

# Lokale Lagrange-Interpolation mit Splineoberflächen

Inauguraldissertation  
zur Erlangung des akademischen Grades  
eines Doktors der Naturwissenschaften  
der Universität Mannheim

vorgelegt von

Diplom-Handelslehrer Andreas Dinh  
aus Ludwigsburg

Mannheim, 2006

Dekan: Professor Dr. Matthias Krause, Universität Mannheim  
Referent: Professor Dr. Günther Nürnberger, Universität Mannheim  
Korreferent: Professor Dr. Gabriele Steidl, Universität Mannheim

Tag der mündlichen Prüfung: 13. Dezember 2006

# Inhaltsverzeichnis

<b>Einleitung</b>	<b>4</b>
<b>1 Grundlagen</b>	<b>10</b>
1.1 Splines auf Triangulierungen und Quadrangulierungen . . . . .	10
1.2 Interpolation mit Splines . . . . .	13
1.3 Bézier-Bernstein-Darstellung . . . . .	15
1.4 Richtungsableitungen und Bézier-Bernstein-Koeffizienten . . . . .	18
1.5 $C^r$ -Bedingungen . . . . .	19
1.6 Approximationsordnung von Splineräumen und Interpolation . . . . .	21
1.7 Anwendungen des Algorithmus von de Casteljau . . . . .	22
<b>2 Splines auf Quadrangulierungen</b>	<b>27</b>
2.1 Grundlagen . . . . .	29
2.2 Algorithmen für Quadrangulierungen . . . . .	31
2.2.1 Konstruktion . . . . .	31
2.2.2 Färbung . . . . .	35
2.2.3 Klassen von Quadrangulierungen . . . . .	41
2.3 Interpolation und Färbung . . . . .	46
2.3.1 Interpolation mit kubischen Splines . . . . .	47
2.3.2 Interpolation mit Splines höheren Grades . . . . .	49
2.3.3 Interpolation bei spezieller Färbung . . . . .	72
2.4 Interpolation und Prioritätsprinzip . . . . .	76
2.5 Vergleich der Interpolationsmethoden . . . . .	82

---

<b>3</b>	<b>3D-Splineoberflächen auf Quadrangulierungen</b>	<b>84</b>
3.1	Grundlagen . . . . .	85
3.2	Berechnung des stetigen Interpolanten . . . . .	92
3.3	Konstruktion einer groben Quadrangulierung im Raum . . . . .	95
3.4	Modifikation von Bézier-Bernstein-Koeffizienten . . . . .	99
3.4.1	Färbungs- und Prioritätsprinzipien . . . . .	100
3.4.2	Glättung des Interpolanten . . . . .	103
3.5	Eigenschaften der Methode . . . . .	106
3.5.1	Lokalität . . . . .	106
3.5.2	Optimale Approximationsordnung . . . . .	108
3.5.3	Datenaufwand der geglätteten Interpolanten . . . . .	116
<b>4</b>	<b>3D-Splineoberflächen auf projizierten Triangulierungen</b>	<b>117</b>
4.1	Bestimmung projizierbarer Teiltriangulierungen . . . . .	118
4.1.1	Zerlegung von 3D-Triangulierungen . . . . .	118
4.1.2	Geometrische Phänomene . . . . .	119
4.1.3	Erweiterte Teilalgorithmen . . . . .	126
4.1.4	Implementierung der Methode . . . . .	133
4.1.5	Algorithmische Tests . . . . .	140
4.2	Bestimmung des Interpolanten . . . . .	142
4.2.1	Berechnung des stetigen Interpolanten und Unterteilung der Dreiecke . . . . .	142
4.2.2	Glättung des Interpolanten . . . . .	143
4.3	Eigenschaften der Methode . . . . .	147
4.3.1	Differenzierbarkeit der geglätteten Interpolanten . . . . .	147
4.3.2	Lokalität und optimale Approximationsordnung . . . . .	148

# Einleitung

Bivariate Splinefunktionen sind von großer Bedeutung hinsichtlich der Konstruktion und Rekonstruktion von Oberflächen, wie sie in verschiedensten wissenschaftlichen Disziplinen Anwendung finden, beispielsweise in der Medizin, im Maschinenbau und im Computer Aided Design.

Obwohl insbesondere in den letzten Jahren die Fortschritte bei der Approximation mit bivariaten Splines sehr groß waren, sind bis heute zahlreiche fundamentale Fragen in Theorie und Anwendung unbeantwortet. Dies liegt hauptsächlich an der komplexen Struktur der zugrunde liegenden Räume. Für eine Triangulierung  $\Delta$  eines polygonalen Gebiets  $\Omega$  in der Ebene ist der Raum der bivariaten Splines von Grad  $q$  und Differenzierbarkeitsordnung  $r$  wie folgt definiert:

$$\mathcal{S}_q^r(\Delta) := \{s \in \mathcal{C}^r(\Omega) : s|_T \in \mathcal{P}_q^2 \text{ für alle } T \in \Delta\}$$

Hierbei bezeichnet  $\mathcal{P}_q^2 := \text{span} \{x^i y^j : 0 \leq i + j \leq q\}$  den Raum der bivariaten Polynome mit totalem Grad  $q$ . In dieser Arbeit untersuchen wir solche Splineräume und entwickeln zugehörige Interpolationsmethoden.

Ein wichtiges Verfahren für die Konstruktion von Oberflächen ist die Interpolation. Eine Menge  $\mathcal{L} = \{z_1, \dots, z_d\}$  von  $d = \dim \mathcal{S}_q^r(\Delta)$  Punkten aus  $\Omega$  heißt Lagrange-Interpolationsmenge für  $\mathcal{S}_q^r(\Delta)$ , falls zu jeder stetigen Funktion  $f : \Omega \rightarrow \mathbb{R}$  ein eindeutiger Spline  $s \in \mathcal{S}_q^r(\Delta)$  existiert, der mit  $f$  auf diesen Punkten übereinstimmt, also die folgenden Interpolationsbedingungen erfüllt:

$$s(z_i) = f(z_i) \text{ für alle } i = 1, \dots, d.$$

Falls neben Funktionswerten außerdem partielle Ableitungen an manchen der Stellen  $z_i$  interpoliert werden, dann heißt die entsprechende Menge  $\mathcal{H}$  der insgesamt  $d = \dim \mathcal{S}_q^r(\Delta)$  Interpolationsbedingungen Hermite-Interpolationsmenge. Eine wichtige Eigenschaft von Interpolationsverfahren ist die Lokalität: Eine Methode heißt lokal, falls sich bei einer Änderung eines Interpolationswertes  $f(z_i)$  der Interpolant nur in der Nähe von  $z_i$  ändert. Methode und Interpolanten heißen stabil, falls, grob gesprochen, eine kleine Veränderung eines Interpolationswerts auch nur zu einer kleinen Änderung des Interpolanten führt.

Ein weiteres wichtiges Kriterium für die approximativen Eigenschaften eines Interpolationsverfahrens und des zugehörigen Splineraums  $\mathcal{S}_q^r(\Delta)$  ist die Approximationsordnung. Unter der Approximationsordnung eines Raumes  $\mathcal{S}_q^r(\Delta)$  versteht man die größte Zahl  $k \in \mathbb{N}$ , für welche eine lediglich von  $f$ ,  $q$  und dem kleinsten Winkel  $\alpha$  von  $\Delta$ , nicht aber vom größten Dreiecksumkreis  $h$  in  $\Delta$  abhängende Konstante  $K > 0$  existiert, sodass die Bedingung

$$\text{dist}(f, \mathcal{S}_q^r(\Delta)) := \inf \{ \|f - s\| : s \in \mathcal{S}_q^r(\Delta) \} \leq K \cdot h^k$$

erfüllt ist. Entsprechend wird bei der Approximationsordnung eines Interpolationsverfahrens der Approximationsfehler  $|f - If|$  des zu  $f$  gehörenden Interpolanten  $If$  anstelle des minimalen Fehlers  $\text{dist}(f, \mathcal{S}_q^r(\Delta))$  der Funktionen im Raum betrachtet. Die Approximationsordnung ist umso höher, je schneller bei einer geeigneten Verfeinerung von  $\Delta$  der Approximationsfehler gegen Null konvergiert. Außer von  $q$  und  $r$  ist die Approximationsordnung insbesondere abhängig von der Lokalität der Interpolationsmethode. Nach Ciarlet und Raviart [13] ist  $q + 1$  eine obere Schranke für die Approximationsordnung und daher optimal. De Boor und Jia [26] zeigten, dass nicht jedes Verfahren optimale Approximationsordnung besitzt.

Aufgrund ihrer hohen Komplexität ist für manche Splineräume eine Reihe fundamentaler Fragen, wie beispielsweise die Bestimmung der Dimension oder die Konstruktion von Interpolationsmengen, derzeit nicht vollständig beantwortet. Eine mögliche Vorgehensweise besteht darin, sich auf Triangulierungen spezieller Geometrie zu beschränken. Beispiele hierfür sind die regelmäßigen  $\Delta_1$ - und  $\Delta_2$ -Triangulierungen, siehe [52–54, 64, 65, 76, 82, 83], und auch Triangulierungen geschlossener ineinander geschachtelter Polygonzüge, siehe [18, 19, 21]. Ein anderer Ansatz ist es, jedes Dreieck der betrachteten Triangulierung in mehrere bestimmte Subdreiecke zu unterteilen, siehe beispielsweise [4, 5, 11, 35, 50, 51, 56, 58, 75, 77, 86].

Eine später entwickelte Vorgehensweise ist es, die Triangulierung in verschiedene Klassen zu zerlegen und die Eigenschaften dieser Klassen für die Konstruktion von Interpolationsmengen zu verwenden. So entwickelten Nürnberger und Zeilfelder [72] eine Methode zur lokalen Lagrange-Interpolation mit bivariaten  $C^1$ -Splines vom Grad  $\geq 3$  auf beliebigen Triangulierungen, wobei die zugrunde liegenden Dreiecke schwarz und weiß gefärbt werden. Auf diese Weise wird nur etwa die Hälfte der Dreiecke unterteilt, und es ergeben sich Splineräume von geringerer Dimension als bei der Methode von Clough und Tocher [16], bei welcher jedes Dreieck unterteilt wird. Die Interpolanten sind lokal bestimmbar und besitzen optimale Approximationsordnung. Ein ähnliches Interpolationsverfahren mit gefärbten Zerlegungen entwickelten Nürnberger, Schumaker und Zeilfelder [68]. Die zum Splineraum gehörende Triangulierung erhält man dabei durch Hinzufügen von Diagonalen zu den Vierecken einer beliebigen strikt konvexen Quadrangulierung.

Gegenüber dieser großen Zahl an Veröffentlichungen über bivariate Splines ist die Interpolation mit 3D-Splineoberflächen ein aktuelles und bislang wenig erforschtes Gebiet. Aufgrund der sehr komplexen Struktur dieser Splinefunktionen und der

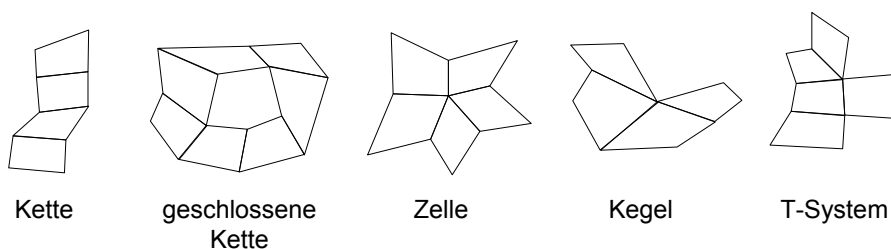


Abbildung 1: Systeme von Vierecken

entsprechenden Differenzierbarkeitsbedingungen (vgl. Farin [36]) existiert zur Zeit lediglich eine einzige Interpolationsmethode. Bei dieser von Kohlmüller, Nürnberger und Zeilfelder [47, 48] entwickelten Methode für kubische Interpolanten ergibt sich Differenzierbarkeit an den Eckpunkten und den Kantenmittelpunkten auf beliebigen Triangulierungen im Raum.

Im Einzelnen ist die Arbeit wie folgt gegliedert: In Kapitel 1 stellen wir wesentliche Grundlagen der Theorie bivariater Splines vor. In diesem Zusammenhang geben wir eine Einführung in die Bézier-Bernstein-Techniken. Dabei handelt es sich um Methoden, welche einen einfachen Umgang mit Stetigkeits- und Differenzierbarkeitsbedingungen und darüber hinaus eine effiziente Berechnung der Polynomstücke ermöglichen. Entwickelt wurden diese Verfahren unter anderem von Bézier [9], de Boor [22], de Casteljau [28] sowie Farin [36]. Darauf aufbauend formulieren wir den Differenzierbarkeitsbegriff in Bézier-Bernstein-Schreibweise. Dabei existiert nach de Boor [22] und Farin [36] ein bedeutender Zusammenhang zwischen den partiellen Ableitungen und den Bézier-Bernstein-Koeffizienten eines bivariaten Polynoms. Weiterhin beschreiben wir das Konzept der minimal bestimmenden Mengen. Die Anzahl der Elemente einer minimal bestimmenden Menge ergibt die Dimension des zugrunde liegenden Splineriums. Schließlich behandeln wir wesentliche Methoden, wie sie für die computertechnische Umsetzung von Interpolationsalgorithmen von Bedeutung sind. Eine wesentliche Rolle spielt hierbei der Algorithmus von de Casteljau [27].

In Kapitel 2 untersuchen wir bivariate  $C^1$ -Splines auf Quadrangulierungen. Nürnberger, Schumaker und Zeilfelder [68] beschreiben Methoden der lokalen Lagrange-Interpolation mit solchen Splines, basierend auf dem Raum  $\mathcal{S}_3^1(\diamond)$  auf triangulierten beliebigen strikt konvexen Quadrangulierungen  $\diamond$ . Dabei zeigt sich, dass die Effizienz des Verfahrens abnimmt, wenn bei der Färbung lange Ketten von benachbarten Vierecken gleicher Farbe entstehen und dass sich schwächere Lokalitätseigenschaften ergeben, wenn geschlossene Ketten auftreten, siehe Abbildung 1. Während sich beliebige Triangulierungen mit Hilfe eines einfachen, von Nürnberger und Zeilfelder [72] entwickelten Verfahrens so schwarzweiß färben lassen, dass gleichfarbige Dreiecke maximal in benachbarten Paaren auftreten, zeigt die Erfahrung, dass das entsprechende Problem für Quadrangulierungen äußerst komplex ist. Abbildung 2.1 (rechts) auf Seite 28 zeigt, dass bereits bei manchen sehr kleinen Quadrangulierungen eine

Färbung mit maximaler Kettenlänge zwei nicht möglich ist. So existiert in der Literatur bislang kein Färbungsverfahren, bei welchem eine maximale Länge der auftretenden Ketten vorgegeben werden kann. Auch ist es bislang nicht gelungen, ein Auftreten geschlossener Ketten zu verhindern. Wir entwickeln in Abschnitt 2.2 einen neuen Färbungsalgorithmus für beliebige Quadrangulierungen, wobei ausschließlich nicht geschlossene Ketten auftreten. Während in [68] eine beliebige Färbung abgeändert wird, färben wir sukzessive durchlaufene Kegel, wobei die entstehenden Ketten möglichst früh unterbrochen werden. Dadurch sind die entstehenden Ketten im Allgemeinen kürzer als in [68]. Darüber hinaus zeigt sich, dass durch diese Vorgehensweise für sehr allgemeine Klassen von Quadrangulierungen ausschließlich Ketten einer vorgegebenen maximalen Länge auftreten. Selbst für den Fall von Kettenlänge zwei erhält man eine große Klasse. Insbesondere sind in den Klassen Quadrangulierungen mit vielen Eckpunkten vom Grad drei enthalten. Die Erfahrung zeigt, dass die Färbung solcher Quadrangulierungen besonders schwierig ist. Ein Beispiel für eine Quadrangulierung mit maximaler Kettenlänge zwei zeigt Abbildung 2.11 auf Seite 46. Bei der sukzessiven Färbung von Kegeln gewährleistet ein spezieller Teilalgorithmus, dass benachbarte gleichfarbige Vierecke stets Ketten bilden. Dabei werden vorübergehend entstehende T-Systeme, siehe Abbildung 1, aufgelöst, ohne dass sich die maximale Kettenlänge vergrößert.

