraic distances. The different solutions describe different

<sup>1</sup>In the following the letter 'Q' is used to denote a quadric and not, as it is usual in adjustment calculation, the co-factor matrix containing accuracies of observations and their correlations. This co-factor matrix is the inverse of the weight matrix.

quadrics, which means that the best fitting solution might be an ellipsoid, while the second best fitting might be a two sheeted hyperboloid.

The values in Eq. A.8 are co-ordinates to the fourth power. Thus it is advantageous to reduce the points to the center of gravity. If the co-ordinates are very huge but relatively close to each other, as it is the example in terrain modelling, any of the given points can be used as reduction point, too. If the points are first reduced to  $\mathbf{M}$ , then these shifted points are used in Eq. A.7, and the formula for the quadric is:

$$(\mathbf{P} - \mathbf{M})^\top \mathbf{Q}(\mathbf{P} - \mathbf{M}) + \mathbf{b}^\top (\mathbf{P} - \mathbf{M}) + c = 0$$

In contrast to an adjusting plane, the value of  $\Delta$  has no geometrical meaning. In the following a method is sketched which can be used to minimize the (sum of squares of) Euclidean distances from the given points to the quadric. It is achieved by a *weight iteration*. To do so, the foot points for each of the given points must be computed and the distances between the foot points and the original points computed. After a first computation with equal weights for all points, the weight is defined as the square of the Euclidean distance divided through the algebraic distance between the given point and its foot point. If the points are not too far away from the quadric the foot point  $\mathbf{F}$  of a point  $\mathbf{P}$  can be determined in an iterative fashion. First the line through  $\mathbf{P}$  and with the direction of the quadric gradient in this point (i.e.  $\hat{\mathbf{Q}}\mathbf{P}$ ) is intersected with the quadric. The first intersection has to be taken. (The number of intersections can reach from zero to two.) This is the first approximation of  $\mathbf{F}$ . The quadric gradient at this intersection point ( $\hat{\mathbf{Q}}\mathbf{F}$ ) is used as improved direction of the line. This process is iterated, until the intersection angle of the line with the quadric is sufficiently close to a right angle. It converges, if the points are not too far away from the quadric.

A method where the computation of the foot point is not necessary is described in [Kager, 2002]. It provides a good approximation for the computation of the real residuals.

### A.2.1 Adjusting quadric through one point

In the previous section the determination of an approximating (adjusting) general quadric was described. The algebraic distances from the given points to the quadric are minimized in a quadratic way, but points with larger distances are not necessarily further away from the quadric – in an Euclidean measure – than points with small distances.

In [Benkő et al., 2001] a method is described for the determination of quadrics which interpolate one point. Additionally, the gradient (normal vector) in this point is a unit vector. This means, that in the vicinity of this point the algebraic distances are close to the Euclidean distances. It shall be noted, that this method is used for the estimation of normal vectors in points from their neighborhood in [Benkő et al., 2001]. Also in [Hartley and Zisserman, 2001] a general quadric approximation is used for an estimation of the normal vector, but it is mentioned there, that the estimation of the quadric itself from noisy data is not reliable if the points cover only a limited area on the computed quadric.

The general form of a quadric is (cf. Eq. A.4) a set of points  $\mathbf{P}$  fulfilling:

$$\mathbf{P}^\top \mathbf{Q} \mathbf{P} + \mathbf{b}^\top \mathbf{P} + c = 0$$

If  $c$  is zero, the quadric interpolated the origin of the co-ordinate system. In order to interpolate a point  $\mathbf{P}_0$  the simplest method is to make this point the origin of a local co-ordinate system by a simple translation. The quadric then takes the form:

$$(\mathbf{P} - \mathbf{P}_0)^\top \mathbf{Q}(\mathbf{P} - \mathbf{P}_0) + \mathbf{b}^\top (\mathbf{P} - \mathbf{P}_0) = 0$$

With the condition  $\|\mathbf{b}\| = 1$  the algebraic distances from points near  $\mathbf{P}_0$  to the quadric are similar to their Euclidean distances. Additionally, the normal vector cannot be the zero vector, which prohibits a cone with its vertex in the origin as a solution. In order to determine a quadric which approximates the points  $\mathbf{P}_i, i = 1, \dots, n$  and interpolates  $\mathbf{P}_0$  under the given side condition, the following term has to be minimized.

$$\begin{aligned} \Delta &= \sum_{i=1}^n \left( (\mathbf{P} - \mathbf{P}_0)^\top \mathbf{Q}(\mathbf{P} - \mathbf{P}_0) + \mathbf{b}^\top (\mathbf{P} - \mathbf{P}_0) \right)^2 \\ \|\mathbf{b}\| &= 1 \end{aligned} \tag{A.9}$$

With  $(\bar{x}_i \bar{y}_i \bar{z}_i)^\top = \mathbf{P}_i - \mathbf{P}_0$  this can be rewritten in matrix form:

$$\begin{aligned}
\Delta &= \mathbf{v}^\top \mathbf{v} \\
\mathbf{v} &= \mathbf{L}\mathbf{q} + \mathbf{X}\mathbf{b} \\
\mathbf{X} &= \begin{pmatrix} \bar{x}_1 & \bar{y}_1 & \bar{z}_1 \\ \vdots & & \vdots \\ \bar{x}_n & \bar{y}_n & \bar{z}_n \end{pmatrix} \\
\mathbf{q} &= (q_{11} \ q_{22} \ q_{33} \ q_{12} \ q_{23} \ q_{13})^\top \\
\mathbf{L} &= \begin{pmatrix} \bar{x}_1^2 & \bar{y}_1^2 & \bar{z}_1^2 & \bar{x}_1\bar{y}_1 & \bar{y}_1\bar{z}_1 & \bar{x}_1\bar{z}_1 \\ \vdots & & & & & \vdots \\ \bar{x}_n^2 & \bar{y}_n^2 & \bar{z}_n^2 & \bar{x}_n\bar{y}_n & \bar{y}_n\bar{z}_n & \bar{x}_n\bar{z}_n \end{pmatrix} \\
\Delta &= \mathbf{q}^\top \mathbf{L}^\top \mathbf{L} \mathbf{q} + 2\mathbf{b}^\top \mathbf{X}^\top \mathbf{L} \mathbf{q} + \mathbf{b}^\top \mathbf{X}^\top \mathbf{X} \mathbf{b}
\end{aligned} \tag{A.10}$$

Minimizing  $\Delta$  under the side condition  $\|\mathbf{b}\| - 1 = 0$  is performed by the Lagrange technique.

$$\begin{aligned}
\Omega &= \mathbf{v}^\top \mathbf{v} - \lambda(\|\mathbf{b}\| - 1) \\
&= \mathbf{q}^\top \mathbf{L}^\top \mathbf{L} \mathbf{q} + 2\mathbf{q}^\top \mathbf{L}^\top \mathbf{X} \mathbf{b} + \mathbf{b}^\top \mathbf{X}^\top \mathbf{X} \mathbf{b} - \lambda(\|\mathbf{b}\| - 1) \\
\frac{\partial \Omega}{\partial \mathbf{q}} &= 2\mathbf{L}^\top \mathbf{L} \mathbf{q} + 2\mathbf{L}^\top \mathbf{X} \mathbf{b} \\
\frac{\partial \Omega}{\partial \mathbf{b}} &= 2\mathbf{X}^\top \mathbf{L} \mathbf{q} + 2\mathbf{X}^\top \mathbf{X} \mathbf{b} - 2\lambda \mathbf{b} \\
\frac{\partial \Omega}{\partial \lambda} &= -\mathbf{b}^\top \mathbf{b} + 1
\end{aligned}$$

Setting the three partial derivatives to zero leads to a system of equations.

$$\begin{aligned}
\frac{\partial \Omega}{\partial \mathbf{q}} &: \mathbf{L}^\top \mathbf{L} \mathbf{q} + \mathbf{L}^\top \mathbf{X} \mathbf{b} = \mathbf{0} \\
&\mathbf{q} = -(\mathbf{L}^\top \mathbf{L})^{-1} \mathbf{L}^\top \mathbf{X} \mathbf{b}
\end{aligned} \tag{A.11}$$

$$\begin{aligned}
\frac{\partial \Omega}{\partial \mathbf{b}} &: \mathbf{X}^\top \mathbf{L} \mathbf{q} + \mathbf{X}^\top \mathbf{X} \mathbf{b} - \lambda \mathbf{b} = \mathbf{0} \\
&-\mathbf{X}^\top \mathbf{L} (\mathbf{L}^\top \mathbf{L})^{-1} \mathbf{L}^\top \mathbf{X} \mathbf{b} + \mathbf{X}^\top \mathbf{X} \mathbf{b} - \lambda \mathbf{b} = \mathbf{0} \\
&(\mathbf{X}^\top \mathbf{X} - \mathbf{X}^\top \mathbf{L} (\mathbf{L}^\top \mathbf{L})^{-1} \mathbf{L}^\top \mathbf{X} - \lambda \mathbf{E}) \mathbf{b} = \mathbf{0}
\end{aligned} \tag{A.12}$$

$$\frac{\partial \Omega}{\partial \lambda} : \mathbf{b}^\top \mathbf{b} = 1 \tag{A.13}$$

In Eq. A.12 an eigenvalue-eigenvector problem is posed for the matrix:

$$\mathbf{X}^\top \mathbf{X} - \mathbf{X}^\top \mathbf{L} (\mathbf{L}^\top \mathbf{L})^{-1} \mathbf{L}^\top \mathbf{X}$$

Of course, this matrix is symmetric and has therefore three real eigenvalues. The eigenvectors are scaled according to Eq. A.13. The eigenvector to the smallest eigenvalue is the normal of the best approximating quadric in  $\mathbf{P}_0$ . If the quadric itself shall be determined, too, then the parameters of  $\mathbf{Q}$  (i.e. the values of  $\mathbf{q}$ ) have to be determined according to Eq. A.11.

A weighting can be introduced by inserting a weight matrix in Eq. A.10 between the residual vectors  $\mathbf{v}$ .

### A.3 Adjusting second order polynomial surface

The formula for a second order polynomial  $\mathbf{q}$  over the  $uv$ -plane, i.e. the parameter domain, is:

$$\mathbf{q}(u, v) = \mathbf{a}u^2 + \mathbf{b}uv + \mathbf{c}v^2 + \mathbf{d}u + \mathbf{e}v + \mathbf{f} \tag{A.14}$$

Here, the vectors  $\mathbf{a}, \dots, \mathbf{f} \in \mathbb{R}^3$  are the parameters of the polynomial. Only the vector-valued case is described. If the vectors are replaced by scalars, the surface is a paraboloid.