Die auf diese Weise gefärbten Quadrangulierungen sind Grundlage für die lokale Interpolation mit  $C^1$ -Splines. Wir zeigen hierbei die Vorteile hinsichtlich Effizienz und Lokalität des Verfahrens. Neben dem in [68] beschriebenen kubischen Fall konstruieren wir Interpolationsmengen auch für Splines von höherem Grad. Dabei sind die Wahl von Interpolationspunkten im Inneren der Vierecke und auch die entsprechenden Beweise komplexer als für  $q = 3$ . Auf diese Weise ergeben sich für Splineräume beliebigen Grades  $q \geq 3$  lokal und stabil bestimmbare Interpolanten, die optimale Approximationsordnung besitzen.

Darüber hinaus entwickeln wir ein weiteres Verfahren für die Interpolation mit  $C^1$ -Splines vom Grad  $\geq 3$  auf Quadrangulierungen. Um Lokalität des Verfahrens zu erreichen, wählt man die Reihenfolge der Vierecke nach einem speziellen Prioritätsprinzip. Ähnliche Prinzipien sind in der Literatur bislang nur für Dreiecke, beispielsweise von Chui, Hecklin, Nürnberger und Zeilfelder [10], und für Tetraederzerlegungen, beispielsweise von Nürnberger, Schumaker und Zeilfelder [69], bekannt. Durch diese Vorgehensweise erhält man auch ohne Färbung der Vierecke lokal und stabil bestimmbare Interpolanten von optimaler Approximationsordnung. Abschließend vergleichen wir die verschiedenen Interpolationsmethoden. Es zeigt sich, dass die entwickelten Färbungs- und Prioritätsprinzipien Vorteile gegenüber [68] besitzen.

Kohlmüller, Nürnberger und Zeilfelder [47, 48] entwickelten das bislang einzige Interpolationsverfahren für 3D-Splineoberflächen. Gegeben eine Oberfläche  $\Omega_f$  im Raum, wird zunächst eine in der Nähe liegende grobe Triangulierung  $\Delta$  konstruiert und auf diesen Dreiecken eine stetige kubische 3D-Splineoberfläche ermittelt, welche  $\Omega_f$  interpoliert. Dies ist eine Abbildung  $\mathbf{s} : \Omega_\Delta \rightarrow \mathbb{R}^3$  in drei Komponenten  $(s_1, s_2, s_3)$ ,



wobei  $s_\tau|_T \in \mathcal{P}_3^2$  ist für alle  $T \in \Delta$  und  $\tau = 1, 2, 3$ . Anschließend wird der stetige Interpolant durch eine Modifikation bestimmter Bézier-Bernstein-Koeffizienten geglättet. Insbesondere ist er dadurch an den Eckpunkten und den Kantenmittelpunkten der Triangulierung differenzierbar.

Wir entwickeln in Kapitel 3 erstmals eine Interpolationsmethode mit 3D-Splineoberflächen auf Quadrangulierungen. Dabei bestimmen wir zunächst ebenfalls eine grobe Triangulierung und interpolieren mit einer stetigen 3D-Splineoberfläche. Im Anschluss fassen wir, grob gesprochen, benachbarte Paare von Dreiecken zu strikt konvexen Vierecken zusammen und erhalten dadurch eine Quadrangulierung im Raum. Ein effizienter Algorithmus garantiert, dass neben den zusammengefassten Vierecken lediglich sehr wenige innere Dreiecke einzeln bleiben. Somit lassen sich die in Kapitel 2 entwickelten Färbungs- und Prioritätsprinzipien in angepasster Weise auch auf 3D-Quadrangulierungen anwenden. Dabei wird die Splineoberfläche durch Modifikation bestimmter Bézier-Bernstein-Koeffizienten geglättet. Dadurch erhält man Differenzierbarkeit auf allen Eckpunkten und Kantenmittelpunkten der Quadrangulierung. Die geglätteten Interpolanten sind lokal bestimmbar. Da die triangulierten Quadrangulierungen weniger Dreiecke besitzen als in [47], ist der Datenaufwand für die Interpolanten geringer. Für Triangulierungen der Ebene ist es uns möglich, Ergebnisse für die Approximationsordnung zu geben. In diesem Fall zeigen wir ähnlich wie in [48], dass die Methode optimale Approximationsordnung besitzt.

Schließlich entwickeln wir in Kapitel 4 ein weiteres Interpolationsverfahren mit kubischen 3D-Splineoberflächen. Bei den bisherigen Methoden sind die Interpolanten lediglich an Eckpunkten und an Kantenmittelpunkten differenzierbar. Wir konstruieren erstmals interpolierende 3D-Splineoberflächen, die fast überall differenzierbar sind. Lediglich für wenige Kanten ergibt sich Differenzierbarkeit nur auf drei Punkten.

Zunächst wird ein stetiger Interpolant konstruiert. Die zugehörige (grobe) Triangulierung wird in Teiltriangulierungen zerlegt, welche sich in bestimmte Ebenen projizieren lassen. Innerhalb dieser Bereiche ergibt sich Differenzierbarkeit für die Spline-Interpolanten. Wir entwickeln effiziente Algorithmen, die selbst für komplizierte 3D-Triangulierungen sehr große projizierbare Bereiche liefern. Nach der Bestimmung dieser Bereiche wird, ausgehend von einer Schwarzweißfärbung, etwa die Hälfte der Dreiecke in jeweils drei Mikrodreiecke unterteilt. Anschließend modifizieren wir Bézier-Bernstein-Koeffizienten, sodass der Interpolant zunächst auf jedem Eckpunkt differenzierbar ist. Diese Modifikation erfolgt komponentenweise für die im Inneren der projizierten Bereiche liegenden Koeffizienten und ähnlich wie in Kapitel 3 für die Koeffizienten zwischen den Bereichen. Anschließend wird der Interpolant auf den Kanten geglättet. Dabei erhält man Differenzierbarkeit für die inneren Kanten jeder Teiltriangulierung und Differenzierbarkeit auf drei Punkten für die verbleibenden Kanten. Die interpolierenden 3D-Splineoberflächen sind lokal bestimmbar. Für ebene Ausgangstriangulierungen besitzt die Methode optimale Approximationsordnung.

---

Für die Ermittlung projizierbarer Bereiche entwickeln wir komplexe Algorithmen, die wir in Java<sup>TM</sup> und dem OpenGL-basierten Java 3D<sup>TM</sup> implementieren und testen. Ausgehend von Startdreiecken werden benachbarte Mengen von Dreiecken mit ähnlichen Normalenvektoren zusammengefasst und in die Ebenen dieser Startdreiecke projiziert. Dabei wird bei der Projektion ein festgelegter Winkel zwischen Dreiecken und Projektionsebenen nicht überschritten, sodass die projizierten Teiltriangulierungen nicht verzerrt sind. Die Erfahrung zeigt, dass die Größe projizierbarer Bereiche von der Wahl der Startdreiecke abhängig ist. Weiterhin entstehen bei einer Anwendung elementarer Algorithmen, grob gesprochen, Bereiche mit Löchern und „nicht glatten“ Rändern. Bei den von uns entwickelten Methoden werden Teiltriangulierungen mit solchen Eigenschaften vermieden. Darüber hinaus werden geeignete Startdreiecke ermittelt, sodass sich möglichst große projizierbare Bereiche ergeben. Auf diese Weise erhält man Spline-Interpolanten, welche durchschnittlich auf etwa 90% der Kanten differenzierbar sind.

Mein Dank gilt den Herren Professor Dr. Günther Nürnberger, Privatdozent Dr. Frank Zeilfelder und Dr. Gero Hecklin für ihre uneingeschränkte Unterstützung und ihre Anregungen bei den zahlreichen konstruktiven Gesprächen.

# Kapitel 1

## Grundlagen

Dieses Kapitel gibt einen Überblick über wesentliche Grundlagen der bivariaten Splinetheorie, wie sie unter anderem von Alfeld, Piper und Schumaker [1], de Boor [22, 23], de Boor und Höllig [25], Farin [36], Lai [55], Nürnberger [62, 63], Nürnberger und Zeilfelder [71] sowie Schumaker [79, 80] entwickelt wurden.

In Abschnitt 1.1 behandeln wir die Begriffe der Triangulierung  $\Delta$  und Quadrangulierung  $\diamond$  eines polygonalen Gebiets  $\Omega$  im  $\mathbb{R}^2$  und definieren Räume bivariater Splines auf solchen Zerlegungen. Das zentrale Problem der Lagrange- und der Hermite-Interpolation beschreiben wir in Abschnitt 1.2. Abschnitt 1.3 enthält eine Einführung in die Bézier-Bernstein-Darstellung bezüglich baryzentrischen Koordinaten. Diese ermöglicht eine einfache Formulierung von Differenzierbarkeitsbedingungen für bivariate Splines, siehe Farin [36]. Diese und wichtige andere Eigenschaften der Bézier-Bernstein-Methoden stellen wir in Abschnitt 1.5 dar. Schließlich geben wir einen Überblick über den Algorithmus von de Casteljau [27] und seine wesentlichen Anwendungen. Dabei handelt es sich um wichtige Techniken für die Darstellung von bivariaten Splineoberflächen sowie 3D-Splineoberflächen.

### 1.1 Splines auf Triangulierungen und Quadrangulierungen

#### Definition 1.1

Eine *Triangulierung*  $\Delta = \{T_1, \dots, T_N\}$  ist eine Unterteilung einer polygonalen Teilmenge  $\Omega \subset \mathbb{R}^2$  in  $N$  nicht degenerierte Dreiecke  $\{T_1, \dots, T_N\}$ , wobei der Durchschnitt jedes Paares  $\{T_i, T_j\}$  verschiedener Dreiecke entweder leer, ein gemeinsamer Eckpunkt oder eine gemeinsame Kante ist.

Dabei heißen Dreiecke mit gemeinsamer Kante *benachbart* und Dreiecke mit gemeinsamem Eckpunkt oder gemeinsamer Kante *einander berührend* oder auch *angren-*

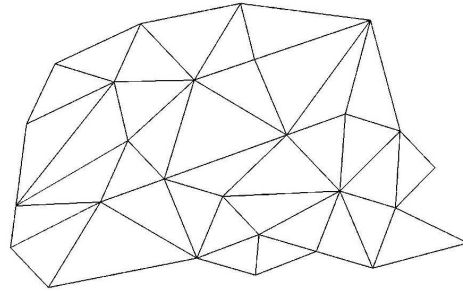


Abbildung 1.1: Eine kleine Triangulierung

*zend.* Ein zu einem Dreieck  $T$  benachbartes Dreieck wird auch als *Nachbar* bezeichnet. Für eine Menge  $\mathcal{M}$  von mehr als zwei Dreiecken gilt für die Benachbarkeit bzw. Angrenzung folgende rekursive Definition:  $\mathcal{M}$  ist benachbart bzw. angrenzend, falls ein Dreieck  $T \in \mathcal{M}$  mit einem benachbarten bzw. angrenzenden Dreieck aus  $\mathcal{M} \setminus \{T\}$  existiert, sodass  $\mathcal{M} \setminus \{T\}$  benachbart bzw. angrenzend ist. Abbildung 1.1 zeigt eine kleine Triangulierung.

Bei einer *Quadrangulierung* wird das polygonale Gebiet  $\Omega$  in Vierecke anstatt in Dreiecke unterteilt. Auch hier schneiden sich verschiedene Vierecke entweder in einem gemeinsamen Eckpunkt, einer gemeinsamen Kante, oder sie sind disjunkt. Die Begriffe von Benachbarkeit und Angrenzung sind analog wie bei Triangulierungen. Eine Quadrangulierung heißt *strikt konvex*, falls jedes enthaltene Viereck strikt konvex ist, also ausschließlich Innenwinkel  $< 180^\circ$  besitzt.

Da bivariate Splines stets auf Triangulierungen definiert sind, werden Quadrangulierungen  $\diamond$  in geeignete Triangulierungen  $\blacklozenge$  abgeändert, indem zu jedem Viereck  $Q \in \diamond$  eine oder beide Diagonalen hinzugefügt werden. Entsprechend heißt  $\blacklozenge$  *triangulierte Quadrangulierung* von  $\diamond$ .

Mit  $\mathcal{V}$  und  $\mathcal{E}$  werden die Menge der Eckpunkte und die Menge der Kanten einer Triangulierung oder Quadrangulierung bezeichnet, mit  $\mathcal{V}_I, \mathcal{V}_B, \mathcal{E}_I, \mathcal{E}_B$  unterscheidet man zwischen den Mengen der inneren Eckpunkte bzw. Kanten und den Mengen der Randeckpunkte bzw. -kanten. Verwendet man die Zeichen  $V$  und  $E$  statt den kalligraphischen Buchstaben, so bezieht man sich auf die Anzahl anstatt auf die Menge der Eckpunkte und Kanten. Ist der Bezug zur Zerlegung  $\Delta$  nicht eindeutig, so fügt man  $[\Delta]$  als Index hinzu. So bezeichnet beispielsweise  $\mathcal{E}_B^{[\Delta]}$  die Menge der Randkanten von  $\Delta$ . Des Weiteren sind mit  $\alpha$  und mit  $h$  der kleinste auftretende Innenwinkel sowie der größte Umkreis von Dreiecken aus  $\Delta$  bezeichnet.

Zwei Kanten  $e_1, e_2 \in \mathcal{E}$  heißen (*fast*) *kollinear*, falls sie (*fast*) die gleiche Steigung besitzen. Man spricht von *benachbarten Kanten*  $e_1, e_2 \in \mathcal{E}$ , falls sie Kanten eines gemeinsamen Dreiecks  $T \in \Delta$  sind. Eine Kante  $[v_1, v_2] \in \mathcal{E}$  heißt (*fast*) *degeneriert*

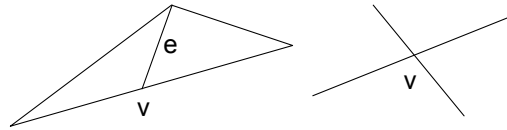


Abbildung 1.2: Bezüglich  $v$  degenerierte Kante  $e$  (links) und singulärer Knoten  $v$  (rechts)

in  $v_1$ , falls zwei (fast) kollineare, bezüglich  $v_1$  benachbarte Kanten von  $[v_1, v_2]$  existieren. Entsprechend heißt jede weitere Kante nichtdegeneriert. Man bezeichnet einen Knoten  $v \in \mathcal{V}$  als *singulär*, falls er Endpunkt von genau vier Kanten aus  $\Delta$  ist und diese auf zwei Geraden liegen. Abbildung 1.2 zeigt eine degenerierte Kante  $e$  und einen singulären Knoten  $v$ .

Als *Grad* eines Eckpunktes  $v \in \mathcal{V}$  bezeichnen wir die Anzahl der Kanten mit Endpunkt  $v$  in  $\Delta$  und schreiben  $\text{degree}(v)$ .

Nach Leonhard Euler erfüllen Triangulierungen die folgenden Zusammenhänge:

$$\begin{aligned} |\Delta| &= 2V_I + V_B - 2 \\ E_I &= 3V_I + V_B - 3 \\ E_B &= V_B \end{aligned} \tag{1.1}$$

Hinsichtlich der Lokalität von Interpolationsverfahren spielen so genannte *stars* von Dreiecken bzw. Vierecken um einen Eckpunkt  $v \in \mathcal{V}$  bzw. ein Dreieck  $T \in \Delta$  eine Rolle. Unter dem  $\text{star}^1(v)$ , dem so genannten *closed star*, versteht man die Menge aller Dreiecke aus  $\Delta$  mit Eckpunkt  $v$ . Für  $n \geq 2$  ist der  $\text{star}^n$  rekursiv definiert als

$$\text{star}^n(v) = \{T \in \Delta : T \cap \Omega_{\text{star}^{n-1}(v)} \neq \emptyset\}. \tag{1.2}$$

Bezüglich eines Dreiecks  $T$  bezeichnet der  $\text{star}^1(T)$ , wiederum *closed star* genannt, die Vereinigung von  $T$  mit allen angrenzenden Dreiecken, und  $\text{star}^n(T)$ ,  $n \geq 2$ , ist analog zu (1.2) definiert. Für Quadrangulierungen gelten die Definitionen entsprechend.