Given are  $n$  points  $\mathbf{P}_i = (x_i \ y_i \ z_i)^\top$ , associated parameter points  $\mathbf{U}_i = (u_i \ v_i)^\top$  and weights  $w_i^2$ . The task is to find an adjusting polynomial surface of degree two, minimizing the sum of squares of distances from the points to the surface. The distances  $d_i$  are not measured orthogonal to the surface, but  $d_i = \|\mathbf{P}_i - \mathbf{q}(u_i, v_i)\|$ . So the squares of distances from the given points to the polynomial at the corresponding parameter locations are minimized.

$$\Delta = \sum_{i=1}^n d_i^2 w_i = \sum_{i=1}^n (\mathbf{P}_i - \mathbf{q}(u_i, v_i))^\top w_i (\mathbf{P}_i - \mathbf{q}(u_i, v_i))$$

Again, this can be formulated in matrix form:

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} u_1^2 & u_1 v_1 & v_1^2 & u_1 & v_1 & 1 \\ \vdots & & & & & \vdots \\ u_n^2 & u_n v_n & v_n^2 & u_n & v_n & 1 \end{pmatrix} \\ \mathbf{W} &= \text{diag}(w_i) \\ \mathbf{x} &= (\mathbf{a} \ \mathbf{b} \ \mathbf{c} \ \mathbf{d} \ \mathbf{e} \ \mathbf{f})^\top \\ \mathbf{1} &= (\mathbf{P}_1 \ \dots \ \mathbf{P}_n)^\top \\ \Delta &= (\mathbf{A}\mathbf{x} - \mathbf{1})^\top \mathbf{W} (\mathbf{A}\mathbf{x} - \mathbf{1}) \end{aligned}$$

Minimizing  $\Delta$  with respect to the unknowns  $\mathbf{x}$  leads to the equation system:

$$\mathbf{A}^\top \mathbf{W} \mathbf{A} \mathbf{x} = \mathbf{A}^\top \mathbf{W} \mathbf{1} \quad (\text{A.15})$$

This is one equation system with three right sides and three unknown vectors  $\mathbf{x}$ , corresponding to each co-ordinate direction. These three equation systems share the same design matrix, but apart from that they are independent from each other. If only the values of the z-co-ordinates are changed, this does not change the parameters (unknowns) for the x- and y-direction.

To solve the system Eq. A.15 at least six points are necessary. In this case no over determination is given and the paraboloid can show unwanted undulations (wiggles). Therefore, always more than six points should be used. A different strategy to avoid these oscillations and solve the system, even if less than six points are given, is to apply preventive regularization. Additional observations are introduced fixing the unknowns in case of under determination, e.g.

$$\mathbf{a} = \mathbf{b} = \mathbf{c} = \mathbf{d} = \mathbf{e} = \mathbf{f} = \mathbf{0} \quad (\text{A.16})$$

Of course, the regularization can only be applied to the first three components, too, or any other combination. The question of weighting these equations versus the point observations is a little tricky. The weight has to be large enough to prevent oscillatory behavior but small enough to allow the surface to adapt to the given point set.

The determination of the adjusting polynomial surface is independent of the datum of the co-ordinate system. By this it is meant that an adjusting polynomial surface determined from the given point set or computed from the point set obtained after applying the transformation differ exactly by this transformation. This is not only valid for congruency transformation, but also for the more general class of affine transformation. For completeness, a proof will be given. The weighting of the observations can be neglected. This is obvious, if the original interpretation of weights, i.e. repetition number for observations, is recalled. To avoid misunderstandings, it is also noted, that the parameterization (the points  $\mathbf{U}_i$ ) are not affected by the transformation.

An affine transformation is composed of a translation  $\Delta \mathbf{P}$  and of a multiplication with a general  $3 \times 3$ -matrix  $\mathbf{R}$ . Using the notation from above it can be written:

$$\tilde{\mathbf{P}} = \mathbf{R} \mathbf{P} + \Delta \mathbf{P}$$

---

<sup>2</sup>If the coefficients are only scalars the surface is a paraboloid and the formulae are much simpler. In this case one can write  $\mathbf{U}_i = (x_i \ y_i)$  and the paraboloids for the first two co-ordinate directions degenerate to planes:  $q(x) = u$  and  $q(y) = v$ . To approximate points in 3D space the transformation to a local co-ordinate system (defined e.g. by an adjusting plane through the points) may be necessary.

The tilde refers to the transformed terms, to the points as well as to the coefficients of the polynomial. It is obvious, that a translation of the point set will only show an effect in the value of  $\mathbf{f}$ :

$$\tilde{\mathbf{f}} = \mathbf{f} + \Delta\mathbf{P}$$

Therefore, only the transformation with the matrix will be investigated in the following. As above it holds:

$$\mathbf{l} = \begin{pmatrix} \mathbf{P}_1^\top \\ \vdots \\ \mathbf{P}_n^\top \end{pmatrix}$$

$$\mathbf{x} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{l}$$

This formula describes the solution given above. The matrix  $\mathbf{x}$  contains the unknown vector coefficients of the polynomial. It confers to the original point set, not subjected to the transformation. The given points under the transformation<sup>3</sup> are:

$$\tilde{\mathbf{P}}_i = \mathbf{R}\mathbf{P}_i$$

$$\tilde{\mathbf{P}}_i^\top = \mathbf{P}_i^\top \mathbf{R}^\top$$

$$\tilde{\mathbf{l}} = \begin{pmatrix} \mathbf{P}_1^\top \\ \vdots \\ \mathbf{P}_n^\top \end{pmatrix} \mathbf{R}^\top = \mathbf{l}\mathbf{R}^\top$$

$$\tilde{\mathbf{x}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \tilde{\mathbf{l}}$$

$$= \mathbf{x}\mathbf{R}^\top$$

So, the coefficients  $\tilde{\mathbf{a}}, \dots, \tilde{\mathbf{f}}$  of the polynomial which approximates the transformed point set are obtained by multiplying the coefficients  $\mathbf{a}, \dots, \mathbf{f}$  from the original polynomial with the transformation matrix  $\mathbf{R}$ .

If, on the other hand, the transformation is applied directly to the original surface, one obtains:

$$\mathbf{q}(u, v) = \mathbf{a}u^2 + \mathbf{b}uv + \mathbf{c}v^2 + \mathbf{d}u + \mathbf{e}v + \mathbf{f}$$

$$\tilde{\mathbf{q}}(u, v) = \mathbf{R}\mathbf{q}(u, v) = \mathbf{R}(\mathbf{a}u^2 + \mathbf{b}uv + \mathbf{c}v^2 + \mathbf{d}u + \mathbf{e}v + \mathbf{f})$$

It is easy to see, that this surface is exactly the one obtained from the adjustment of the transformed point set  $\tilde{\mathbf{P}}_i$ .

---

<sup>3</sup>As the translation is not considered here, it is only a linear transformation and not an affine one.



## Appendix B

# Bézier triangles

In this section some properties of Bézier triangles are presented. It is by no means complete, a more detailed description can be found in [Hoschek and Lasser, 1993] and [Farin, 2002]. The following description is taken from [Pfeifer and Pottmann, 1996].

A Bézier triangle (triangular Bézier patch) represents a polynomial surface of degree  $n$  in  $\mathbb{R}^3$ . Its parameter domain is a triangle  $\triangle(\mathbf{R}, \mathbf{S}, \mathbf{T})$ . A point  $\mathbf{U}$  in this triangle can be described by its barycentric coordinates, the triple  $(r, s, t)$  with  $r + s + t = 1$  and  $\mathbf{U}(r, s, t) = r\mathbf{R} + s\mathbf{S} + t\mathbf{T}$ . If  $r, s, t$  are larger than zero, the point lies in the interior of the triangle (Fig. B.1).

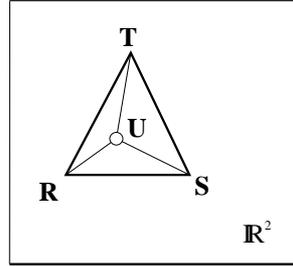


Figure B.1: Barycentric co-ordinates, for  $\mathbf{U}$  the co-ordinates are  $(r, s, t) = (0.458, 0.250, 0.292)$

The coefficients of the polynomial are given in the form of a control net  $\mathbf{P}_{ijk}$ , with  $i, j, k \geq 0$  and  $i + j + k = n$  (see Fig. B.2). The points  $\mathbf{P}_{ijk}$  are called control points, as the shape of the surface can be controlled by their position. For a patch of degree  $n$  there are  $(n + 1)(n + 2)/2$  control points, also called Bézier points.

The point  $\mathbf{P}(r, s, t)$  on the surface to the point  $\mathbf{U}$  in the parameter triangle is computed with the following recursive de Casteljau algorithm (see Fig. B.3, right):

1.  $\mathbf{P}_{ijk}^0 := \mathbf{P}_{ijk}$
2.  $\mathbf{P}_{opq}^l := r\mathbf{P}_{o+1,p,q}^{l-1} + s\mathbf{P}_{o,p+1,q}^{l-1} + t\mathbf{P}_{o,p,q+1}^{l-1}$  with  $o + p + q + l = n$ .  $\mathbf{P}_{opq}^l$  is therefore the image of  $\mathbf{U}$  under the affine map from  $\triangle(\mathbf{R}, \mathbf{S}, \mathbf{T})$  onto  $\triangle(\mathbf{P}_{o+1,p,q}^{l-1}, \mathbf{P}_{o,p+1,q}^{l-1}, \mathbf{P}_{o,p,q+1}^{l-1})$ .
3.  $\mathbf{P}_{000}^n := \mathbf{P}(r, s, t)$  is the desired point on the surface.

Bézier triangles have the following properties:

- The triangular Bézier patch is coupled in an affine invariant way to the control net.

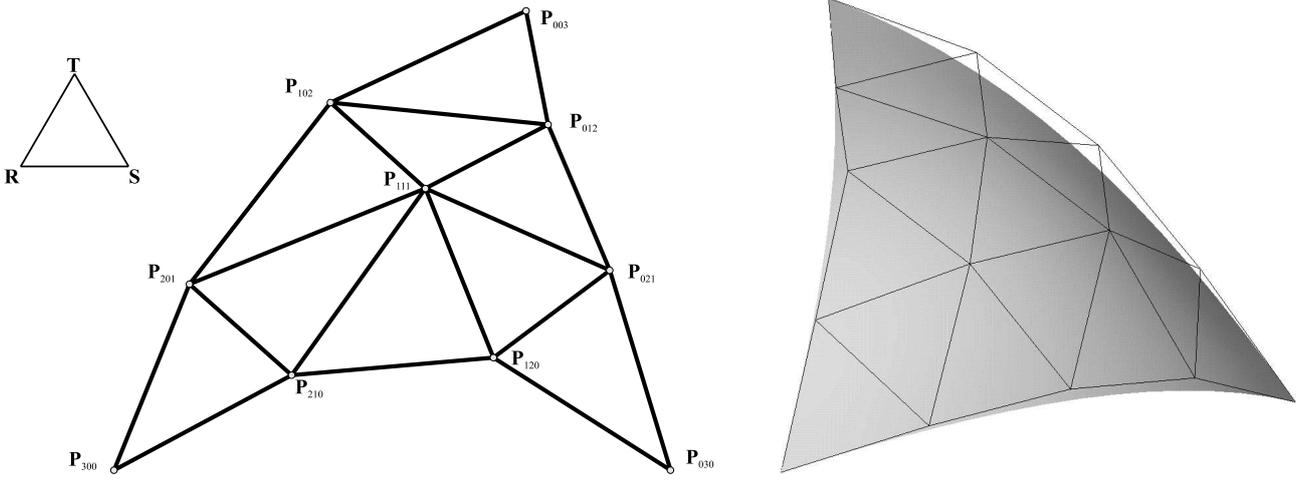


Figure B.2: To the left the triangular parameter domain is shown. In the middle a control net for a Bézier triangle of degree 3 and the enumeration of control points is shown. Right a Bézier triangle of degree 4 and its control net are shown.