Wir definieren nun Räume bivariater Splinefunktionen. Splines sind stückweise Polynome von vorgegebenem Grad  $q$ , d. h. die zu approximierenden Funktionen werden lokal durch Polynome angenähert.

**Definition 1.2**

Der Raum

$$\mathcal{S}_q^r(\Delta) := \{s \in \mathcal{C}^r(\Omega) : s|_T \in \mathcal{P}_q^2 \text{ für alle } T \in \Delta\} \tag{1.3}$$

heißt *Raum der bivariaten Splinefunktionen* von Grad  $q \in \mathbb{N}$  und Differenzierbarkeitsordnung  $r \in \mathbb{N}_0$ ,  $r < q$ , und ist definiert auf einer Triangulierung  $\Delta$  des polygonalen Gebiets  $\Omega$ . Hierbei bezeichnet  $\mathcal{P}_q^2 = \text{span} \{x^i y^j : i, j \in \mathbb{N}_0, i + j \leq q\}$  den

Raum der bivariaten Polynome vom totalen Grad  $q$ . Sind durch natürliche Zahlen  $\rho_i$ ,  $i = 1, \dots, d = V$ ,  $r \leq \rho_i < q$ ,  $i = 1, \dots, d$ , außerdem Differenzierbarkeitsordnungen für die Knoten  $\{v_1, \dots, v_d\}$  von  $\Delta$  gegeben, so definiert

$$\mathcal{S}_q^{r,\theta}(\Delta) = \{s \in \mathcal{S}_q^r(\Delta) : s \in C^{\rho_i}(v_i), i = 1, \dots, d\} \quad (1.4)$$

den *Supersplinerraum* vom Grad  $q$ , einen Teilraum des Raums  $\mathcal{S}_q^r(\Delta)$ . Dabei ist  $\theta = (\rho_1, \dots, \rho_d)$ .

Die Dimension des bivariaten Polynomraums vom totalen Grad  $q$  beträgt

$$\dim \mathcal{P}_q^2 = \sum_{i=0}^q (i+1) = \frac{(q+1)(q+2)}{2} = \binom{q+2}{2}. \quad (1.5)$$

Die Dimension von Splineräumen  $\mathcal{S}_q^r(\Delta)$  und Supersplineräumen  $\mathcal{S}_q^{r,\theta}(\Delta)$  ist je nach Wahl von  $q$ ,  $r$  und  $\theta$  oftmals sehr schwierig zu ermitteln, was mit der komplexen Struktur dieser Räume zu tun hat. So ist beispielsweise die Dimension des Raums  $\mathcal{S}_3^1(\Delta)$  auf beliebigen Triangulierungen  $\Delta$  bislang nicht bekannt.

Die Frage nach der Dimension eines Splineriums  $\mathcal{S}_q^r(\Delta)$  hängt mit dem Problem der *minimal bestimmenden Menge* zusammen. Darunter versteht man eine kleinstmögliche Menge von Bézier-Bernstein-Punkten (siehe Abschnitt 1.3), sodass die zugehörigen Bézier-Bernstein-Koeffizienten alle verbleibenden Bézier-Bernstein-Koeffizienten eindeutig bestimmen. Nach Alfeld und Schumaker [3] existiert für jeden Splinerraum eine solche minimal bestimmende Menge, und die Anzahl der enthaltenen Elemente gibt die Dimension von  $\mathcal{S}_q^r(\Delta)$  an.

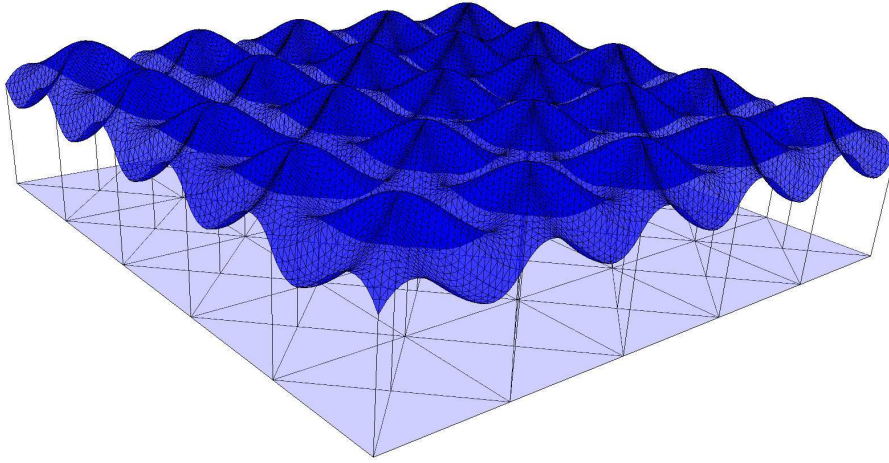
Als *Graphen*  $G(s)$  eines Splines  $s : \Omega \rightarrow \mathbb{R}$  wie auch einer anderen bivariaten Abbildung bezeichnet man dessen visuelle Darstellung im Raum:

$$s : \begin{cases} \Omega & \rightarrow \mathbb{R} \\ (x, y) & \mapsto p(x, y) \end{cases} \\ G(s) = \{ (z, s(z)) : z \in \Omega \} \subseteq \mathbb{R}^3$$

Abbildung 1.3 zeigt den Graphen eines Splines, definiert auf einer regelmäßigen  $\Delta^2$ -Triangulierung. In ähnlicher Weise sind zu 3D-Splineoberflächen gehörende Graphen definiert, siehe Kapitel 3.

## 1.2 Interpolation mit Splines

Während die Interpolation mit univariaten Splines durch die Bedingungen von Schoenberg und Whitney [78] vollständig charakterisiert ist, ist der bivariate Fall

Abbildung 1.3: Spline auf einer  $\Delta^2$ -Triangulierung

aufgrund der komplexen Struktur der zugrunde liegenden Splineräume wesentlich schwieriger.

Ein häufig verwendetes Verfahren ist die *Lagrange-Interpolation* für einen Splineräum  $\mathcal{S}_q^r(\Delta)$ . Man konstruiert eine Menge  $\{z_1, \dots, z_d\}$  von  $d = \dim \mathcal{S}_q^r(\Delta)$  Punkten, sodass für jede Funktion  $f \in \mathcal{C}(\Omega)$  ein eindeutiger Spline  $s \in \mathcal{S}_q^r(\Delta)$  existiert, welcher die Interpolationseigenschaften

$$s(z_i) = f(z_i), \quad i = 1, \dots, d, \quad (1.6)$$

erfüllt. Man bezeichnet diese Menge als Lagrange-Interpolationsmenge. Offensichtlich genügt es, wenn von  $f$  lediglich die  $d$  Auswertungen  $f_{z_i} := f(z_i)$ ,  $i = 1, \dots, d$ , vorliegen.

Für Interpolationsmengen bezüglich Splineräumen ist von Bedeutung, welche Interpolationspunkte für die einzelnen Polynomstücke  $s|_{T_i}$ ,  $i = 1, \dots, N$ , in Frage kommen. Es existiert ein zentraler Satz hinsichtlich der Lage von Interpolationspunkten für bivariate Polynome, welcher besagt, dass für eine Interpolationsmenge  $\mathcal{L}_q = \{z_1, \dots, z_{\binom{q+2}{2}}\}$  des Raums  $\mathcal{P}_q^2$  eine hinreichende Bedingung die Existenz von  $q$  Geraden  $g_i$ ,  $i = 1, \dots, q$ , ist, sodass auf jeder Geraden  $g_i$  genau  $i + 1$  Punkte in  $\mathcal{L}_q$  enthalten sind, aber auf keinem Geradenschnittpunkt Interpolationspunkte liegen. Eine solche Lage wird *Standardlage von Interpolationspunkten* für bivariate Polynome genannt. Die wichtigste Variante der Standardlage sind die so genannten Bézier-Bernstein-Punkte, die wir im folgenden Abschnitt einführen. Abbildung 1.4 zeigt diese Punkte auf einem Dreieck für den Fall  $q = 7$ .

Falls nicht nur Übereinstimmungen von Funktionswerten sondern auch von partiellen Ableitungen gefordert werden, die Anzahl der Bedingungen aber weiterhin mit

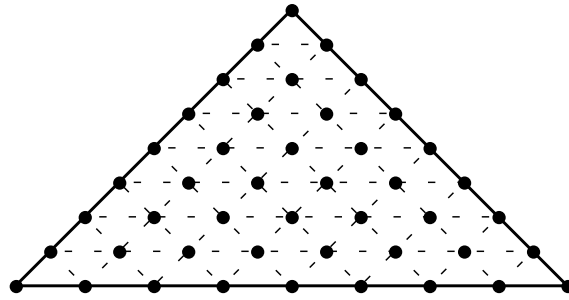


Abbildung 1.4: Bézier-Bernstein-Punkte eines Dreiecks als Standardlage von Interpolationspunkten ( $q = 7$ )

der Dimension von  $\mathcal{S}_q^r(\Delta)$  übereinstimmt, so spricht man von einer *Hermite-Interpolation*.

Für eine gegebene Lagrange-Interpolationsmenge  $\{z_1, \dots, z_d\}$  ist es zweckmäßig, den Interpolanten  $s$  bezüglich der so genannten *dualen Basis* darzustellen. Deren Basisfunktionen  $\mathcal{B}_1, \dots, \mathcal{B}_d : \Omega \rightarrow \mathbb{R}$ , die so genannten *Basis-splines*, sind charakterisiert durch die Bedingung

$$\mathcal{B}_i(z_j) = \delta_{ij} := \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j \end{cases} \quad \text{für alle } i, j = 1, \dots, d.$$

Aufgrund der Interpolationseigenschaft der Menge  $\{z_1, \dots, z_d\}$  ist diese Basis in eindeutiger Weise gegeben und der Interpolant  $s$  von  $f$  hat die folgende Darstellung:

$$s \equiv \sum_{i=1}^d f(z_i) \mathcal{B}_i \tag{1.7}$$

### 1.3 Bézier-Bernstein-Darstellung

Die Bézier-Bernstein-Methoden vereinfachen die Interpolation mit bivariaten Splinesfunktionen und deren Darstellung. Diese Techniken basieren auf der so genannten Bézier-Bernstein-Darstellung der über den Dreiecken  $T \in \Delta$  definierten Polynomstücke  $p \in \mathcal{P}_q^2$ . Die Punkte des Dreiecks werden hierbei nicht bezüglich eines kartesischen sondern eines *baryzentrischen Koordinatensystems* dargestellt. Die Verwendung baryzentrischer Koordinaten bietet insbesondere den Vorteil der affinen Invarianz.

Bezüglich eines nicht degenerierten Dreiecks  $T := [v_1, v_2, v_3]$  der Ebene existiert für jeden Punkt  $z \in \mathbb{R}^2$  ein eindeutiges Tripel  $\phi(z) = (\phi_1, \phi_2, \phi_3)(z) \in \mathbb{R}^3$ , genannt



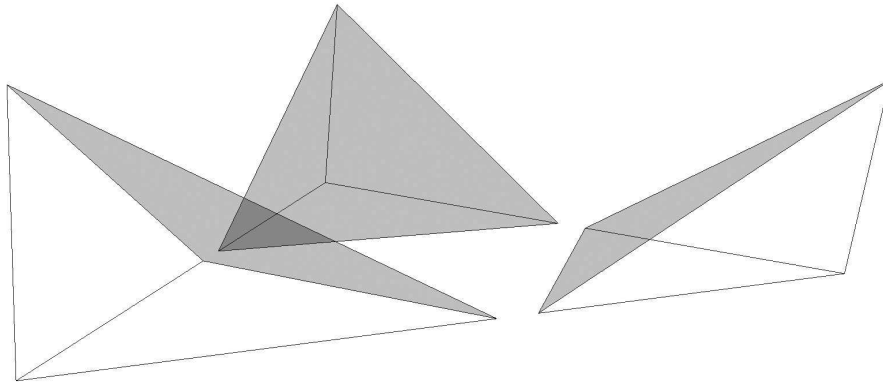


Abbildung 1.5: Baryzentrische Koordinaten  $\phi_1, \phi_2$  und  $\phi_3$  über einem Dreieck

*baryzentrischer Koordinatenvektor* von  $z$ , mit folgender Eigenschaft:

$$z = \sum_{i=1}^3 \phi_i(z)v_i, \quad \text{wobei} \quad \phi_1(z) + \phi_2(z) + \phi_3(z) = 1 \quad (1.8)$$

Somit ist jede baryzentrische Koordinate  $\phi_i : T \rightarrow \mathbb{R}$ ,  $i = 1, 2, 3$ , eine durch die Beziehung  $\phi_i(z_j) = \delta_{ij}$ ,  $i, j = 1, 2, 3$ , eindeutig festgelegte lineare Abbildung.

Für die Ableitung von durch baryzentrische Koordinaten dargestellten Funktionen verwendet man *baryzentrische Richtungsvektoren*. Hierbei wird ein Richtungsvektor  $v$  von einem Punkt  $z$  zu einem Punkt  $\tilde{z}$  dargestellt durch

$$\vec{\phi}(v) = \phi(\tilde{z}) - \phi(z). \quad (1.9)$$

Im Gegensatz zu baryzentrischen Koordinaten ist offensichtlich  $\sum_{i=1,2,3} \vec{\phi}_i(v) = 0$ .

Mit der Verwendung von baryzentrischen Koordinaten lassen sich nun die so genannten *bivariaten Bernsteinpolynome* definieren, welche eine Basis des bivariaten Polynomraums  $\mathcal{P}_q^2$  bilden:

$$B_{ijk}^q \equiv \frac{q!}{i!j!k!} \phi_1^i \phi_2^j \phi_3^k, \quad i + j + k = q \quad (1.10)$$

Die Indizes  $i, j, k$  liegen hierbei und im Folgenden in  $\mathbb{N}_0$ . Bivariate Bernsteinpolynome besitzen eine Reihe wichtiger Eigenschaften. Beispielsweise nimmt ein bivariates Polynom  $B_{ijk}^q$  auf Punkten des zugrunde liegenden Dreiecks stets Werte im Intervall  $[0, 1]$  an, lässt sich rekursiv aus Bernsteinpolynomen vom Grad  $q - 1$  berechnen und degeneriert auf jeder Kante zu einem univariaten Bernsteinpolynom in zwei baryzentrischen Koordinaten. Weiterhin gilt  $\sum_{i+j+k=q} B_{ijk}^q \equiv 1$ .