- The Bézier triangle lies in the convex hull of the control net.
- The surface has a polynomial parametric representation of degree  $n$ , expressible with Bernstein polynomials as

$$\mathbf{P}(r, s, t) = \sum_{i+j+k=n} \mathbf{P}_{ijk} B_{ijk}^n(r, s, t) \quad (\text{B.1})$$

$$\text{with } B_{ijk}^n(r, s, t) = \frac{n!}{i!j!k!} r^i s^j t^k.$$

- End point interpolation:  $\mathbf{P}_{n00}$  is the point on the surface corresponding to the point  $\mathbf{R}$  in parameter space. The tangent plane at  $\mathbf{P}_{n00}$  contains the points  $\mathbf{P}_{n-1,1,0}$  and  $\mathbf{P}_{n-1,0,1}$ . Analogy applies for  $\mathbf{P}_{0n0}$  and  $\mathbf{P}_{00n}$ .
- The boundary curves of a Bézier triangle are Bézier curves. Their control polygons are the boundary polygons of the control net. The other control points of the surface are called ‘inner control points’.
- A Bézier triangle of degree  $n$  can be written as a Bézier triangle of degree  $n + 1$  with the Bézier points  $\mathbf{P}_{ijk}^{(1)}$ . This is known as degree elevation. The new points are:

$$\mathbf{P}_{ijk}^{(1)} = \frac{1}{n+1} \left( i\mathbf{P}_{i-1,j,k} + j\mathbf{P}_{i,j-1,k} + k\mathbf{P}_{i,j,k-1} \right), \quad i + j + k = n + 1$$

- The tangent plane at  $\mathbf{P}_{000}^n$  is defined by the three points  $\mathbf{P}_{100}^{n-1}, \mathbf{P}_{010}^{n-1}, \mathbf{P}_{001}^{n-1}$  in the construction of a surface with the de Casteljau algorithm.
- The directional derivative  $\mathbf{D}_v$  to the direction  $\mathbf{v}$  in parameter space is given by the affine image of  $\mathbf{v}$  in the triangle that is obtained in the last but one step of the de Casteljau algorithm (Fig.B.3, left).
- The de Casteljau algorithm has the following subdivision property: During the computation of  $\mathbf{P}(r, s, t)$  the control nets of the three subpatches to the parameter triangles  $\triangle(\mathbf{R}, \mathbf{S}, \mathbf{U})$ ,  $\triangle(\mathbf{U}, \mathbf{S}, \mathbf{T})$  and  $\triangle(\mathbf{R}, \mathbf{U}, \mathbf{T})$  are obtained. The control points are  $\mathbf{P}_{ijl}^l$  with  $i + j + l = n$  and analogously for the other two control nets. As an example, in Fig. B.3 (right) one of the new nets is  $\mathbf{P}_{200}, \mathbf{P}_{110}, \mathbf{P}_{020}, \mathbf{P}_{100}^1, \mathbf{P}_{010}^1, \mathbf{P}_{000}^2$ . The new control nets lie closer to the surface than the original one.

This is the first subdivision algorithm. By using the fundamental recursion ([Hoschek and Lasser, 1993]) the second and third subdivision algorithms can be found. This algorithms provide the control points for a Bézier curve and a Bézier triangle, respectively, on the given patch.

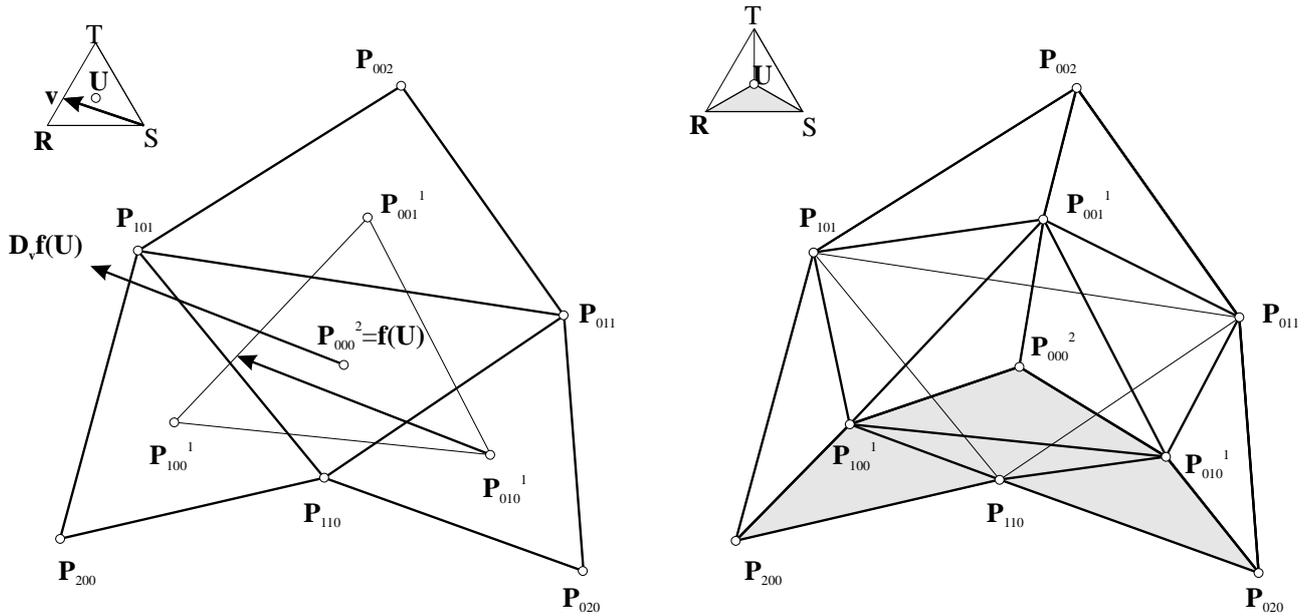


Figure B.3: A Bézier triangle of degree 2, Left: directional derivative with direction  $\mathbf{v}$  in parameter space, the directional derivative lies in the direction  $\mathbf{D}_v \mathbf{f}(\mathbf{U})$ . Right: with the de Casteljau algorithm three new control nets for the Bézier triangle are obtained. The parameter space  $\Delta(\mathbf{RSU})$  and the corresponding control net  $\mathbf{P}_{ij0}^1$  are shaded grey.

The subdivision algorithm mentioned above can be performed for more points in parameter space simultaneously. Choosing three points in the parameter space at the midpoints of each edge of the parameter triangle yields four new control nets describing the surface. Each of these control nets can be subdivided with the same points in parameter space again. This leads to a sequence of control nets that are fastly converging towards the surface (see [Hoschek and Lasser, 1993]). This is illustrated in Fig. B.4. If subdivision for Bézier triangles is mentioned, it has most often to be understood as subdivision based on the edge midpoints of the parameter triangle.

### Tangent planes and inner control points along the boundary curves

In this short section the construction of tangent planes along the boundary curves will be demonstrated. The tangent planes do only depend on the control points along the boundary curve and the control points in the next row. This is shown in Fig. B.5 for degree 3 along the boundary curve with the control points  $\mathbf{P}_{300}, \mathbf{P}_{210}, \mathbf{P}_{120}, \mathbf{P}_{030}$ . This curve  $\mathbf{k}_1(t)$  is parameterized by  $t \in [0, 1]$ . Let additionally  $\mathbf{Q}_1(t)$  be the Bézier curve to the control polygon  $\mathbf{P}_{201}, \mathbf{P}_{111}, \mathbf{P}_{021}$ . Then  $\mathbf{q}_1(t) = \mathbf{Q}_1(t) - \mathbf{k}_1(t)$  is a vector in the tangent plane in the point  $\mathbf{k}_1(t)$ . A second vector in the tangent plane is  $\dot{\mathbf{k}}_1(t) = \partial \mathbf{k}_1(t) / \partial t$ .

Alternatively, it can be said that the tangent plane in  $\mathbf{k}_1(t)$  is spanned by three points. The first point is  $\mathbf{Q}_1(t)$ . The other two points are the curve points to the parameter  $t$  of the Bézier curves to the control polygons  $\mathbf{P}_{300}, \mathbf{P}_{210}, \mathbf{P}_{120}$  and  $\mathbf{P}_{210}, \mathbf{P}_{120}, \mathbf{P}_{030}$ , respectively. These three curves have degree 2.

### Remarks

The theory on Bézier curves can be obtained from the above simply by reduction of the parameter space from  $\mathbb{R}^2$  to  $\mathbb{R}$ . The parameter triple reduces to a parameter pair, but instead of the pair  $s$  and  $t$  the values  $(1-t)$  and  $t$  are used. The control net reduces to a control polygon. For a curve of degree  $n$  the number of vertices is

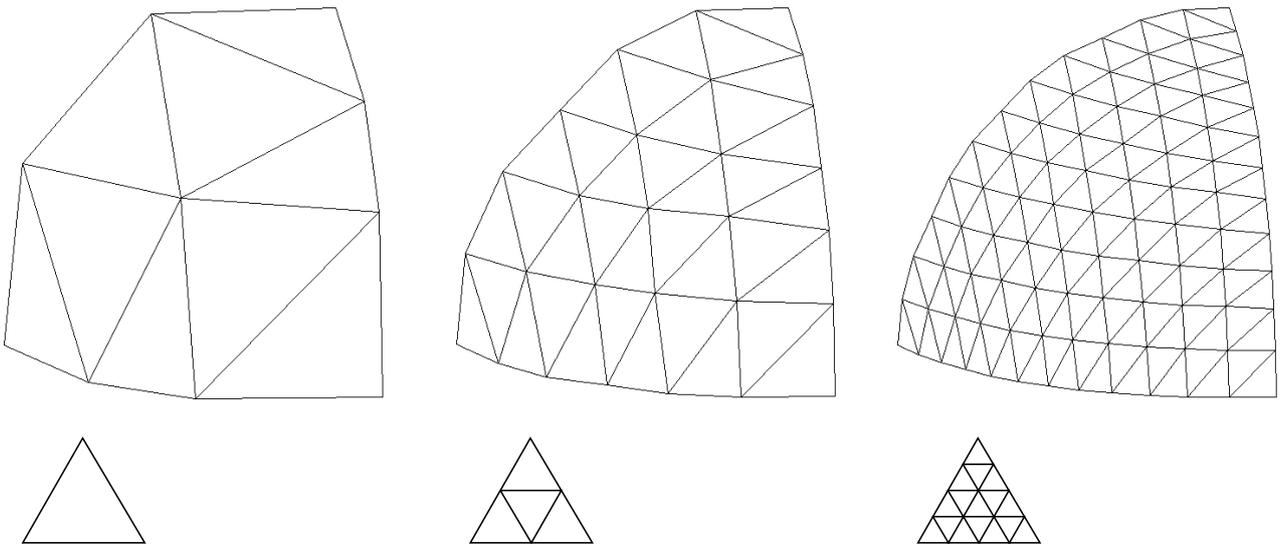


Figure B.4: The third subdivision algorithm is applied to a Bézier triangle (degree 3) based on the midpoints of the edges. The left image shows the parameter domain and the control net, in the middle, the parameter domain has been subdivided into 4 triangles and 4 control nets describe the corresponding parts of the surface. This scheme has been repeated once more to obtain the right image.

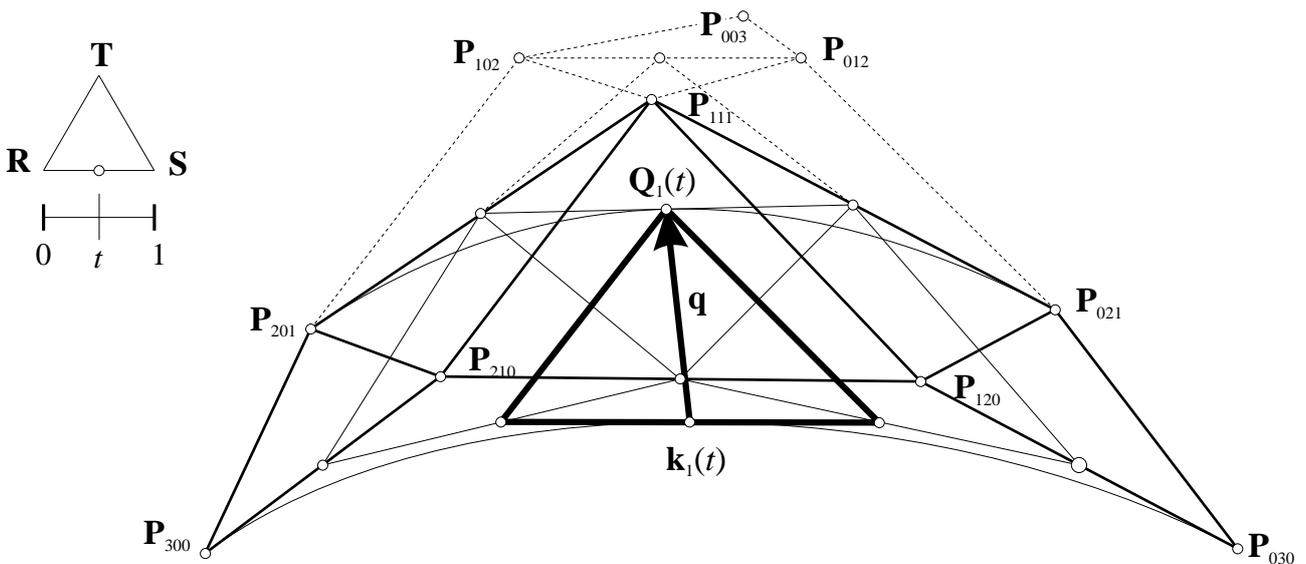


Figure B.5: De Casteljau algorithm for the construction of the tangent plane along the boundary curve.

$n + 1$ . Usually the points are numbered  $P_0, \dots, P_n$ . The parameter description with the Bernstein polynomials is:

$$\begin{aligned}
 \mathbf{P}^n(t) &= \sum_{i=0}^n B_i^n(t) \mathbf{P}_i \\
 \text{with} \\
 B_i^n(t) &= \binom{n}{i} t^i (1-t)^{n-i}
 \end{aligned} \tag{B.2}$$

Generally, the derivatives are:

$$\begin{aligned}\frac{d^r}{dt^r} \mathbf{P}^n(t) &= \frac{n!}{(n-r)!} \sum_{j=0}^{n-r} B_j^{n-r}(t) \Delta^r \mathbf{P}_j \\ \Delta^1 \mathbf{P}_i &= \mathbf{P}_{i+1} - \mathbf{P}_i \\ \Delta^r \mathbf{P}_j &:= \Delta^{r-1} \mathbf{P}_{j+1} - \Delta^{r-1} \mathbf{P}_j\end{aligned}$$

There are two different extensions from Bézier curves to Bézier surfaces. Above the Bézier triangles have been described, a different extension are the so-called tensor product Bézier surfaces. They are defined over a rectangular parameter domain. A description can be found in [Hoschek and Lasser, 1993].



# Appendix C

## Loop and Butterfly subdivision

In this section the subdivision coefficients and subdivision masks for two triangular subdivision schemes will be presented. These are the Loop scheme, which is approximating and the Modified Butterfly scheme, which is interpolating. There are different masks for the refinement of the boundary points and of the inner points. The 'boundary' masks depend only on the boundary points, so the boundary curves are not influenced by the points in the interior. Additionally, there are specific coefficients and masks for extraordinary (i.e. not regular) vertices. For an interpolating scheme there are only rules for the refinement of the odd (newly inserted) vertices, whereas for the approximating schemes there are different rules for the refinement of the odd and the even (inherited from the previous level) vertices. The masks show the weights which are used for the computation of a vertex. The co-ordinates are obtained as the weighted average from the vertices in the mask.

### C.1 Loop

The Loop scheme [Loop, 1987] is an approximating scheme. The coefficients and mask for even and odd vertices in the interior and on a boundary are given in Fig. C.1. A black vertex indicates a newly inserted one (odd vertex), the light grey vertices are inherited from the previous level (even). Squares indicate a vertex for which the co-ordinates are (re-)computed. In the subdivision rule for even interior vertices  $d$  denotes the valency of the vertex, and:

$$\alpha = \frac{1}{d} \left( \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{d} \right)^2 \right)$$

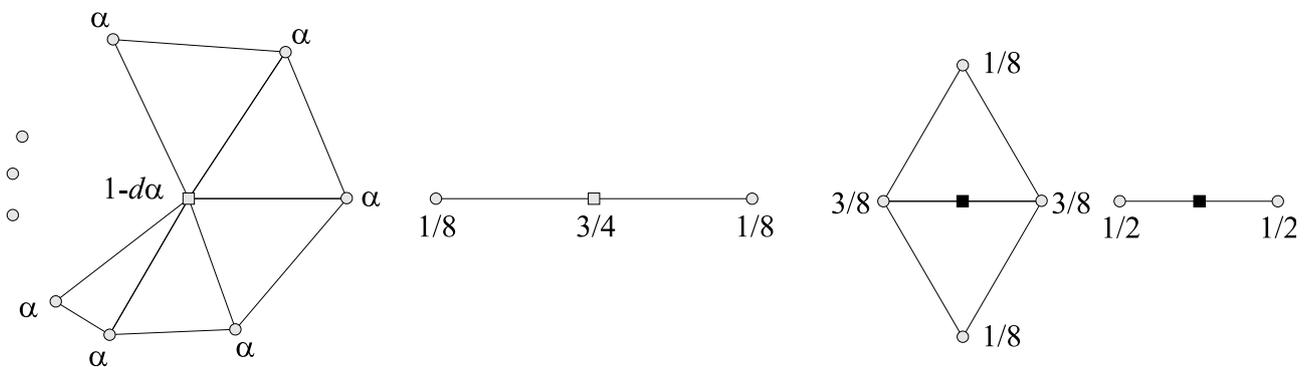


Figure C.1: Masks for Loop subdivision, from left to right: Mask for even interior vertices, for even boundary vertices, for odd interior vertices, and for odd boundary vertices, respectively.

There are other choices for  $\alpha$ , which also produce a C1 surface ([Zorin and Schröder, 2000]).

In all cases the coefficients sum up to one, which means that the new points are affine combinations of the old ones. The rules for the boundary produce splines of degree three. The surface is C1 everywhere and C2 at regular vertices and composed of polynomial quartic patches.

## C.2 Modified Butterfly

The Modified Butterfly subdivision scheme is an extension of the Butterfly scheme which produced interpolating C1 surfaces over regular triangulations<sup>1</sup>. It has been modified in order to achieve interpolating C1 surfaces over any triangulation. The masks for the subdivision are given in Fig. C.2, the color and shape convention is the same as above for the Loop scheme. As the scheme is interpolating, only rules for the computation of odd vertices are necessary.

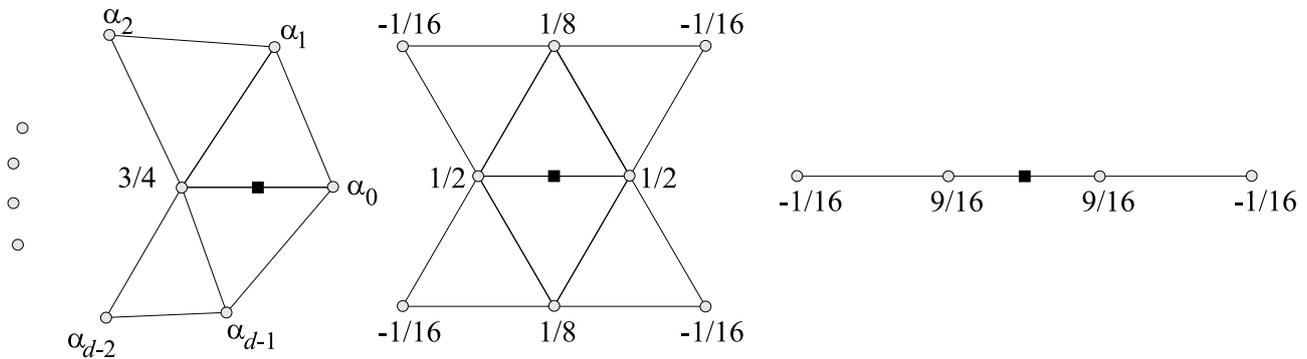


Figure C.2: Masks for odd vertices with the Modified Butterfly scheme, from left to right: Mask for a vertex next to an extraordinary vertex, mask for a vertex adjacent to two regular vertices, mask for a boundary vertex.

If an odd vertex is adjacent to two regular vertices, the rule in the middle of Fig. C.2 is applied. For boundary vertices the rule is shown right.