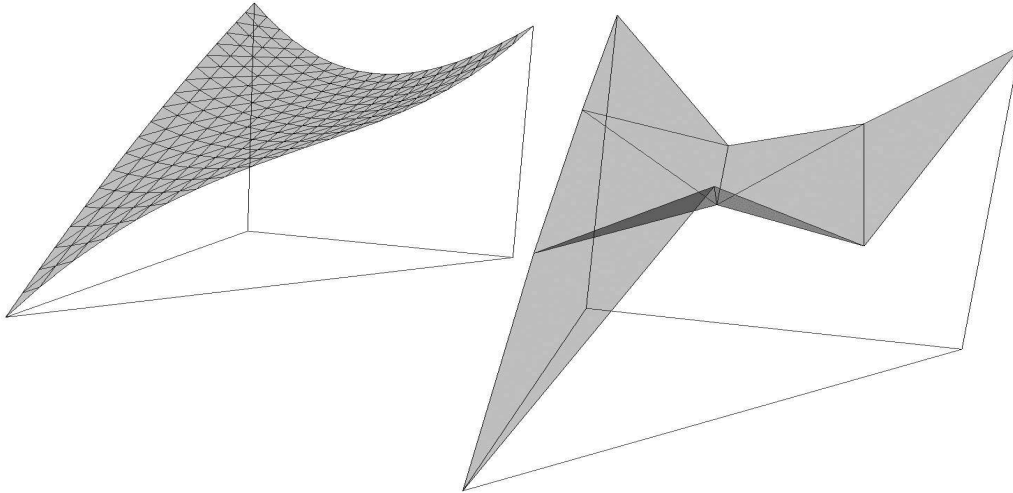


Abbildung 1.6: Bézier-Bernstein-Netz eines Dreiecks

Mit der Bernsteinbasis lässt sich jedes Polynom  $p \in \mathcal{P}_q^2$  eindeutig darstellen als

$$p \equiv \sum_{i+j+k=q} c_{ijk} B_{ijk}^q, \quad (1.11)$$

wobei die Werte  $c_{ijk} \in \mathbb{R}$  *Bézier-Bernstein-Koeffizienten* von  $p$  heißen. Diese Darstellung bivariater Polynome heißt *Bézier-Bernstein-Darstellung*.

Die Bézier-Bernstein-Darstellung eines Polynoms  $p \in \mathcal{P}_q^2$  lässt sich durch das so genannte *Bézier-Bernstein-Netz* geometrisch veranschaulichen. Den durch

$$\mathcal{P}_q(T) := \left\{ P_{ijk} := \frac{i}{q}v_1 + \frac{j}{q}v_2 + \frac{k}{q}v_3 \quad : \quad i + j + k = q \right\} \quad (1.12)$$

definierten *Bézier-Bernstein-Punkten in  $T$*  ordnet man die Koeffizienten  $c_{ijk}$  von  $p$  zu und erhält das Bézier-Bernstein-Netz von  $p$  als

$$\mathcal{N}_p(T) := \{ N_{ijk} := (P_{ijk}, c_{ijk}) \quad : \quad i + j + k = q \}. \quad (1.13)$$

Von den  $\dim \mathcal{P}_q^2 = \frac{(q+1)(q+2)}{2}$  Bézier-Bernstein-Punkten werden die drei zu den Eckpunkten von  $T$  gehörenden Punkte *Eckpunkte*, die auf den Kanten von  $T$  liegenden Punkte *Randpunkte* und die verbleibenden Punkte *innere* Bézier-Bernstein-Punkte genannt. Außerdem ist der *Abstand zweier Bézier-Bernstein-Punkte* durch

$$d(P_{ijk}, P_{\tilde{i}\tilde{j}\tilde{k}}) := \max \left\{ |i - \tilde{i}|, |j - \tilde{j}|, |k - \tilde{k}| \right\}, \quad (1.14)$$

die *Benachbarkeit* zweier Punkte als Abstand von eins sowie der *Abstand eines Punktes zu einer Kante* als minimalen Abstand zu den Punkten auf der Kante definiert.

Unter einem  $C^n$ -Ring bzw. einer  $C^n$ -Scheibe,  $n \in \mathbb{N}_0$ , um einen Punkt versteht man die Menge aller Punkte mit Abstand  $n$  bzw. Abstand  $\leq n$  zu diesem Punkt.

Selbige Notationen verwendet man auch für die zu den Punkten  $P$  gehörenden Netzpunkte und Bézier-Bernstein-Koeffizienten. Abbildung 1.6 zeigt ein bivariates kubisches Polynom und das zugehörige Bézier-Bernstein-Netz.

Bézier-Bernstein-Netze besitzen einige für die Anwendung wichtige Eigenschaften. Beispielsweise interpolieren die drei zu den Eckpunkten gehörenden Netzpunkte das zugehörige Polynom  $p$ . Weiterhin liegt der Graph von  $p|_T$  in der konvexen Hülle der Netzpunkte, und für jeden Eckpunkt  $v$  von  $T$  spannen die drei Netzpunkte mit maximalem Abstand eins zu  $v$  die Tangentialebene von  $p$  in  $v$  auf.

Diese Darstellung lässt sich auf bivariate Splines  $s \in \mathcal{S}_q^r(\Delta)$  übertragen: Jedes Splinstück  $s|_T \in \mathcal{P}_q^2$ ,  $T \in \Delta$ , wird in Bézier-Bernstein-Darstellung repräsentiert durch die Menge  $\{c_{ijk}^{[T]} : i + j + k = q\}$  der Koeffizienten. Definiert man demzufolge das Bézier-Bernstein-Netz von  $s$  als Vereinigung der zugehörigen Polynomnetze, so erhält man eine grafisch darstellbare Repräsentation von  $s$ , nämlich das *Bézier-Bernstein-Netz*

$$\mathcal{N}_s(\Delta) = \bigcup_{T \in \Delta} \mathcal{N}_{s|_T}(T) \tag{1.15}$$

über den Bézier-Bernstein-Punkten  $\mathcal{P}_q(\Delta) = \bigcup_{T \in \Delta} \mathcal{P}_q(T)$ . Benachbarkeit, Abstand, Scheiben und Ringe von Bézier-Bernstein-Punkten, -Netzpunkten sowie -Koeffizienten sind entsprechend dem polynomialen Fall definiert.

## 1.4 Richtungsableitungen und Bézier-Bernstein-Koeffizienten

Für Richtungsableitungen von bivariaten reellwertigen Abbildungen  $f : \Omega \rightarrow \mathbb{R}$  an einer Stelle  $(x, y) \in \Omega$  in Richtung  $v := (v_1, v_2) \in \mathbb{R}^2 \setminus \{(0, 0)\}$  schreiben wir

$$D_v f(x, y) = v \operatorname{grad} f(x, y) = v_1 \frac{\partial}{\partial x} f(x, y) + v_2 \frac{\partial}{\partial y} f(x, y).$$

Ferner bezeichne  $D_v^n f(x, y) = \underbrace{D_v \dots D_v}_{n \text{ mal}} f(x, y)$  die  $n$ -te Richtungsableitung von  $f$ .

Partielle Ableitungen von Polynomstücken können dabei direkt auf entsprechende Ableitungen der Bernstein-Basispolynome zurückgeführt werden, für Details siehe z. B. [59], S. 32f. Daraus ergibt sich ein Zusammenhang zwischen partiellen Spline-Ableitungen in den Knoten und Kanten der Triangulierung und in der Nähe liegenden Bézier-Bernstein-Koeffizienten. In einem Eckpunkt gilt die folgende Relation:

**Theorem 1.3**

Für ein Polynom  $p \in \mathcal{P}_q^2$  mit Bézier-Bernstein-Koeffizienten  $c_{ijk}$ ,  $i + j + k = q$ , bezüglich eines Dreiecks  $T = [v_1, v_2, v_3]$  und für alle  $\rho, \sigma \in \mathbb{N}_0$  mit  $\rho + \sigma \leq q$  sind äquivalent:

- (i) Die Bézier-Bernstein-Koeffizienten  $c_{q-j-k,j,k}$
- (ii) Die partiellen Ableitungen  $\frac{\partial^{j+k} p}{\partial^j (v_2-v_1) \partial^k (v_3-v_1)}(v_1)$

sind für alle  $j = 0, \dots, \rho$  und  $k = 0, \dots, \sigma$  eindeutig bestimmt.

Eine ähnliche Aussage existiert für partielle Ableitungen an den Kanten von  $T$ .

## 1.5 $C^r$ -Bedingungen

Durch die Bézier-Bernstein-Darstellung können Differenzierbarkeitsbedingungen an den Eckpunkten und Kanten der Triangulierung direkt durch Beziehungen von Bézier-Bernstein-Koeffizienten formuliert werden, siehe beispielsweise [11, 12, 22, 23, 33, 36]. Diese Formulierung vereinfacht ganz erheblich den Umgang mit Splineräumen und zugehörigen Interpolationsproblemen. Gemäß Theorem 1.3 und der entsprechenden Aussage für die Kanten der Dreiecke spielen, grob gesprochen, für Differenzierbarkeitsbedingungen  $r$ -ter Ordnung über eine Kante bzw. einen Eckpunkt genau die Bézier-Bernstein-Koeffizienten mit Abstand  $\leq r$  zu der Kante bzw. dem Eckpunkt eine Rolle, siehe auch Abschnitt 1.7.

Betrachten wir im Folgenden Polynomstücke  $p_1, p_2$  über zwei Dreiecken  $T_1 := [v_1, v_2, v_3] \subset \mathbb{R}^2$  und  $T_2 := [w_1, w_2, w_3] \subset \mathbb{R}^2$  der Ebene, welche sich in genau einem Eckpunkt  $v_1 = w_1$  bzw. genau einer Kante  $[v_1, v_2]$ ,  $v_i = w_i$ ,  $i = 1, 2$ , schneiden und formulieren die  $C^r$ -Bedingungen über  $v_1$  bzw.  $[v_1, v_2]$  in Bézier-Bernstein-Schreibweise. Hierbei bezeichnen wir für  $l = 1, 2$  mit  $\phi_l(v) = [\phi_l^1(v), \phi_l^2(v), \phi_l^3(v)]$  die baryzentrischen Koordinaten eines Punktes  $v \in \mathbb{R}^2$  bezüglich  $T_l$ . Dann lässt sich nach Alfeld, Piper und Schumaker [2] für die zusammengesetzte Abbildung

$$s : T_1 \cup T_2 \rightarrow \mathbb{R} \text{ definiert durch } s(v) := \begin{cases} p_1(v) & \text{für } v \in T_1 \\ p_2(v) & \text{für } v \in T_2 \setminus T_1 \end{cases}$$

die  $C^r$ -Bedingung für den Fall einer gemeinsamen Kante  $[v_1, v_2]$  wie folgt darstellen:

**Theorem 1.4**

Schneiden sich  $T_1$  und  $T_2$  genau in der Kante  $[v_1, v_2] = [w_1, w_2]$ , so ist  $s$  genau dann  $r$ -mal stetig differenzierbar in dieser Kante, wenn für alle  $k \leq r$ ,  $i + j + k = q$ , die Bedingung

$$c_{ijk}^{[T_2]} = \sum_{i_1+j_1+k_1=k} c_{i+i_1,j+j_1,k_1}^{[T_1]} B_{i_1 j_1 k_1}^k[\phi_1(w_3)] \tag{1.16}$$

erfüllt ist.

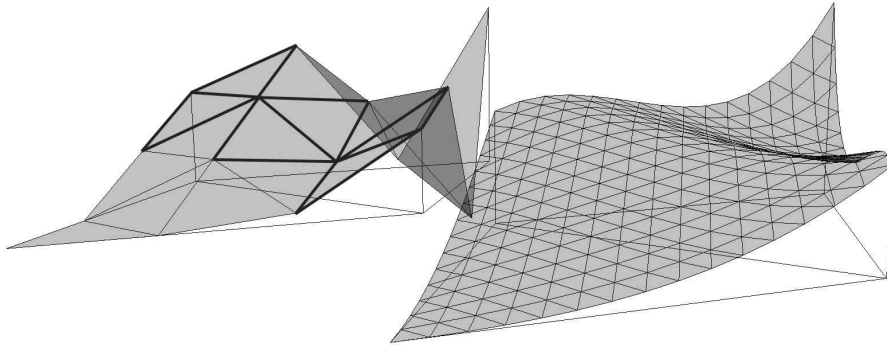


Abbildung 1.7: Erfüllte  $C^1$ -Bedingungen über eine gemeinsame Kante  $q = 3$

Aus diesem Theorem folgt ein ähnlicher Zusammenhang für die  $C^r$ -Bedingung, falls  $T_1$  und  $T_2$  sich in einem Eckpunkt  $v_1$  schneiden:

**Korollar 1.5**

Schneiden sich  $T_1$  und  $T_2$  genau im Eckpunkt  $v_1 = w_1$ , so ist  $s$  genau dann  $r$ -mal stetig differenzierbar in diesem Eckpunkt, wenn für alle  $j + k \leq r$ ,  $i + j + k = q$ , die Bedingung

$$c_{ijk}^{[T_2]} = \sum_{i_1+j_1+k_1=j} \sum_{i_2+j_2+k_2=k} c_{i+i_1+i_2, j_1+j_2, k_1+k_2}^{[T_1]} B_{i_1 j_1 k_1}^j[\phi_1(w_2)] B_{i_2 j_2 k_2}^k[\phi_1(w_3)] \tag{1.17}$$

erfüllt ist.

Für  $r = 0, 1$  lassen sich diese Bedingungen grafisch anschaulich interpretieren: Stetigkeit in  $v_1$  bzw.  $[v_1, v_2]$  ist genau dann gegeben, wenn die auf  $v_1$  bzw.  $[v_1, v_2]$  liegenden Bézier-Bernstein-Koeffizienten für die angrenzenden Polynomstücke identisch sind. Differenzierbarkeit über einen Eckpunkt  $v_1$  liegt genau dann vor, wenn die Bézier-Bernstein-Netzpunkte von  $s$  der  $C^1$ -Scheibe um  $v_1$  in einer Ebene liegen. Für die Differenzierbarkeit über eine Kante  $[v_1, v_2]$  sind alle Paare gegenüber liegender Netzpunkte mit Abstand eins zu  $[v_1, v_2]$  zu betrachten. Genau dann, wenn jedes solche Paar mit den beiden jeweils benachbarten Netzpunkten auf der gemeinsamen Kante in einer Ebene liegt, ist die stetige Differenzierbarkeit über  $[v_1, v_2]$  gegeben. Als Beispiel zeigt Abbildung 1.7 zwei an einer Kante stetig differenzierbar zusammengesetzte Polynomstücke und deren Bézier-Bernstein-Netze.

## 1.6 Approximationsordnung von Splineräumen und Interpolation

Während ein Spline-Interpolant  $\mathcal{S}_q^{r,\theta}(\Delta)$  und die zu interpolierende Funktion  $f \in C^{q+1}(\Omega_\Delta)$  auf jedem Interpolationspunkt  $z \in \mathcal{L}$  übereinstimmen, ist für die Approximationsgüte entscheidend, dass auch für alle weiteren Interpolationswerte  $f(z)$ ,  $z \in \Omega_\Delta \setminus \mathcal{L}$ , die Abweichungen  $|f(z) - s(z)|$  gering sind. Dabei ist wichtig, dass bei einer geeigneten Verfeinerung von  $\Delta$  der Interpolationsfehler

$$\|f - s\| := \max_{z \in \Omega_\Delta} |f(z) - s(z)|$$

schnell gegen Null geht. Hierfür ist die Approximationsordnung ein Kriterium:

### Definition 1.6

Sei  $f \in C^{q+1}(\Omega_\Delta)$ . Die größte natürliche Zahl  $k$ , für welche eine lediglich von  $f$  und dem kleinsten Winkel  $\alpha$  in  $\Delta$  aber nicht vom größten Dreiecksumkreis  $h$  in  $\Delta$  abhängende Konstante  $K > 0$  existiert mit

$$d(f, \mathcal{S}_q^{r,\theta}(\Delta)) := \inf \{ \|f - s\| : s \in \mathcal{S}_q^{r,\theta}(\Delta) \} \leq K \cdot h^k$$

heißt *Approximationsordnung* von  $\mathcal{S}_q^{r,\theta}(\Delta)$ .

Ciarlet und Raviart [13] zeigten, dass  $q + 1$  eine obere Schranke für die Approximationsordnung ist. Daher heißt die Approximationsordnung für den Fall  $k = q + 1$  *optimal*. Nach de Boor und Jia [26] wird dies nicht durch jede Interpolationsmethode erreicht. Neben Grad und Differenzierbarkeitsordnung des betrachteten Splineräums und der Geometrie von  $\Delta$  spielen insbesondere die Lokalitätseigenschaften der Methode eine Rolle, siehe de Boor und Höllig [24,25] sowie Davydov, Nürnberger und Zeilfelder [17,20].