If an odd vertex is next to one extraordinary vertex, the rule as shown in the left part of Fig. C.2 is applied. The extraordinary vertex is given the weight 0.75, and the neighbors of the extraordinary vertex the weights  $\alpha_i$ . The choice of the weights depend on the valency of the vertex. It holds:

$$\begin{aligned}
 d = 3 \quad \alpha_0 &= \frac{5}{12}, \quad \alpha_1 = \alpha_2 = -\frac{1}{12} \\
 d = 4 \quad \alpha_0 &= \frac{3}{8}, \quad \alpha_2 = -\frac{1}{8}, \quad \alpha_1 = \alpha_3 = 0 \\
 d \geq 5 \quad \frac{1}{d} &\left( \frac{1}{4} + \cos \frac{2i\pi}{d} + \frac{1}{2} \cos \frac{4i\pi}{d} \right), \quad i = 0, \dots, d-1
 \end{aligned}$$

If an odd vertex is adjacent to two extraordinary vertices, the rules for extraordinary vertices are applied for each one and the result is averaged.

The Modified Butterfly scheme can be seen as an extension of the interpolating subdivision rule for curves, shown right in Fig. C.2, to surfaces. The weighting for this curve scheme can be more general, with weights  $1/2 + w$  for the interior vertices and  $-w$  for the exterior vertices [Dyn et al., 1987]. If the vertices are numbered  $\mathbf{P}_{-2}, \mathbf{P}_{-1}, \mathbf{P}_1, \mathbf{P}_2$  from left to right, the newly inserted vertex can be obtained with the equivalent formula  $(\mathbf{P}_{-1} + \mathbf{P}_1)/2 + w(\mathbf{P}_{-1} - \mathbf{P}_{-2}) + w(\mathbf{P}_1 - \mathbf{P}_2)$ . This scheme converges toward a C1 limit curve, if  $0 < w < 1/8$ . For  $w = 0$  the linear spline is achieved. The scheme is known as Four point interpolatory subdivision scheme.

<sup>1</sup>In a regular triangulation every vertex is regular, i.e. has valence 6.

## Appendix D

# “Demo”-Example with interpolating and approximating subdivision schemes

This example consists of photogrammetric profile measurements and breaklines<sup>1</sup>. It has been used in this work for testing and demonstrating different algorithms.

The following figures are enlargements of Fig. 5.9 in Sec. 5.3.5 (p. 66) and Fig. 5.11 in Sec. 5.3.6 (p. 67).

The different determination of the approximating local surfaces and their impact on the subdivision surface is shown. Fig. D.1 shows the original data set, and in comparison Fig. D.2 shows the same data after application of kriging (linear prediction) for the removal of random noise.

Fig. D.3 shows the subdivision surface based on a interpolating/approximating scheme. In comparison, Fig. D.4 shows the surface obtained with the same subdivision scheme, but the filtered points were used as the input triangulation.

Fig. D.5 shows the subdivision surface for a strictly interpolating scheme and Fig. D.6 shows a purely approximating scheme. Both examples are based on the unfiltered original data.

The impact of different methods of computing the odd point from the local surfaces in the adjacent even points (i.e. ‘averaging’) is demonstrated in Figs. D.7 to D.10. In the first figure the original triangulation (unfiltered) is shown, which is the basis for the computation of the subdivision surface in the following figures.

The interpolating/approximating scheme was used with paraboloids computed from the star shaped neighborhood as local surfaces. The other figures show the surfaces obtained with three different methods of averaging. The points on the paraboloids to the mid edge point in the parameter space can be seen in Fig. D.8. In Fig. D.9 the two points which are averaged are obtained by intersection the paraboloids with one line. This line goes through the edge midpoint and its direction is the mean of the normal vectors of the reference planes (projection planes). In Fig. D.10 the paraboloids are intersection with a similar line, its direction is the mean of the surface normals in the edge end points.

---

<sup>1</sup>It is in use at the Institute of Photogrammetry and Remote Sensing, Vienna University of Technology for test and demonstration purposes and therefore known as the Demo example.

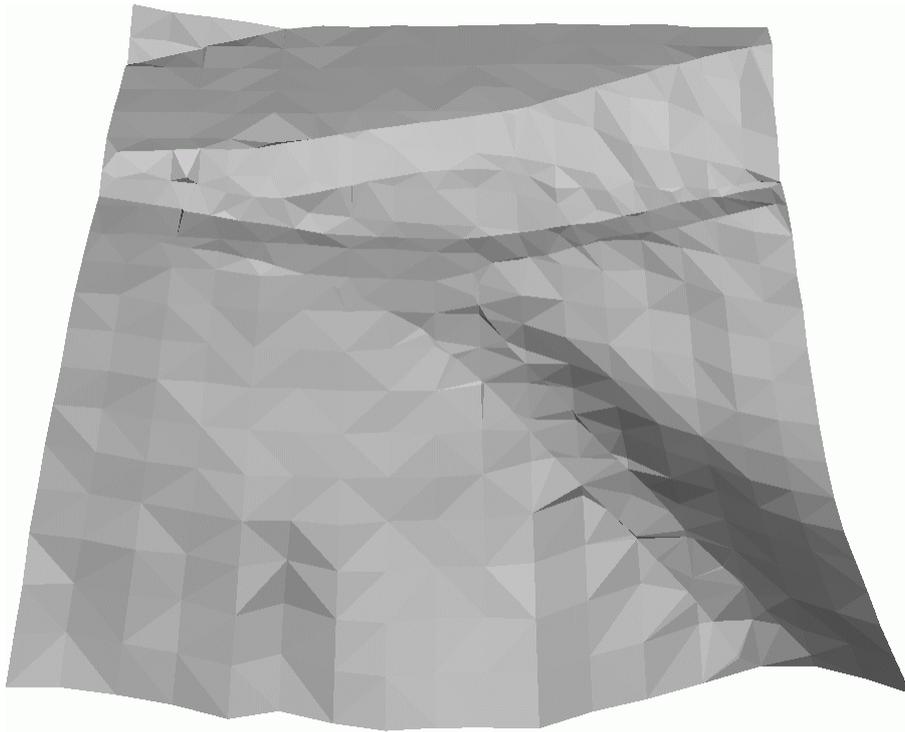


Figure D.1: Original data

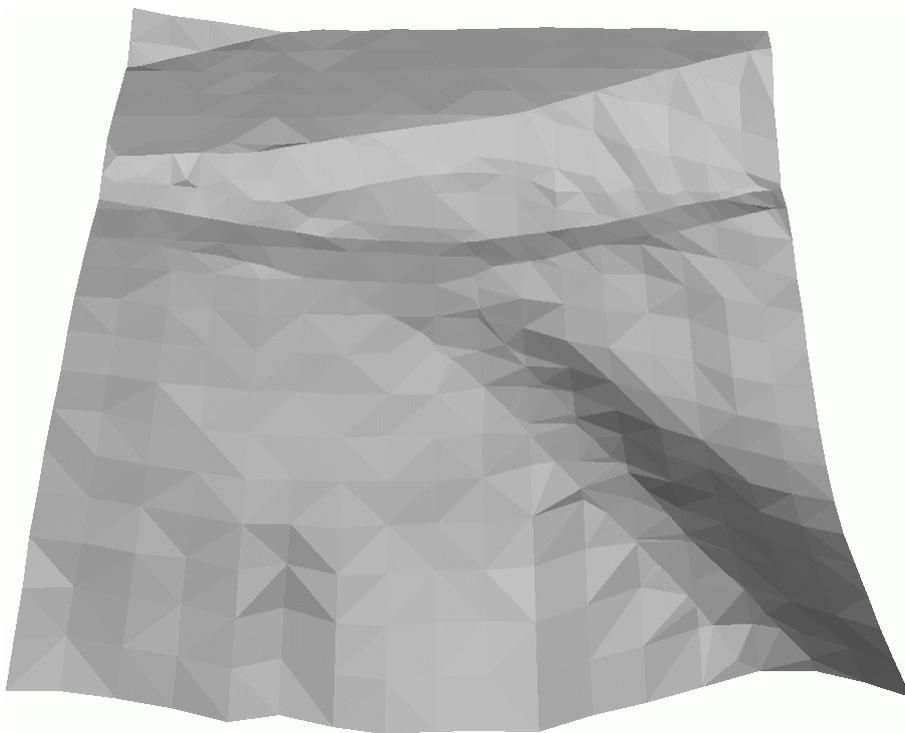


Figure D.2: Filtered original data

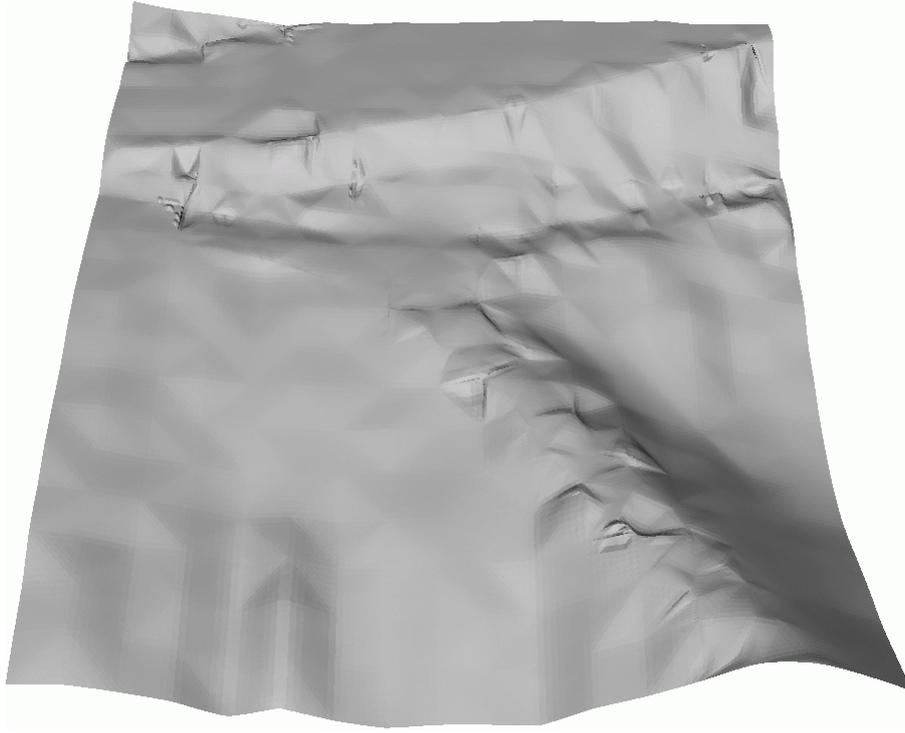


Figure D.3: Subdivision surface with interpolating/approximating scheme



Figure D.4: Subdivision surface with interpolating/approximating scheme based on the filtered version of the original data

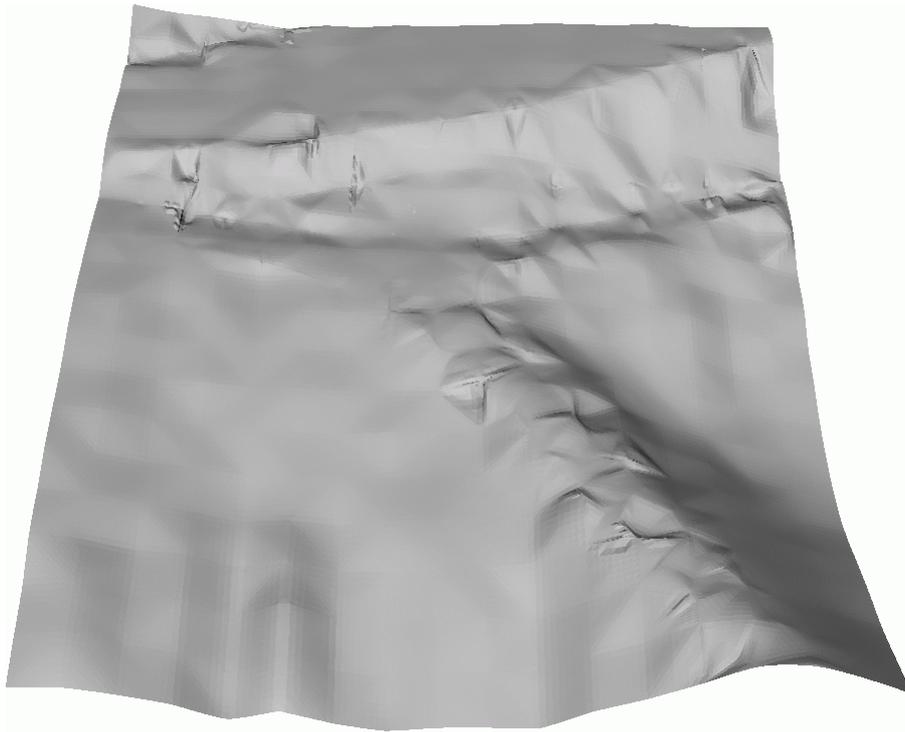


Figure D.5: Subdivision surface with strictly interpolating scheme



Figure D.6: Subdivision surface with purely approximating scheme

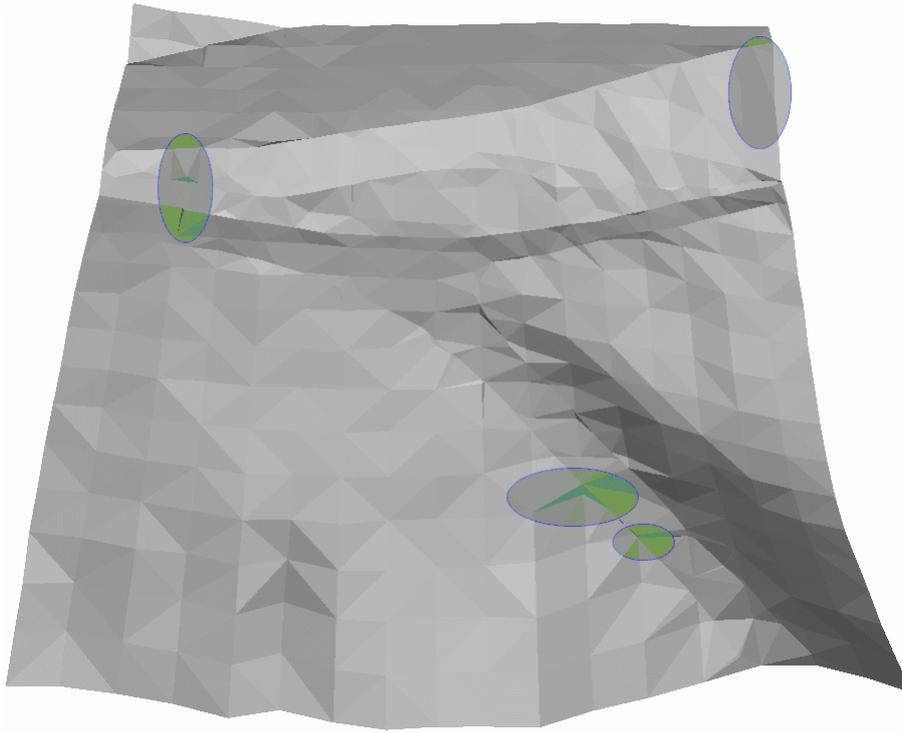


Figure D.7: Original triangulation with indication of positions where differences between the different averaging method can be found.

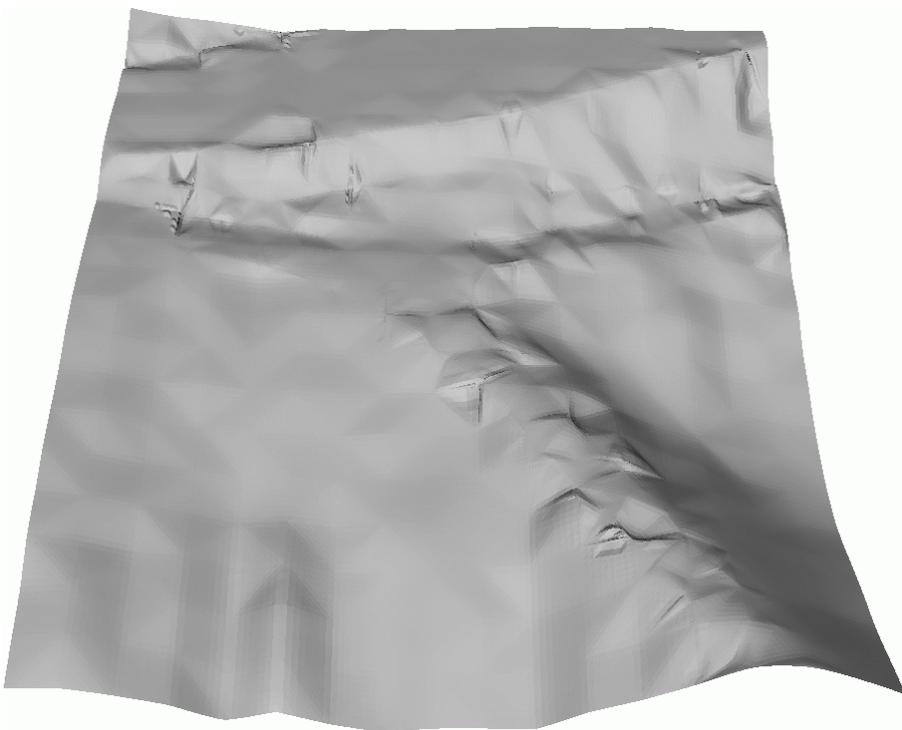


Figure D.8: Subdivision surface with interpolating/approximating scheme with averaging based on the mid-points in parameter space.

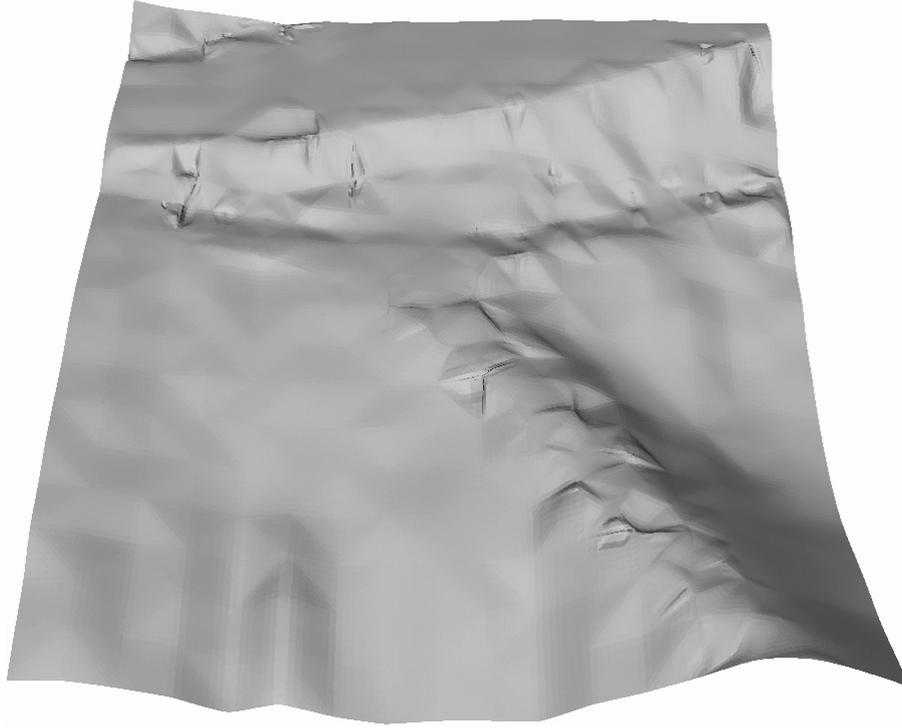


Figure D.9: Subdivision surface with interpolating/approximating scheme with averaging of intersections between local surfaces and line through edge midpoint with direction of the mean of local surface normals in the edge points.

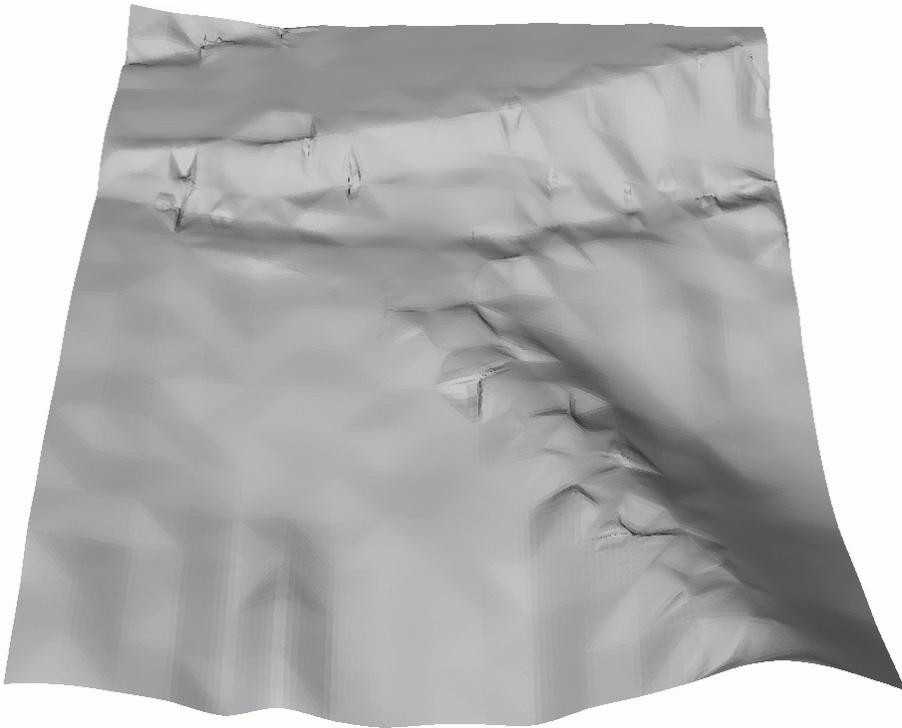


Figure D.10: Subdivision surface with interpolating/approximating scheme with averaging of intersections between local surfaces and line through edge midpoint with direction of the mean of the reference plane normals.