Ein häufig verwendetes Verfahren zum Nachweis der Approximationsordnung ist die schwache Interpolation. Sie wurde von Nürnberger [63], Nürnberger und Walz [70], Davydov, Nürnberger und Zeilfelder [20] sowie Nürnberger und Zeilfelder [72] entwickelt. Die Approximationsordnung  $k$  der Methode wird dabei durch eine Bedingung gezeigt, die etwas „schwächer“ ist als das Interpolationskriterium. Grob gesprochen zeigt man für jedes Dreieck  $T \in \Delta$  die Existenz von  $m = \dim \mathcal{P}_q^2$  Punkten  $z_i \in T$ ,  $i = 1, \dots, m$ , welche die Bedingung

$$|(f - s)(z_i)| \leq K \cdot h^k, \quad i = 1, \dots, m,$$

erfüllen und folgert

$$\|f - s|_T\| \leq K \cdot h^k$$

für eine geeignete Konstante  $K$ . Verfährt man in dieser Weise für jedes Dreieck aus  $\Delta$ , so folgt unmittelbar, dass  $s$  Approximationsordnung  $k$  besitzt.

## 1.7 Anwendungen des Algorithmus von de Casteljau

Die rekursiv-algorithmische Auswertung von Bernsteinpolynomen durch den Algorithmus von de Casteljau spielt innerhalb der Bézier-Bernstein-Methoden eine zentrale Rolle. Sie ermöglicht nicht nur eine effiziente Berechnung von Splines, sondern liefert insbesondere auch Differenzierbarkeitsbedingungen, siehe z. B. [8]. Das Verfahren geht auf einen 1959 erschienenen Artikel des Citroën-Mitarbeiters P. de Casteljau [27] zurück, in welchem sowohl univariate Bézier-Kurven als auch über Dreiecken definierte bivariate Polynomstücke behandelt werden. Bei der Beschreibung der Methode beschränken wir uns im Folgenden auf den bivariaten Fall.

Für die Vorgehensweise von de Casteljau ist entscheidend, dass sich jedes Bernsteinpolynom  $B_{ijk}^q$ ,  $i + j + k = q$ ,  $q \geq 1$ , wie folgt rekursiv aus drei  $(q - 1)$ -gradigen Polynomen bestimmen lässt:

$$B_{ijk}^q(\phi) \equiv \phi_1 \cdot B_{i-1,j,k}^{q-1}(\phi) + \phi_2 \cdot B_{i,j-1,k}^{q-1}(\phi) + \phi_3 \cdot B_{i,j,k-1}^{q-1}(\phi) \quad (1.18)$$

Das Theorem von de Casteljau liefert eine rekursive Vorgehensweise zur Auswertung von bivariaten Bernsteinpolynomen:

### Satz 1.7 (de Casteljau)

Seien  $p \equiv \sum_{i+j+k=q} c_{ijk} B_{ijk}^q$  ein bivariates Polynom vom totalen Grad  $q$  und  $\phi$  die baryzentrischen Koordinaten einer auszuwertenden Stelle. Dann lässt sich aus den rekursiv definierten de Casteljau-Koeffizienten

$$\begin{aligned} c_{ijk}^0(\phi) &= c_{ijk}, & i + j + k &= q, \\ c_{ijk}^r(\phi) &= \phi_1 \cdot c_{i+1,j,k}^{r-1} + \phi_2 \cdot c_{i,j+1,k}^{r-1} + \phi_3 \cdot c_{i,j,k+1}^{r-1}, & i + j + k &= q - r, \end{aligned}$$

für alle  $0 \leq r \leq q$  die Auswertung  $p(\phi)$  wie folgt berechnen:

$$p(\phi) = \sum_{i+j+k=q-r} c_{ijk}^r B_{ijk}^{q-r}(\phi)$$

Insbesondere erhält man offensichtlich für  $r = 0$  die Bézier-Bernstein-Darstellung von  $p$  und für  $r = q$  den Zusammenhang

$$p(\phi) = c_{000}^q(\phi). \quad (1.19)$$

Abbildung 1.8 veranschaulicht den Algorithmus für den kubischen Fall.

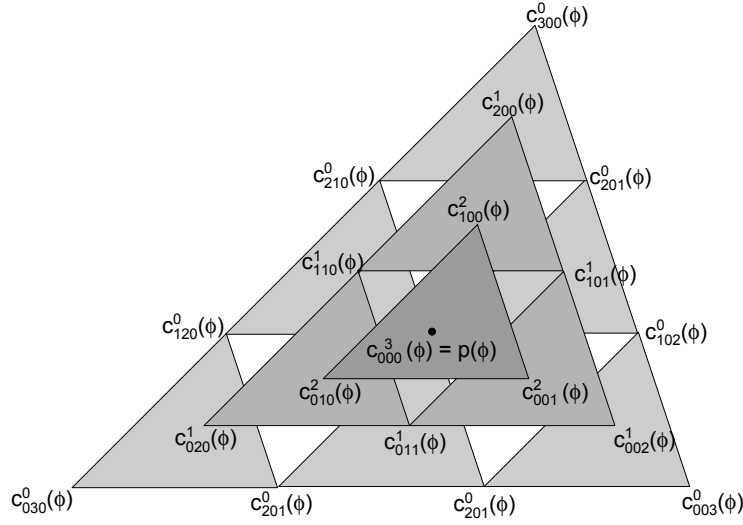


Abbildung 1.8: Algorithmus von de Casteljau ( $q = 3$ ); Jeweils drei de Casteljau-Koeffizienten  $c_{ijk}^r(\phi)$ ,  $i + j + k = q - r$ ,  $r = 0, \dots, q - 1$ , werden für die Bestimmung eines Koeffizienten mit Index  $r + 1$  verwendet, bis in der  $q$ -ten Iteration die Auswertung  $p(\phi) = c_{000}^q(\phi)$  ermittelt ist. Dabei gilt  $c_{ijk}^0(\phi) = c_{ijk}$ ,  $i + j + k = q$ .

Durch ein ähnliches Verfahren lassen sich auch Richtungsableitungen von Bézier-Bernstein-Polynomen berechnen. Dabei ist von Bedeutung, dass sich die  $r$ -te Ableitung eines Bernsteinpolynoms  $B_{ijk}^q$  in Richtung  $\vec{\phi}$  wie folgt als Auswertung geeigneter Bernsteinpolynome bestimmen lässt:

$$D_{\vec{\phi}}^r B_{ijk}^q(\phi) = \frac{q!}{(q-r)!} \sum_{i'+j'+k'=r} B_{i'j'k'}^r(\vec{\phi}) B_{i-i',j-j',k-k'}^{q-r}(\phi) \quad (1.20)$$

Hierzu bemerken wir, dass sich Bernsteinpolynome für baryzentrische Richtungsvektoren  $\vec{\phi}$  auswerten lassen, auch wenn  $\sum_{i=1,2,3} \vec{\phi}_i = 0 \neq 1$  ist.

Damit erhält man den folgenden Zusammenhang zur einfachen Bestimmung von Richtungsableitungen bivariater Polynome. Sei  $p \equiv \sum_{i+j+k=q} c_{ijk} B_{ijk}^q$  ein bivariates Polynom.

### Satz 1.8

Die  $r$ -te Ableitung von  $p$  an der Stelle  $\phi$  in Richtung  $\vec{\phi}$  lässt sich mit Hilfe der de Casteljau-Koeffizienten wie folgt bestimmen:

$$D_{\vec{\phi}}^r p(\phi) = \frac{q!}{(q-r)!} \sum_{i+j+k=r} c_{ijk}^{q-r}(\phi) B_{ijk}^r(\vec{\phi})$$



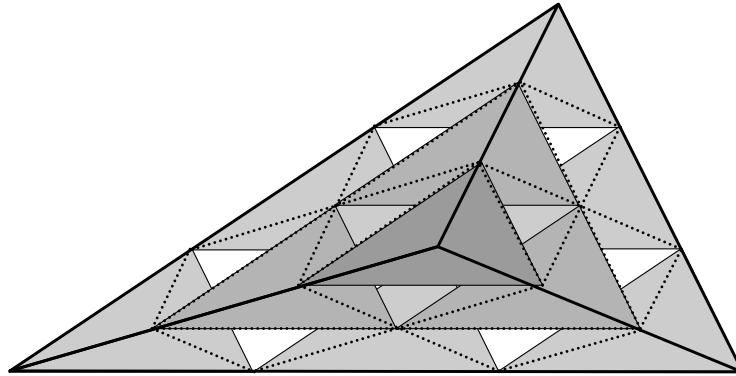


Abbildung 1.9: Umrechnung von Bézier-Bernstein-Koeffizienten auf Clough-Tochter-Mikrodreiecke

Insbesondere ergibt sich, dass die drei de Casteljau-Koeffizienten der vorletzten Iteration die Tangentialebene von  $p$  im Punkt  $\phi$  bestimmen, siehe auch Abbildung 1.8. Für den Fall, dass  $\phi$  hierbei auf einer Kante des zugrunde liegenden Dreiecks liegt, spielen für die Bestimmung der Tangentialebene offensichtlich nur diejenigen Bézier-Bernstein-Koeffizienten von  $p$  mit Abstand  $\leq 1$  zu dieser Kante eine Rolle, denn mit  $\phi_3 = 0$  verschwindet der letzte Summand in der Rekursionsformel für die de Casteljau-Koeffizienten. In gleicher Weise wird deutlich, dass die drei Koeffizienten mit Abstand  $\leq 1$  zu einem Eckpunkt die Tangentialebene in diesem Eckpunkt bestimmen.

Eine weitere wichtige Anwendung des Algorithmus von de Casteljau besteht darin, bivariate Polynome von einem Dreieck auf ein anderes umzurechnen:

**Satz 1.9**

Gegeben zwei bivariate Polynome  $p_l \equiv \sum_{i+j+k=q} (c_l)_{ijk} B_{ijk}^q$ ,  $l = 1, 2$ , auf zwei Dreiecken  $T_l = [u, v, w_l]$  mit gemeinsamer Kante  $[u, v]$ , dann gilt:

$$(c_2)_{ijk} = c_{ij_0}^k(\phi), \quad i + j + k = q,$$

wobei  $\phi$  den baryzentrischen Koordinatenvektor von  $w_2$  bezüglich  $T_1$  bezeichnet.

Auch dieser Satz lässt sich geometrisch veranschaulichen: Um die transformierten Koeffizienten  $(c_2)_{ijk}$  zu erhalten, wendet man den Algorithmus von de Casteljau auf die Stelle  $\phi$  an und behält in jedem Rekursionsschritt  $r$  die de Casteljau-Koeffizienten  $c_{ij_0}^r(\phi)$ .

Eine nützliche Eigenschaft dieses Zusammenhangs zeigt sich bei einem Clough-Tochter-Split eines Dreiecks, auf dem ein bivariates Polynom  $p$  definiert ist. Hier liefern die Koeffizienten des de Casteljau-Algorithmus direkt die Bézier-Bernstein-Koeffizienten von  $p$  für die einzelnen Mikrodreiecke, siehe Abbildung 1.9.

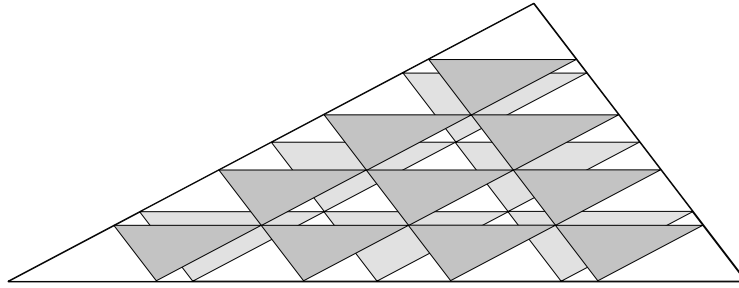


Abbildung 1.10: Umrechnung von Bézier-Bernstein-Koeffizienten bei Erhöhung des Polynomgrads

Da für  $0 \leq i < j$  der Raum  $\mathcal{P}_i^2$  in  $\mathcal{P}_j^2$  enthalten ist, lässt sich ein beliebiges Polynom  $p \in \mathcal{P}_i^2$  auch als  $j$ -gradiges Polynom schreiben. Eine solche Transformation in der Darstellung von  $p$  nennt sich *Erhöhung des Grades*. Dabei lässt sich die höhergradige Bézier-Bernstein-Darstellung in einfacher Weise wie folgt iterativ bestimmen:

**Lemma 1.10**

Sei  $p \equiv \sum_{i+j+k=q} c_{ijk} B_{ijk}^q$  ein bivariates Polynom vom totalen Grad  $q$ . Dann ist es möglich,  $p$  als Polynom vom Grad  $q + 1$  durch  $\sum_{i+j+k=q+1} \tilde{c}_{ijk} B_{ijk}^{q+1}$  darzustellen, wobei sich die Koeffizienten der höhergradigen Darstellung wie folgt berechnen:

$$\tilde{c}_{ijk} = \frac{1}{q+1} (i c_{i-1,j,k} + j c_{i,j-1,k} + k c_{i,j,k-1}), \quad i+j+k = q+1$$

Wir bemerken, dass zwar negative Koordinaten in der Rekursionsformel auftreten, die entsprechenden Summanden jedoch verschwinden. Der Zusammenhang hat eine anschauliche geometrische Interpretation: Trägt man über dem zu  $p$  gehörenden Dreieck  $T$  das  $q$ -gradige Bézier-Bernstein-Netz ab, so entsprechen die  $(q+1)$ -gradigen Koeffizienten den Auswertungen dieses Netzes an den äquidistanten Bézier-Bernstein-Punkten. Hierbei betrachten wir den stückweise linearen Spline, welcher die Netzpunkte interpoliert. Bei der grafischen Darstellung der Berechnung der Koeffizienten der  $(q+1)$ -gradigen Darstellung aus den Koeffizienten der  $q$ -gradigen Darstellung ergeben sich „umgedrehte“, zu  $T$  ähnliche Dreiecke. Insbesondere wird auch deutlich, dass die zu den Eckpunkten von  $T$  gehörenden Bézier-Bernstein-Koeffizienten unverändert bleiben. Ein Beispiel für die Erhöhung des Grades von 4 auf 5 zeigt Abbildung 1.10.