## Appendix E

### Color pages

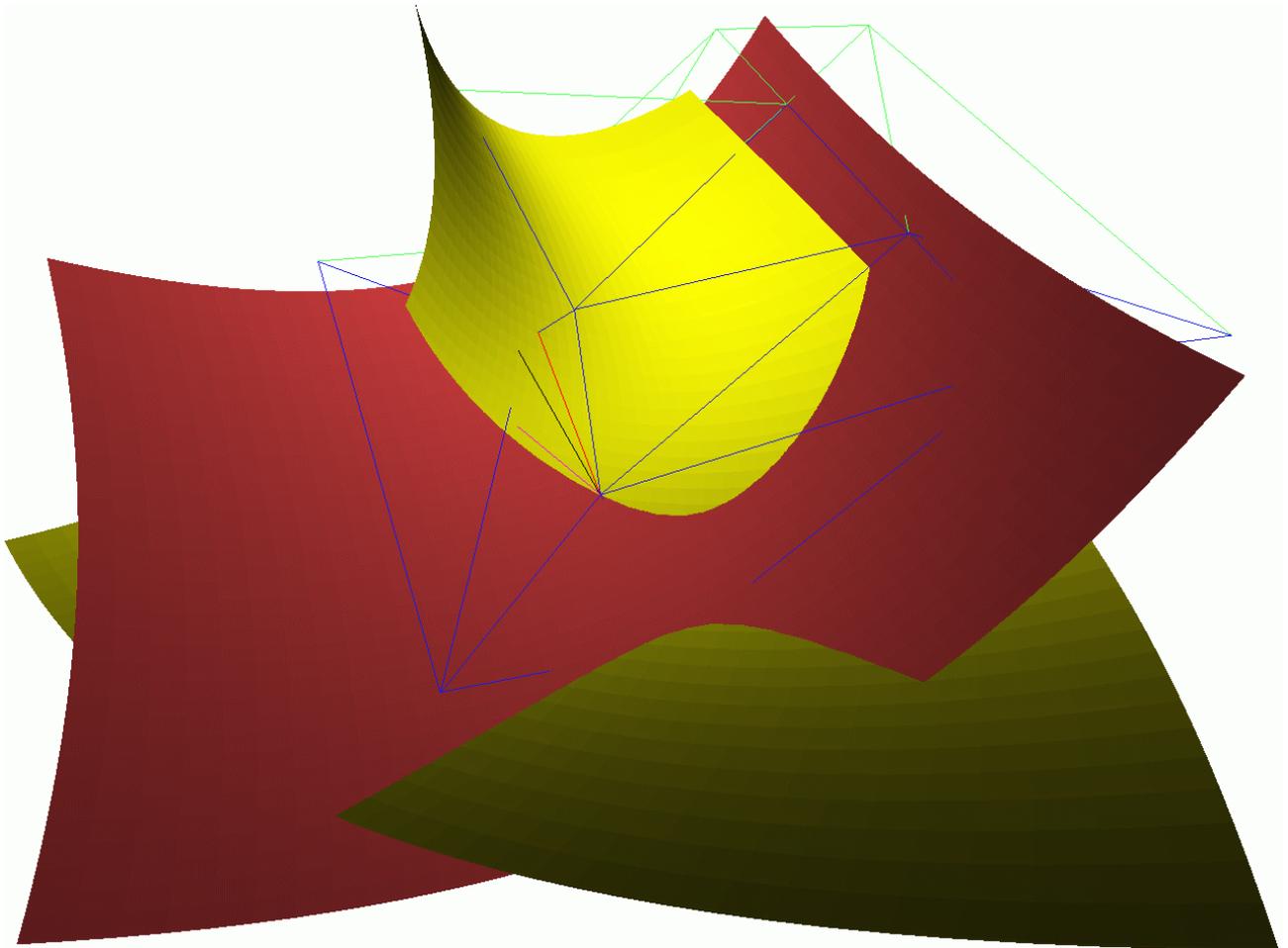


Figure E.1: Adjusting second order polynomials through one point of the overhang. The blue edges are those which belong to the star shaped neighborhood of the center vertex, the green ones do not. The red paraboloid is the graph of a bivariate function which has as parameter domain an adjusting plane (black normal vector) through the points. The yellow surface is a vector-valued polynomial and describes the overall shape of the surface (i.e. the overhang) better.

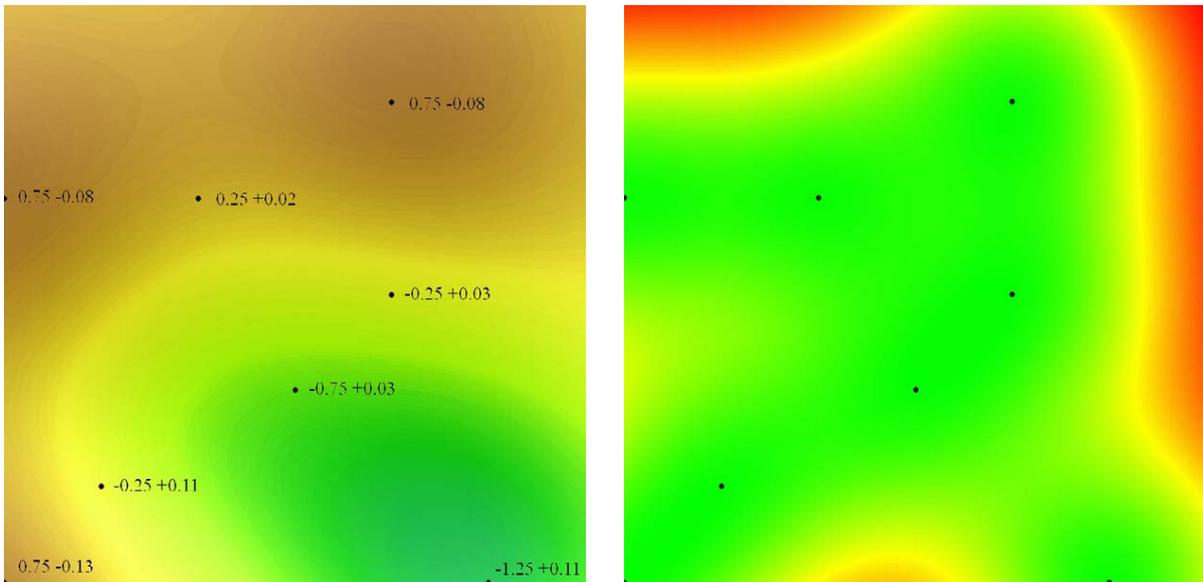


Figure E.2: Terrain model and accuracy model. Terrain heights reach from -1.25m to 0.75m (green to brown), next to the observed heights the residuals are shown. The accuracies range from  $\pm 3\text{cm}$  to  $\pm 44\text{cm}$  (green to red).

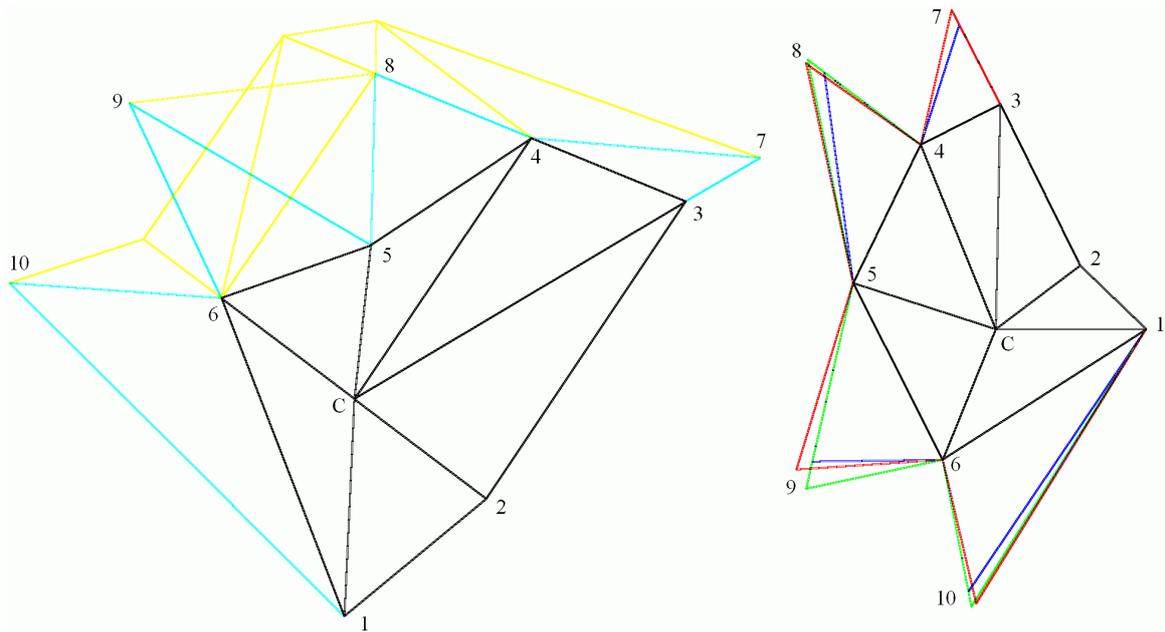


Figure E.3: Parameterization of the star-shaped neighborhood of point C. Left: perspective view of the given data, yellow edges do not belong to the star shaped neighborhood of C. Right: parameterization by three different methods, the inner triangles are drawn in black, those of generation one and a half have different colors according to the method. Green: edge length, blue: similar triangles, red: foot of altitude.