Dieser Zusammenhang ist von großer Bedeutung für die computertechnische Darstellung bivariater Polynome. Es zeigt sich, dass Bézier-Bernstein-Netze von ausreichend hohem Grad eine gute Näherung des Polynoms liefern:

**Satz 1.11**

Sei für ein bivariates Polynomstück  $p|_T \in \mathcal{P}_q^2$  über einem Dreieck  $T$  mit  $(\mathcal{N}_n)_{n \geq q}$  die Folge der Bézier-Bernstein-Netze gegeben. Dann konvergiert diese Folge gegen den Graphen von  $p$ , d. h. es gilt:

$$\lim_{n \rightarrow \infty} \mathcal{N}_n = G(p).$$

## Kapitel 2

# Splines auf Quadrangulierungen

In diesem Kapitel betrachten wir  $C^1$ -Splines von beliebigem Grad auf triangulierten Quadrangulierungen. In der derzeitigen Literatur entwickelten Nürnberger, Schumaker und Zeilfelder Interpolationsmethoden zunächst für Klassen spezieller Quadrangulierungen, so genannten *checkerboard quadrangulations* [66] und *separable quadrangulations* [67], später auf beliebigen strikt konvexen Quadrangulierungen [68]. Letztere Methode basiert auf dem Raum  $\mathcal{S}_3^1(\diamond)$  der kubischen  $C^1$ -Splines auf einer Triangulierung  $\diamond$ , die sich durch ein Hinzufügen von Diagonalen zu den Vierecken der Quadrangulierung ergibt. Dabei werden die Vierecke schwarzweiß gefärbt, sodass benachbarte gleichfarbige Vierecke stets Ketten bilden, siehe Abbildung 1 auf Seite 6. Es zeigt sich, dass das Verfahren effizienter wird, falls in der Färbung kurze Ketten gleichfarbiger Vierecke auftreten und dass Interpolanten höhere Lokaltätseigenschaften aufweisen, falls keine geschlossenen Ketten vorkommen. Ziel ist daher eine Schwarzweißfärbung von Quadrangulierungen, sodass gleichfarbige Vierecke ausschließlich nicht geschlossene Ketten von möglichst kurzer Länge bilden. Für Triangulierungen geben Nürnberger und Zeilfelder [72] einen einfachen Algorithmus, bei welchem benachbarte gleichfarbige Dreiecke höchstens Paare bilden. Die Erfahrung zeigt, dass die Entwicklung entsprechender Algorithmen für beliebige Quadrangulierungen ein komplexes Problem ist. In der von Nürnberger, Schumaker und Zeilfelder [68] vorgestellten Methode wird eine beliebige Schwarzweißfärbung solange modifiziert, bis benachbarte gleichfarbige Vierecke ausschließlich in Ketten auftreten, siehe Algorithmus 2.9 auf Seite 47. Es ist leicht zu zeigen, dass dieser Algorithmus in linearer Komplexität terminiert. Allerdings treten dabei lange Ketten gleichfarbiger Vierecke und auch geschlossene Ketten auf. Wir durchlaufen die zu färbende Quadrangulierung in geeigneter Weise und färben dabei Vierecke, sodass entstehende Ketten möglichst früh unterbrochen werden, was im Allgemeinen zu einer kürzeren Länge der Ketten führt. Bei einer solchen Vorgehensweise tritt die folgende Situation auf: Sind Vierecke mit mindestens einem schwarzen und mindestens einem weißen Nachbarn zu färben, so ist, grob gesprochen, eine Entstehung von geschlossenen Ketten und T-Systemen, vgl. Abbildung 2.2, möglich. Des Weiteren

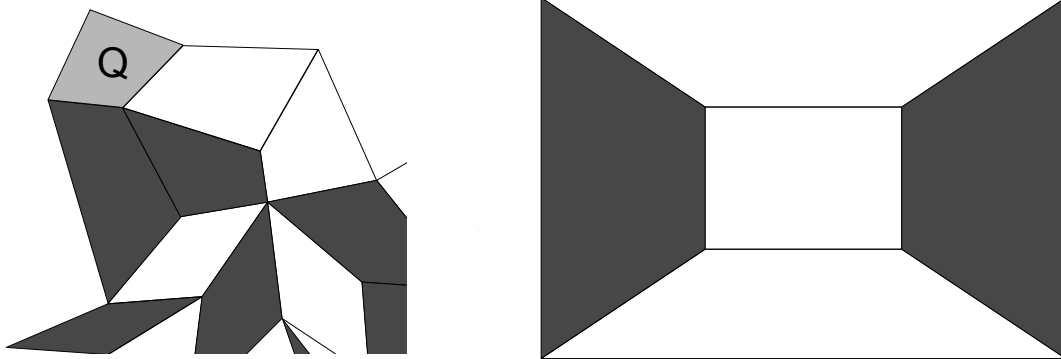


Abbildung 2.1: Links: Bei einer Färbung des Dreiecks  $Q$  vergrößert sich die maximale Kettenlänge. Rechts: Beispiel für eine nur mit Kettenlänge drei färbbare Quadrangulierung

ren soll so gefärbt werden, dass sich die maximal auftretende Kettenlänge möglichst nicht erhöht. Abbildung 2.1 (links) zeigt eine Situation, in welcher sich eine Verlängerung der Ketten nicht verhindern lässt. Abbildung 2.1 (rechts) zeigt eine einfache Quadrangulierung, bei der für jede mögliche Schwarzweißfärbung mindestens Kettenlänge drei auftritt.

Wir entwickeln in Abschnitt 2.2 einen Färbungsalgorithmus, mit dem sich die Vierecke einer beliebigen Quadrangulierung erstmals so schwarzweiß färben lassen, dass gleichfarbige benachbarte Vierecke stets Ketten von im Allgemeinen kürzerer Länge als in [68] bilden, die darüber hinaus nicht geschlossen sind. Grob gesprochen können die bereits entstandenen Ketten dabei in den meisten Fällen gestoppt werden. In seltenen Fällen ist nicht vermeidbar, dass im Laufe der Färbung so genannte T-Systeme gleichfarbiger Vierecke auftreten. Wir entwickeln diesbezüglich einen effizienten rekursiven Teilalgorithmus, welcher die bestehende Teilfärbung so abändert, dass gleichfarbige benachbarte Vierecke danach wieder ausschließlich nicht geschlossene Ketten bilden. Der Algorithmus terminiert in linearer Komplexität. Mit dem entwickelten Verfahren ergeben sich für vorgegebene maximale Kettenlängen große Klassen von Quadrangulierungen, sodass sich Ketten stets stoppen lassen, bevor diese Länge überschritten wird. Dabei erhält man bereits für maximale Kettenlänge zwei eine allgemeine Klasse, welche insbesondere Quadrangulierungen mit vielen Eckpunkten vom Grad drei enthält, was bei der Färbung schwierig ist. Abbildung 2.11 auf Seite 46 zeigt eine Quadrangulierung dieser Klasse.

In Abschnitt 2.3 verwenden wir den entwickelten Färbungsalgorithmus für die lokale Lagrange-Interpolation, basierend auf dem Raum  $\mathcal{S}_3^1(\diamond)$  der bivariaten kubischen  $C^1$ -Splines auf den gefärbten triangulierten Quadrangulierungen  $\diamond$ . Dabei ergeben sich eine höhere Effizienz und stärkere Lokalitätseigenschaften des Verfahrens als in [68], außerdem vereinfacht sich die Vorgehensweise. Im Anschluss verallgemeinern wir die Methode auf  $C^1$ -Splines von beliebigem Grad  $q \geq 3$ . Wir zeigen,

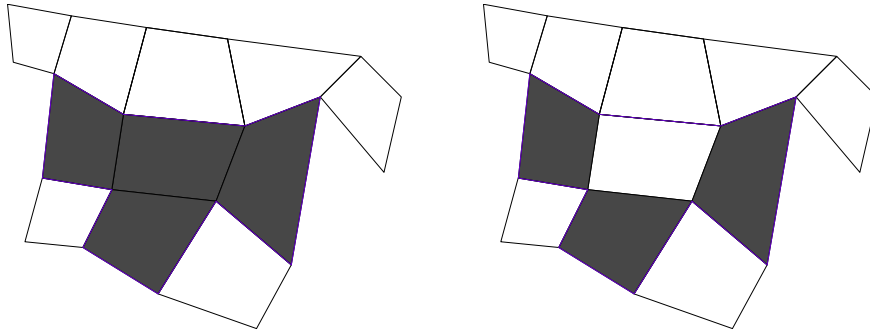


Abbildung 2.2: Auflösung eines T-Systems

dass die konstruierten Interpolanten lokal und stabil bestimmbar sind und optimale Approximationsordnung besitzen. Es handelt sich dabei um umfangreiche Beweise. Schließlich untersuchen wir, wie sich verschiedene Färbungen auf wesentliche Interpolationseigenschaften auswirken.

In Abschnitt 2.4 entwickeln wir ein weiteres Verfahren für die lokale Interpolation mit  $C^1$ -Splines vom Grad  $q \geq 3$  auf beliebigen strikt konvexen Quadrangulierungen, wobei erstmals keine Färbung der Vierecke benötigt wird. Die Reihenfolge, in welcher auf den Vierecken Interpolationspunkte gewählt werden, wird dabei durch ein Prioritätsprinzip festgelegt. Ähnliche Prinzipien wurden bislang für Triangulierungen [10] und für Tetraederzerlegungen [69], nicht aber für Quadrangulierungen entwickelt. Bei der in dieser Arbeit entwickelten Methode werden stets diejenigen Vierecke aus  $\diamond$  zuerst gewählt, welche mit den bereits durchlaufenen Vierecken möglichst wenige Kanten gemeinsam haben. Auch dieses Verfahren liefert lokal und stabil bestimmbare Interpolanten von optimaler Approximationsordnung.

Zuletzt stellen wir in Abschnitt 2.5 wesentliche Eigenschaften der Verfahren wie Effizienz, Lokalität und Dimension entsprechender Interpolationsräume gegenüber. Es zeigt sich, dass beide entwickelten Methoden, besonders die Verwendung des Färbungsalgorithmus, Vorteile gegenüber [68] besitzen.

## 2.1 Grundlagen

Wir führen zunächst einige grundlegende Begriffe ein. In erzeugten Färbungen treten gleichfarbige, benachbarte Vierecke lediglich in *Ketten* auf. Bei einer *nicht geschlossenen Kette* von Vierecken der Länge  $l \geq 1$  handelt es sich um ein System

$$\mathcal{C} := (Q_1, \dots, Q_n)$$

von Vierecken mit der Eigenschaft, dass in  $\mathcal{C} \times \mathcal{C}$  genau die Paare  $\{Q_i, Q_{i+1}\}$  benachbart sind für alle  $i = 1, \dots, n - 1$ . Falls außerdem  $Q_1$  und  $Q_n$  benachbart sind,

so heißt  $\mathcal{C}$  *geschlossene Kette*. In beiden Fällen heißt  $n$  *Länge* der Kette. Weiterhin heißt jedes Viereck aus  $\mathcal{C}$  mit genau einem Nachbarn aus  $\mathcal{C}$  *Endpunkt*. Schließlich heißt ein Viereck  $Q_i$ ,  $i \in \{1, \dots, n\}$ , einer Kette *gerade*, falls der Index  $i$  gerade ist, ansonsten heißt es *ungerade*. Ein Beispiel zeigt Abbildung 1 auf Seite 6.

Während der Färbung treten in seltenen Fällen vorübergehend auch *T-Systeme* gleichfarbiger Vierecke auf. Dabei handelt es sich um eine Menge  $\mathcal{T}$  benachbarter Vierecke mit folgenden Eigenschaften:

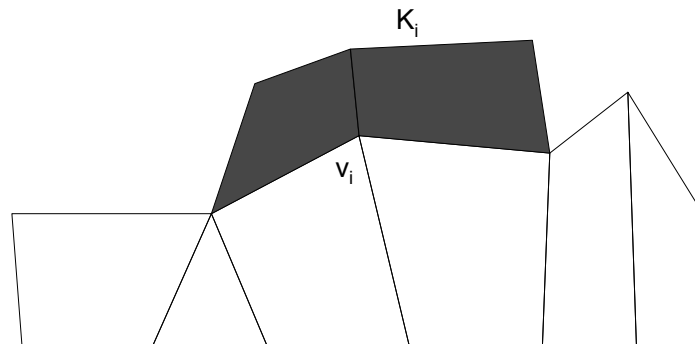
- (i) Genau ein Viereck  $Q_T \in \mathcal{T}$ , genannt *Kreuzungspunkt*, besitzt genau drei Nachbarn in  $\mathcal{T}$ .
- (ii) Genau drei Vierecke in  $\mathcal{T}$ , genannt *Endpunkte*, besitzen jeweils genau einen Nachbarn in  $\mathcal{T}$ .
- (iii) Existieren weitere Vierecke in  $\mathcal{T}$ , so besitzt jedes davon genau zwei Nachbarn in  $\mathcal{T}$ .

Liegt nun in einer Quadrangulierung  $\diamond$  ein T-System  $\mathcal{T}$  gleichfarbiger Vierecke vor, so sagen wir,  $\mathcal{T}$  *wird aufgelöst*, falls die Färbung so geändert wird, dass in  $\mathcal{T}$  nur noch Ketten gleicher Farbe auftreten. Wir bemerken, dass durch die Farbwechsel ein weiteres T-System gleichfarbiger Vierecke in  $\diamond$  entstehen kann, ein Beispiel zeigt Abbildung 2.2.

Im folgenden Abschnitt werden Quadrangulierungen erzeugt, indem an eine Startzelle sukzessive Kegel angefügt werden: Ein *Kegel*  $K$  ist eine Quadrangulierung mit Randeckpunkt  $v$ , welcher Eckpunkt jedes Vierecks  $Q \in K$  ist. Der Eckpunkt  $v$  heißt hierbei *Scheitelpunkt* von  $K$ .

Wir beschreiben nun, auf welche Weise Kegel an eine bestehende Quadrangulierung angefügt werden: Sei  $v$  ein Randeckpunkt einer Quadrangulierung  $\diamond$ . Ferner sei  $K$  ein Kegel um  $v$  mit  $\diamond \cap K = \emptyset$ , sodass  $\tilde{\diamond} = \diamond \cup K$  eine Quadrangulierung mit innerem Eckpunkt  $v$  ist. Dann sagen wir, wir erhalten  $\tilde{\diamond}$  durch *Anfügen von  $K$  an  $\diamond$  bzw. zu  $v$* . Ist  $K$  einelementig, so sprechen wir auch vom Anfügen des (einzelnen) Vierecks  $Q \in K$ . Bezeichne  $k$  die Anzahl der Kanten, die  $\diamond$  und  $K$  gemeinsam haben. Dann nennen wir das Anfügen auch *Anfügen mit  $k$  gemeinsamen Kanten*. Da  $v$  nach dem Anfügen innerer Eckpunkt von  $\tilde{\diamond}$  ist, gilt stets  $k \geq 2$ .

In den erzeugten Färbungen sollen benachbarte gleichfarbige Vierecke möglichst kurze Ketten bilden. Insbesondere sollen möglichst wenige Kanten  $E$  existieren, welche zwei gleichfarbigen Vierecken angehören. Eine Kante mit dieser Eigenschaft nennen wir *nicht passend*, andernfalls heißt sie *passend*. Wir nennen die Färbung einer Quadrangulierung *abwechselnd schwarzweiß*, falls alle Kanten passend sind.

Abbildung 2.3: Anfügen eines Kegels  $K_i$ 

## 2.2 Algorithmen für Quadrangulierungen

### 2.2.1 Konstruktion

Mit dem folgenden Algorithmus entwickeln wir eine Methode zur Konstruktion von Quadrangulierungen.

#### Algorithmus 2.1

Man konstruiere eine Quadrangulierung  $\diamond$  wie folgt: Ausgehend von einer Zelle  $\diamond_0 = K_0$  von Vierecken mit innerem Eckpunkt  $v_0$  füge man für  $i = 1, \dots, n$  Kegel  $K_i$  an Eckpunkte  $v_i$  von  $\diamond_{i-1}$  an und bezeichne die entstehenden Teilquadrangulierungen mit  $\diamond_i$ .

Nach diesem Schritt entferne man eventuelle Hilfsvierecke am Rand und bezeichne die resultierende Quadrangulierung mit  $\diamond$ .

Dieser Algorithmus ist nicht nur für die Konstruktion, sondern auch für den Durchlauf von beliebigen vorgegebenen Quadrangulierungen geeignet:

#### Theorem 2.2

*Mit Algorithmus 2.1 kann jede vorgegebene, beliebige strikt konvexe Quadrangulierung  $\diamond$  durchlaufen werden.*

**Beweis:** Offensichtlich ist es möglich, mit einer beliebigen Zelle aus  $\diamond$  bzw., falls eine solche nicht existiert, mit einer Zelle, welche möglichst viele Vierecke aus  $\diamond$  enthält, zu starten und dann sukzessive alle verbleibenden Vierecke in Form von geeigneten Kegeln anzufügen. Man durchlaufe  $\diamond$  dabei wie folgt:

1. Man wähle einen inneren Eckpunkt  $v_0$  von  $\diamond$  bzw. einen Randeckpunkt, falls  $\diamond$  keinen inneren Eckpunkt enthält. Man wähle als Startzelle  $\diamond_0$  die Zelle aus  $\diamond$



- um  $v_0$  bzw., falls  $v_0$  Randeckpunkt ist, eine Zelle, welche alle Vierecke aus  $\diamond$  mit Eckpunkt  $v_0$  enthält.
2. Solange nicht hinzugefügte Vierecke in  $\diamond$  existieren, gehe man für  $i = 1, \dots, n$  wie folgt vor: Man wähle einen Eckpunkt  $v_i$ , den ein nicht hinzugefügtes Viereck mit der bestehenden Teilquadrangulierung  $\diamond_{i-1}$  gemeinsam hat. Dabei wähle man, falls möglich, einen inneren Eckpunkt von  $\diamond$ , ansonsten einen Randeckpunkt. Als anzufügenden Kegel wähle man alle in  $\diamond \setminus \diamond_{i-1}$  enthaltenen Vierecke mit Eckpunkt  $v_i$ , falls  $v_i$  innerer Eckpunkt von  $\diamond$  ist. Ist  $v_i$  ein Randeckpunkt von  $\diamond$ , so wähle man einen Kegel um  $v_i$ , welcher alle diese Vierecke enthält. Man wähle  $v_i$  dabei so, dass die entstehende Teilquadrangulierung  $\diamond_i$  ein einfach zusammenhängendes Gebiet  $\Omega_{\diamond_i}$  besitzt. Da das zu  $\diamond$  gehörende Gebiet  $\Omega_{\diamond}$  einfach zusammenhängend ist, ist eine solche Wahl stets möglich.
  3. Falls eine nichtleere Menge  $\diamond_n \setminus \diamond$  von Hilfsvierecken existiert, entferne man diese Vierecke.