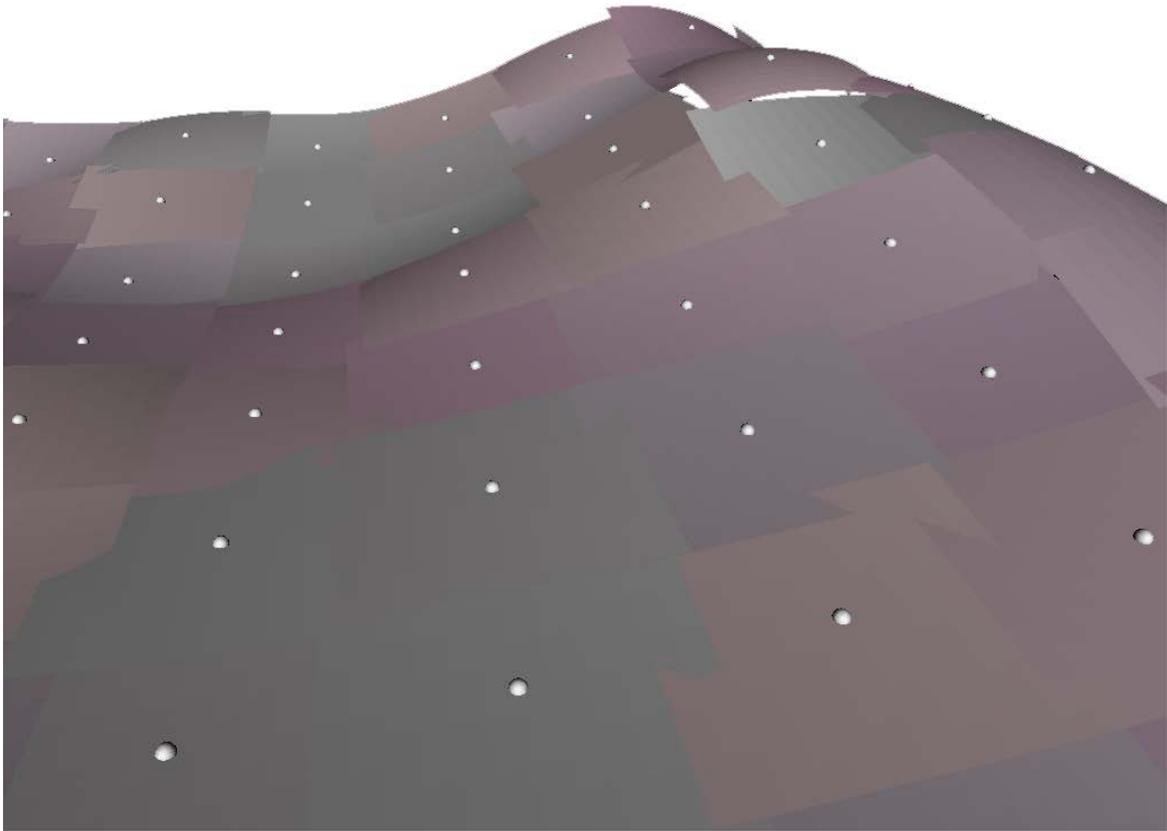


Figure E.4: Estimated paraboloids for surface points. Each paraboloid has been given a color in order to make the paraboloids distinguishable.

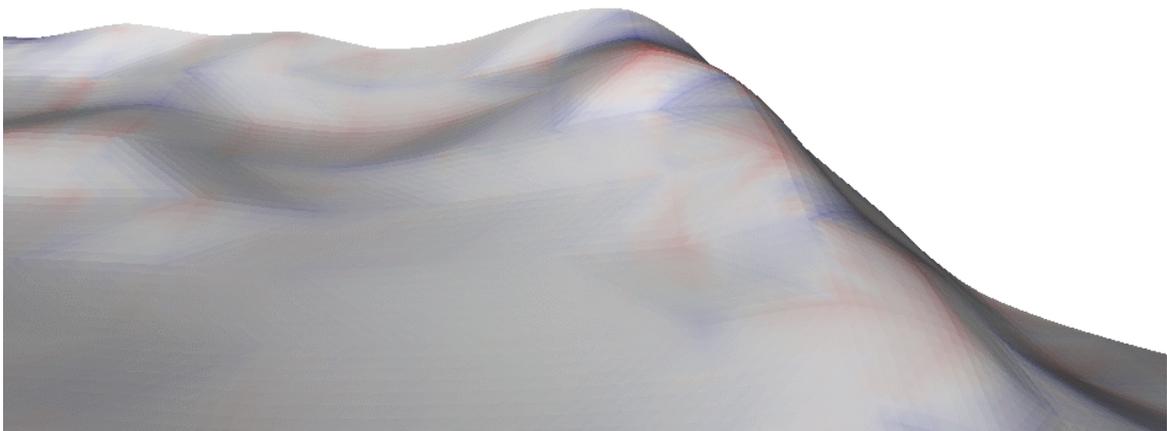


Figure E.5: Detail of the “Albis” surface computed with a subdivision approach. Red and blue areas stand for positive and negative, respectively, gaussian curvature. A similar perspective on the same terrain is also shown in Fig. 5.7.

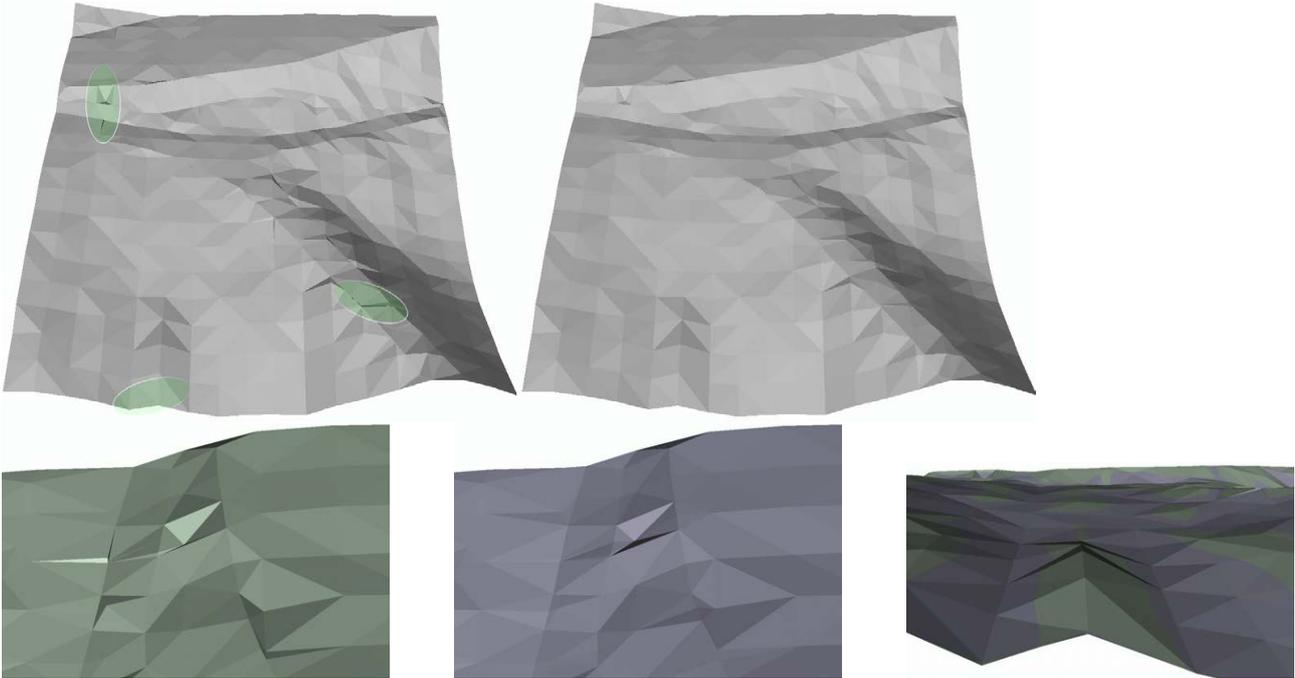


Figure E.6: Filtering of random measurement errors in a triangulation. Upper left shows the original triangulation, Upper right the filtered version. The first two images in the lower row show the uppermost indicated area viewed from the right hand side. The last image shows an overlay of both triangulations at the lowest indicated area. The green triangles belong to the original triangulation, the blue ones to the filtered one.

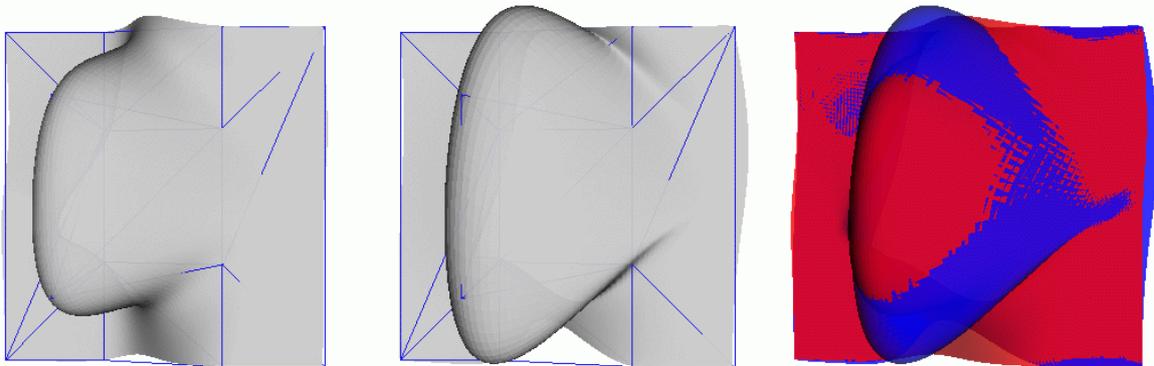


Figure E.7: Left and middle: two different models describing an overhang, viewed from above. Right, the two models are superimposed. How to define the difference between those two models? The two models have been obtained by computing a vector-valued surface, stored in a grid model, over the parameterizations of the overhang triangulation shown in Fig. 3.4.

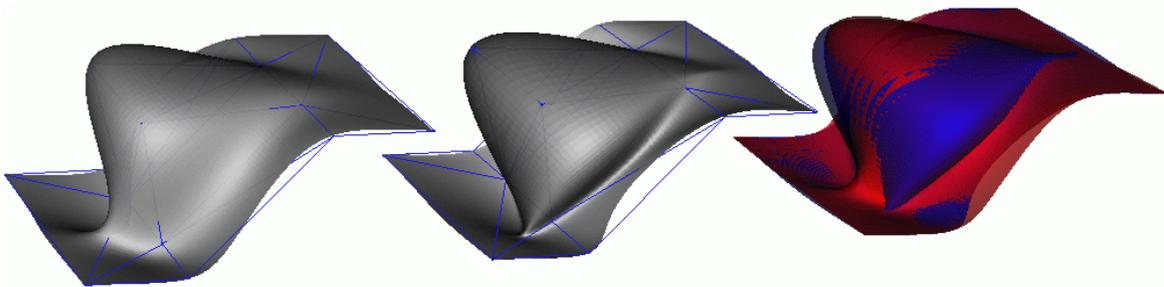


Figure E.8: A different view on the data shown in Fig. 7.2.

# Curriculum Vitae of Norbert Pfeifer

---

birth	27. May 1971 in Vienna, Austria
citizenship	Austrian

---

1977-1981	primary school in Vienna (Volksschule)
1981-1989	grammar school with emphasis on nature sciences (Bundesrealgymnasium Wien XIX)
06/1989	school leaving exam (Matura)
1989-1997	study of surveying at the Vienna University of Technology
07-09/1994	stay abroad at Melbourne University, Department of Geomatics (with a grant from the Austrian Ministry for Science and Research)
1995, 1996	diploma thesis at the Institute of Photogrammetry and Remote Sensing: „3-dimensionale Oberflächenmodelle auf Basis einer Triangulierung – Flächenrepräsentation“ (“3-dimensional surface models on the basis of a triangular mesh – surface reconstruction”)
09/1997	diploma examination with excellence – graduated „Diplomingenieur“
10/2002	Ph.D. examination with excellence – graduated „Doctor technicae“

---