□

Die Färbung eines mit mindestens drei gemeinsamen Kanten angefügten Kegels  $K$  wird im Färbungsalgorithmus nur für den Fall  $|K| = 1$  behandelt. Dies ist keine Beschränkung der Allgemeinheit, denn für  $|K| \geq 2$  können statt einem Anfügen von  $K$  mit demselben Ergebnis nacheinander einzelne Vierecke  $Q \in K$  angefügt werden, bis ein Kegel  $\tilde{K} \subseteq K$  mit lediglich zwei gemeinsamen Kanten zur entsprechenden Teilquadrangulierung verbleibt. Dies zeigen wir mit dem folgenden

**Lemma 2.3**

*Sei  $K$  ein Kegel von mindestens zwei Vierecken, der an eine Quadrangulierung  $\diamond$  angefügt wird und mit dieser mindestens drei gemeinsame Kanten besitzt. Dann lässt sich  $K$  in ein Tupel  $(K_1, \dots, K_n)$  von Kegeln mit folgenden Eigenschaften zerlegen, sodass  $K_1, \dots, K_n$  nacheinander zu  $\diamond$  angefügt werden können:*

- (i) *Es ist  $|K_i| = 1$  für alle  $i = 1, \dots, n - 1$ .*
- (ii) *Weiterhin ist  $|K_n| = 1$ , oder  $K_n$  wird mit genau zwei gemeinsamen Kanten angefügt.*

**Beweis:** Seien  $v$  der Eckpunkt, an welchen  $K$  angefügt wird, und  $N = |K|$ . O. B. d. A. seien die Vierecke von  $K$  im Uhrzeigersinn nummeriert von  $Q_1$  bis  $Q_N$ , d. h. es ist

$$K = \{Q_1, \dots, Q_N\}.$$

Ferner sei mit  $Q_0$  das Viereck aus  $\diamond$  bezeichnet, das zu  $Q_1$  benachbart ist und  $v$  als Eckpunkt besitzt. Entsprechend sei mit  $Q_{N+1}$  das Viereck aus  $\diamond$  bezeichnet, das zu  $Q_N$  benachbart ist und  $v$  als Eckpunkt besitzt. Wegen der Definition des Anfügens von Kegeln sind diese Vierecke eindeutig existent. Hierbei ist genau dann  $Q_0 = Q_{N+1}$ , wenn  $v$  Eckpunkt eines einzigen Vierecks aus  $\diamond$  ist.

Jedes Viereck  $Q_j \in K$  besitzt genau zwei Kanten  $\mathcal{M}_j = \{E_j^1, E_j^2\}$ , welche  $v$  nicht enthalten. Für  $j = 1, \dots, N$  bezeichnen wir  $Q_j$  als vom Typ A, falls beide Kanten aus  $\mathcal{M}_j$  Kanten von  $\diamond$  sind. Ist genau eine Kante aus  $\mathcal{M}_j$  Kante von  $\diamond$ , so sei  $Q_j$  vom Typ B. Schließlich sei  $Q_j$  vom Typ C, falls keine der Kanten aus  $\mathcal{M}_j$  Kante von  $\diamond$  ist.

Es sei  $\mathcal{A}$  die Menge aller Vierecke aus  $K$  vom Typ A,  $\mathcal{B}$  die Menge der Vierecke vom Typ B und  $\mathcal{C}$  die Menge der Vierecke vom Typ C. Die Mengen  $\mathcal{A}, \mathcal{B}, \mathcal{C}$  bilden demnach eine Zerlegung von  $K$ :

$$K = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}, \quad \mathcal{A}, \mathcal{B}, \mathcal{C} \text{ paarweise disjunkt}$$

Da  $\Omega_\diamond$  einfach zusammenhängend ist, existiert mindestens eine Randkante von  $K$ , welche keine Kante von  $\diamond$  ist (Aussage I). Da ferner  $\Omega_{\diamond_{i+1}}$  einfach zusammenhängend ist, bildet die Menge der Randkanten von  $K$ , welche keine Kanten von  $\diamond$  sind, eine einfache polygonale Kette (Aussage II). Da  $K$  und  $\diamond$  mindestens drei gemeinsame Kanten besitzen, besitzen sie mindestens eine gemeinsame Kante, welche Eckpunkt  $v$  nicht enthält (Aussage III).

Wegen diesen Aussagen lässt sich  $\mathcal{A}$  in zwei Mengen  $\mathcal{A}_1$  sowie  $\mathcal{A}_2$  zerlegen und  $\mathcal{B}$  in zwei Mengen  $\mathcal{B}_1$  sowie  $\mathcal{B}_2$ , sodass folgende Eigenschaften gelten:

- (a) Ist  $\mathcal{A}_j \neq \emptyset$  für  $j = 1, 2$ , dann ist  $\mathcal{A}_j$  benachbart.
- (b) Ist  $\mathcal{A}_1 \neq \emptyset$ , dann ist  $\mathcal{A}_1$  benachbart zu  $Q_0$ . Ist  $\mathcal{A}_2 \neq \emptyset$ , dann ist  $\mathcal{A}_2$  benachbart zu  $Q_{N+1}$ .
- (c) Ist  $\mathcal{B}_j \neq \emptyset$  für  $j = 1, 2$ , dann ist  $|\mathcal{B}_j| = 1$ .
- (d) Ist  $\mathcal{B}_1 \neq \emptyset$ , dann ist  $\mathcal{B}_1$  benachbart zu  $\mathcal{A}_1$ , falls  $\mathcal{A}_1$  nichtleer ist, andernfalls zu  $Q_0$ .
- (e) Ist  $\mathcal{B}_2 \neq \emptyset$ , dann ist  $\mathcal{B}_2$  benachbart zu  $\mathcal{A}_2$ , falls  $\mathcal{A}_2$  nichtleer ist, andernfalls zu  $Q_{N+1}$ .

Aus den Eigenschaften folgt:

- (f) Ist  $\mathcal{C} \neq \emptyset$ , so ist  $\mathcal{C}$  benachbart.

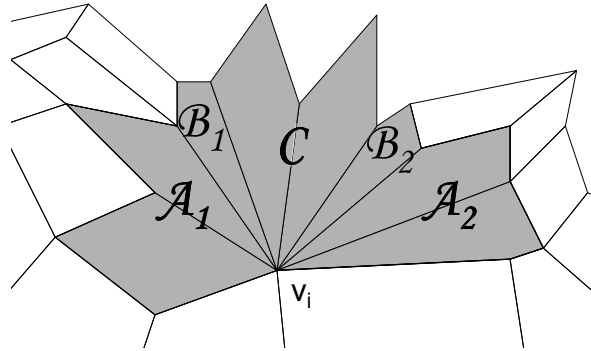


Abbildung 2.4: Zerlegung eines Kegels  $K$  in die Mengen  $\mathcal{A}_1, \mathcal{B}_1, \mathcal{C}, \mathcal{B}_2$  und  $\mathcal{A}_2$

- (g) Für  $j = 1, 2$  gilt: Ist  $\mathcal{C} \neq \emptyset$ , so ist  $\mathcal{C}$  benachbart zu  $\mathcal{B}_j$ , falls  $\mathcal{B}_j$  nichtleer ist, andernfalls zu  $\mathcal{A}_j$ , falls  $\mathcal{A}_j$  nichtleer ist. Ist auch  $\mathcal{A}_j$  leer, so ist  $\mathcal{C}$  benachbart zu  $Q_0$  für  $j = 1$  bzw. zu  $Q_{N+1}$  für  $j = 2$ .
- (h) Wegen Aussage III gilt:  $\mathcal{A} \cup \mathcal{B} \neq \emptyset$
- (k) Wegen Aussage I gilt:  $\mathcal{B} \cup \mathcal{C} \neq \emptyset$

Wir haben also  $K$  im Uhrzeigersinn zerlegt in

$$K = \mathcal{A}_1 \cup \mathcal{B}_1 \cup \mathcal{C} \cup \mathcal{B}_2 \cup \mathcal{A}_2,$$

wobei die folgenden Eigenschaften gelten:

$$\mathcal{A} \cup \mathcal{B} \neq \emptyset, \quad \mathcal{B} \cup \mathcal{C} \neq \emptyset, \quad |\mathcal{B}_j| \leq 1 \quad \text{für } j = 1, 2$$

Abbildung 2.4 veranschaulicht diese Notation.

In den folgenden Schritten fügen wir sukzessive die Vierecke aus diesen Mengen an  $\diamond$  an. Die jeweils aktuelle Quadrangulierung sei dabei mit  $\diamond$  bezeichnet.

1. Solange noch nicht alle Vierecke aus  $\mathcal{A}$  angefügt worden sind, existiert ein noch nicht angefügtes Viereck  $Q \in \mathcal{A}$  mit genau einer Kante  $E \notin \mathcal{E}^{[\diamond]}$ . Dies begründen wir wie folgt: Jedes Viereck vom Typ  $A$  besitzt definitionsgemäß zwei Kanten aus  $\mathcal{E}^{[\diamond]}$ . Angenommen es existiert ein noch nicht angefügtes Viereck  $Q \in \mathcal{A}$ . Wegen den Aussagen (a) und (b) existiert dann ein noch nicht angefügtes Viereck  $Q \in \mathcal{A}$  mit einer dritten Kante in  $\mathcal{E}^{[\diamond]}$ . Wegen Aussage (k) besitzt  $Q$  eine Kante, die nicht in  $\mathcal{E}^{[\diamond]}$  ist, insgesamt also genau drei Kanten in  $\mathcal{E}^{[\diamond]}$ .

Folglich lassen sich sukzessive alle Vierecke aus  $\mathcal{A}$  als einelementige Kegel anfügen. Dabei existieren stets jeweils drei gemeinsame Kanten.

2. Solange ein noch nicht angefügtes Viereck  $Q \in \mathcal{B}$  existiert, besitzt  $Q$  mindestens zwei benachbarte Kanten  $E_1$  und  $E_2$ , maximal aber drei Kanten in  $\mathcal{E}^{[\diamond]}$ . Dies begründen wir wie folgt: Definitionsgemäß besitzt jedes Viereck vom Typ  $B$  eine Kante, die nicht in  $\mathcal{E}^{[\diamond]}$  liegt und eine Kante  $E$ , die in  $\mathcal{E}^{[\diamond]}$  liegt. Da alle Vierecke aus  $\mathcal{A}$  bereits angefügt wurden, folgt mit den Aussagen (c) bis (e) sowie Aussage II, dass  $Q$  eine zu  $E$  benachbarte Kante aus  $\mathcal{E}^{[\diamond]}$  besitzt. Folglich lassen sich sukzessive alle Vierecke aus  $\mathcal{B}$  als einelementige Kegel  $K$  anfügen. Dabei existieren stets zwei oder drei gemeinsame Kanten.
3. Für den Fall  $\mathcal{C} = \emptyset$  ist nichts zu zeigen. Andernfalls ist  $\mathcal{C} \subset K$  wegen (f) ein Kegel. Da alle Vierecke aus  $\mathcal{A}$  und  $\mathcal{B}$  bereits angefügt wurden, folgt mit (g), dass die beiden Randkanten von  $\mathcal{C}$ , die  $v$  enthalten, auch Kanten von  $\diamond$  sind. Da alle Vierecke aus  $\mathcal{C}$  vom Typ  $C$  sind, sind alle anderen Kanten von  $\mathcal{C}$  keine Kanten von  $\diamond$ .
- Folglich lässt sich der Kegel  $\mathcal{C}$  mit zwei gemeinsamen Kanten an die bestehende Quadrangulierung  $\diamond$  anfügen.

Diese Vorgehensweise liefert eine Zerlegung von  $K$ , welche (i) und (ii) erfüllt. Im Fall  $\mathcal{C} \neq \emptyset$  ist  $\mathcal{C} = K_n$ . Ist  $\mathcal{C}$  leer, so ist  $K_n$  einelementig und enthält das zuletzt angefügte Viereck aus  $\mathcal{B}$ .  $\square$

## 2.2.2 Färbung

Im Folgenden entwickeln wir einen Algorithmus zur Schwarzweißfärbung von wie in Algorithmus 2.1 konstruierten bzw. durchlaufenen Quadrangulierungen.

Dabei betrachten wir Färbungen mit Kettenlängen  $\geq 2$ , denn ansonsten ist der Fall offensichtlich: Eine Quadrangulierung lässt sich genau dann, in einfacher Weise, abwechselnd schwarzweiß färben, wenn jeder innere Eckpunkt von geradem Grad ist.

Wir konstruieren bzw. durchlaufen eine Quadrangulierung wie in Algorithmus 2.1 beschrieben und färben angefügte Vierecke wie folgt:

### Algorithmus 2.4

Man betrachte  $K_0$ : Falls  $|K_0|$  gerade ist, färbe man abwechselnd schwarzweiß, andernfalls färbe man, sodass genau eine Kante nicht passend ist.

Im Anschluss färbe man sukzessive jeden angefügten Kegel  $K_i$ ,  $i = 1, \dots, n$ .

Besitzt  $K_i$  dabei *mindestens zwei Kegel und drei gemeinsame Kanten* zu  $\diamond_{i-1}$ , so färbe man die entsprechenden Teilkegel gemäß Lemma 2.3.

Andernfalls betrachten wir drei Fälle:

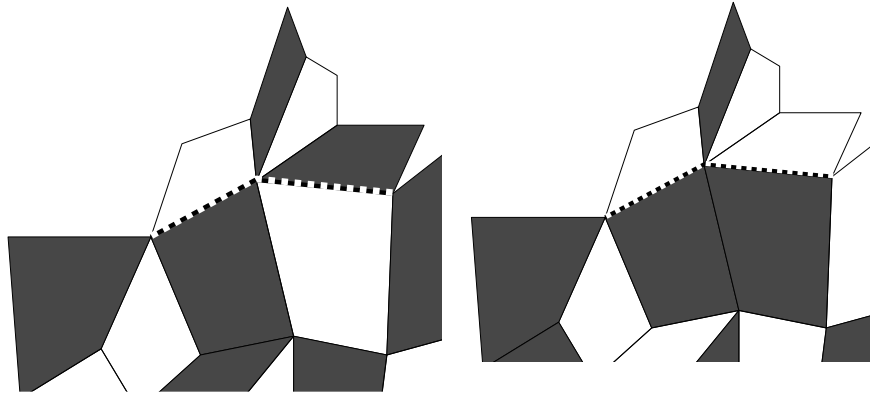


Abbildung 2.5: Färbung von Kegeln mit gerader Anzahl von Vierecken

Fall 1:  $|K_i|$  gerade:

Wegen Lemma 2.3 besitzen  $K_i$  und  $\diamond_{i-1}$  genau zwei gemeinsame Kanten. Existieren zwei zu  $K_i$  benachbarte Vierecke in  $\diamond_{i-1}$  mit unterschiedlicher Farbe, dann lässt sich  $K_i$  so schwarzweiß färben, dass alle Kanten in  $K_i$  mit Eckpunkt  $v_i$  passend sind, siehe Abbildung 2.5 (links). Andernfalls färbe man die beiden Vierecke aus  $K_i$  mit Nachbarn aus  $\diamond_{i-1}$ , sodass die gemeinsamen Kanten von  $K_i$  und  $\diamond_{i-1}$  passend sind. Die verbleibenden Vierecke aus  $K_i$  lassen sich so schwarz-weiß färben, dass genau eine der Kanten in  $K_i$  mit Eckpunkt  $v_i$  nicht passend ist, siehe Abbildung 2.5 (rechts).

Fall 2:  $|K_i|$  ungerade,  $|K_i| \geq 3$ :

Wegen Lemma 2.3 besitzen  $K_i$  und  $\diamond_{i-1}$  genau zwei gemeinsame Kanten. Existieren zwei zu  $K_i$  benachbarte Vierecke in  $\diamond_{i-1}$  mit unterschiedlichen Farben, dann färbe man die beiden Vierecke aus  $K_i$  mit Nachbarn in  $\diamond_{i-1}$ , sodass die gemeinsamen Kanten von  $K_i$  und  $\diamond_{i-1}$  passend sind. Die verbleibenden Vierecke aus  $K_i$  lassen sich so schwarzweiß färben, dass genau eine der Kanten in  $K_i$  mit Eckpunkt  $v_i$  nicht passend ist, siehe Abbildung 2.6 (rechts). Andernfalls lässt sich  $K_i$  so schwarzweiß färben, dass alle Kanten in  $K_i$  mit Eckpunkt  $v_i$  passend sind, siehe Abbildung 2.6 (links).

Fall 3:  $|K_i| = 1$ :

Wir behandeln zunächst den Fall, dass  $K_i = \{Q_1\}$  und  $\diamond_{i-1}$  genau zwei gemeinsame Kanten besitzen. Haben die in  $\diamond_{i-1}$  zu  $Q_1$  benachbarten Vierecke  $Q_0$  und  $Q_2$  dieselbe Farbe, so färbe man  $Q_1$  in der anderen Farbe.

Andernfalls färbe man  $Q_1$ , sodass es mit seinen gleichfarbigen Nachbarn eine Kette  $\mathcal{C}$  bildet. Ist dies für beide Farbwahlen von  $Q_1$  gegeben, so wähle man diejenige Farbe, für welche die zugehörige Kette  $\mathcal{C}$  kürzer ist. Ist die entstehende Länge für beide

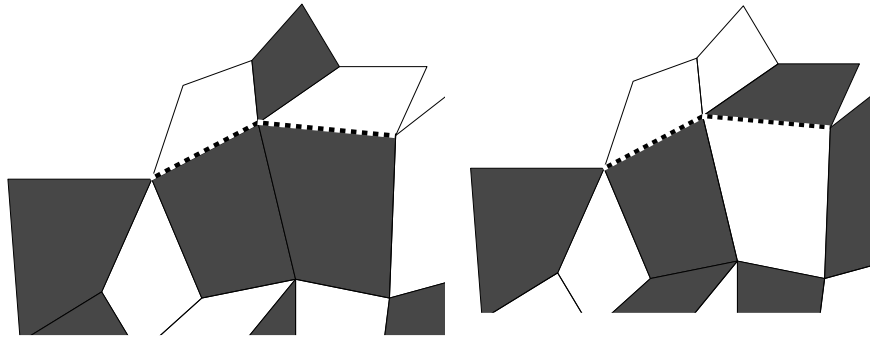


Abbildung 2.6: Färbung von Kegeln mit ungerader Anzahl von Vierecken

Farbwahlen gleich, so wähle man diejenige Farbe, für welche  $\mathcal{C}$  möglichst wenige Randkanten bezüglich der entstehenden Quadrangulierung  $\diamond_i$  hat.

Bildet andernfalls  $Q_1$  bei beiden Farbwahlen mit seinen gleichfarbigen Nachbarn keine Kette, dann entsteht wegen Theorem 2.6, (i) und (ii) in beiden Fällen ein T-System  $\mathcal{T}$ . Ein solches T-System kann offensichtlich nur in Teilquadrangulierungen mit maximaler Kettenlänge  $\geq 3$  auftreten. Man färbe in diesem Fall  $Q_1$  weiß und gehe wie folgt vor, siehe Abbildung 2.7:

Eines der beiden Vierecke  $Q_0$  und  $Q_2$  ist Kreuzungspunkt  $Q_T$  von  $\mathcal{T}$ . Folglich ist  $Q_T$  mit  $Q_1$  und genau zwei weiteren Vierecken  $Q_3$  und  $Q_4$  gleicher Farbe benachbart. Man wechsele nun die Farbe von  $Q_T$ . Offensichtlich löst der Farbwechsel das T-System auf. Existiert kein vierter Nachbar  $Q_5$  von  $Q_T$ , so treten in der resultierenden Färbung ausschließlich nicht geschlossene Ketten auf. Existiert jedoch ein vierter Nachbar  $Q_5$ , so hat dieser, da  $Q_1$  vor dem Farbwechsel T-System war, die neue Farbe von  $Q_T$ . Enthält  $\diamond_i$  nach dem Farbwechsel ein T-System mit Kreuzungspunkt  $Q_5$ , so verfähre man mit  $Q_5$  rekursiv in derselben Weise.

Im Anschluss stelle man die ursprüngliche Färbung von  $\diamond_{i-1}$  wieder her und verfähre analog mit der Ausgangsfarbe schwarz für  $Q_1$ . Man wähle diejenige Färbung, die in der möglichst kürzeren maximalen Kettenlänge in  $\diamond_i$  resultiert.

Es bleibt der Fall, dass  $K_i$  und  $\diamond_{i-1}$  drei Kanten gemeinsam haben.  $Q_1$  hat dann genau drei Nachbarn in  $\diamond_{i-1}$ . Besitzen alle Nachbarn dieselbe Farbe, so färbe man  $Q_1$  in der anderen Farbe.

Andernfalls färbe man  $Q_1$  in der Farbe, die genau ein Nachbar von  $Q_1$  besitzt. Entsteht dabei ein T-System, so verfähre man analog zum Fall des Anfügens mit zwei gemeinsamen Kanten.

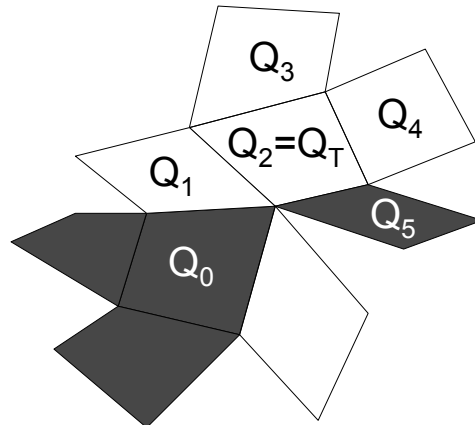


Abbildung 2.7: Für beide Farbwahlen des angefügten einzelnen Vierecks  $Q_1$  entsteht ein T-System in  $\diamond_i$ .

Wir zeigen zunächst, dass der rekursive Teilalgorithmus terminiert. Dabei verwenden wir ähnliche Argumente wie Nürnberger, Schumaker und Zeilfelder [68]:

**Lemma 2.5**

*Die rekursive Vorgehensweise aus Algorithmus 2.4, Fall 3, terminiert in linearer Komplexität.*

**Beweis:** Sei  $Q_T$  ein Viereck, welches im Verlauf der Rekursion die Farbe wechselt. Da  $Q_T$  Kreuzungspunkt eines T-Systems ist, besitzt  $Q_T$  genau drei gleichfarbige und maximal einen andersfarbigen Nachbarn in  $\diamond_i$ . Pro Farbwechsel nimmt daher die Anzahl der nicht passenden Kanten in  $\diamond_i$  um mindestens zwei ab. Daher terminiert der Rekursionsalgorithmus in linearer Komplexität zu  $|\diamond|$ .  $\square$

Wir zeigen nun wichtige Eigenschaften gefärbter Quadrangulierungen:

**Theorem 2.6**

*Mit Algorithmus 2.4 gefärbte Quadrangulierungen besitzen die folgenden Eigenschaften:*

- (i) *Mengen gleichfarbiger benachbarter Vierecke sind stets Ketten.*
- (ii) *Ketten gleichfarbiger benachbarter Vierecke sind stets nicht geschlossen.*
- (iii) *Beim Anfügen eines Kegels vergrößert sich die maximale Kettenlänge höchstens um eins.*

- (iv) Eine solche Vergrößerung ist nur in Fall 3 möglich, es sei denn die Vergrößerung ist von eins auf zwei, was nicht von Bedeutung ist, da stets Quadrangulierungen mit maximaler Kettenlänge  $\geq 2$  betrachtet werden.

**Beweis:** Wir zeigen die Aussagen (i) bis (iv) induktiv. Zunächst bemerken wir, dass ein Kegel eine nicht geschlossene Kette von Vierecken ist. Die Aussagen (i) und (ii) gelten somit für jeden Kegel  $K \subseteq \diamond$ .

Die Färbung von  $K_0$  enthält maximal zwei benachbarte gleichfarbige Vierecke, sodass (i) bis (iv) für  $\diamond_0$  erfüllt sind. Seien nun die Aussagen (i) und (ii) nun für die Teilquadrangulierung  $\diamond_{i-1}$ ,  $i \in \mathbb{N}$ , erfüllt. Folglich gehören die zu  $K_i$  benachbarten Vierecke in  $\diamond_{i-1}$  nicht geschlossenen Ketten an. Die maximale Länge dieser Ketten sei mit  $l$  bezeichnet.

Wir betrachten zunächst die Fälle 1 und 2. Hier existieren genau zwei Kanten, die  $\diamond_{i-1}$  und  $K_i$  gemeinsam haben. In beiden Fällen wird so gefärbt, dass diese zwei Kanten passend bleiben. Ferner wird  $K_i$  mit maximal einer nicht passenden inneren Kante gefärbt. Also bleiben die benachbarten Mengen gleichfarbiger Vierecke in  $\diamond_i$  nicht geschlossene Ketten, und die maximale Kettenlänge vergrößert sich nicht oder von eins auf zwei. Es folgen die Aussagen (i) bis (iv) für die Fälle 1 und 2.

Es bleibt Fall 3, und wir betrachten die folgenden Aussagen:

- (a) Laut Voraussetzung bilden gleichfarbige Vierecke in  $\diamond_{i-1}$  ausschließlich nicht geschlossene Ketten.
- (b)  $\diamond_i$  resultiert aus  $\diamond_{i-1}$  durch Anfügen genau eines Vierecks  $Q_1$ .
- (c)  $Q_1$  besitzt maximal einen gleichfarbigen Nachbarn in der bestehenden Quadrangulierung  $\diamond_{i-1}$ .

Aufgrund dieser Aussagen ist in  $\diamond_i$  das Viereck  $Q_1$

1. entweder in einer nicht geschlossenen Kette
2. oder in einem T-System  $\mathcal{T}$

von Vierecken gleicher Farbe enthalten.

Bilde  $Q_1$  zunächst eine nicht geschlossene Kette gleicher Farbe. Es folgen die Aussagen (i) und (ii). Aufgrund von Aussage (c) vergrößert sich  $l$  um eins oder bleibt unverändert. Es folgt Aussage (iii) für Fall 3. Aussage (iv) ist offensichtlich erfüllt.

Bilde  $Q_1$  nun ein T-System gleicher Farbe. Die Aussagen (i) und (ii) ergeben sich unmittelbar aus Lemma 2.5. Aussage (iv) ist offensichtlich erfüllt. Es bleibt Aussage (iii): Mit der Entstehung des T-Systems erhöht sich die maximale Kettenlänge



zunächst nicht. Nach dem Farbwechsel eines Kreuzungspunkts  $Q_T$  in einem Rekursionsschritt besitzt  $Q_T$  maximal einen Nachbarn  $\tilde{Q}_T$  gleicher Farbe, d. h. die maximale Kettenlänge erhöht sich um maximal eins. Diese Erhöhung ist ferner nur beim letzten Farbwechsel vor der Terminierung relevant. Andernfalls nämlich wäre  $\tilde{Q}_T$  im folgenden Rekursionsschritt Kreuzungspunkt des neu entstandenen T-Bausteins und würde deshalb die Farbe wechseln. Insbesondere folgt Aussage (iii) für  $\diamond_i$ .

Also gelten die Aussagen (i) und (ii) für  $\diamond_i$  und die Aussagen (iii) und (iv) für das Anfügen von  $K_i$ . Per Induktion folgt die Korrektheit der Aussagen für  $\diamond$ .  $\square$

Wir geben im Folgenden eine obere Schranke für die Anzahl der Rekursionsschritte des rekursiven Teilalgorithmus:

**Lemma 2.7**

*Für die Anzahl der Rekursionsschritte ist  $\frac{E}{4}$  eine obere Schranke.*

**Beweis:** Sei  $\diamond$  eine beliebige Quadrangulierung mit einer Färbung, in der gleichfarbige benachbarte Vierecke stets in nicht geschlossenen Ketten auftreten. Ferner sei  $\mathcal{E}_D$  ein Tupel, das die inneren Kanten von  $\diamond$  doppelt, die Randkanten einmal und keine weiteren Elemente enthält, und wir setzen  $E_D := |\mathcal{E}_D|$ . Schließlich bezeichnen wir mit  $\mu_D$  bzw.  $\mu$  die Anzahl der passenden Kanten in  $\mathcal{E}_D$  bzw.  $\mathcal{E}$  und mit  $\nu_D$  bzw.  $\nu$  die Anzahl der nicht passenden Kanten in  $\mathcal{E}_D$  bzw.  $\mathcal{E}$ .

Da in  $\diamond$  gleichfarbige Vierecke stets in Ketten auftreten, besitzt jedes Viereck  $Q \in \diamond$  mindestens genauso viele passende Kanten wie nicht passende. Da Ketten gleichfarbiger Vierecke in  $\diamond$  nicht geschlossen sind, existieren mit den Endpunkten der nicht geschlossenen Ketten folglich Vierecke  $Q \in \diamond$ , die mehr passende Kanten besitzen als nicht passende. Da jede innere Kante von  $\diamond$  in genau zwei Vierecken  $Q_1, Q_2 \in \diamond$  enthalten ist und jede äußere Kante in genau einem, folgt mit der Definition von  $\mathcal{E}_D$

$$\mu_D > \nu_D. \tag{2.1}$$

Da  $\mathcal{E}_B$  die inneren Kanten zweimal enthält, ist

$$E_D = E + E_B.$$

Diese Gleichung impliziert wegen der Tatsache, dass alle Randkanten  $\mathcal{E}_B$  passend sind

$$\begin{aligned} \mu_D &= \mu + (\mu - E_B) = 2\mu - E_B \\ \nu_D &= \nu + (\nu - 0) = 2\nu \end{aligned}$$

Eingesetzt in Gleichung (2.1) erhält man

$$(2\mu - E_B) > (2\nu - 0).$$

In jeder Quadrangulierung ist  $E_B \geq 4$ . Es folgt  $2\mu > 2\nu + 4$  und schließlich  $\mu > \nu + 2$ . Wegen Bemerkung 2.6 treten in einer Teilquadrangulierung  $\diamond_i$ , abgesehen von einem einzigen T-System, gleichfarbige Vierecke stets in Ketten auf. Demnach besitzt genau ein Viereck  $Q_T \in \diamond_i$  mehr nicht passende Kanten als passende. Überträgt man obige Abschätzung für die Färbung von  $\diamond$  auf die Färbung von  $\diamond_i$ , so existiert hier folglich genau eine passende Kante weniger und eine nicht passende Kante mehr. Bezeichnet  $\dot{\mu}$  die Anzahl der passenden und  $\dot{\nu}$  die Anzahl der nicht passenden Kanten in  $\diamond_i$ , so folgt demnach mit  $(\dot{\mu} + 1) > (\dot{\nu} - 1) + 2$  schließlich

$$\dot{\mu} > \dot{\nu},$$

d. h. es ist mehr als die Hälfte der Kanten in  $\diamond_i$  passend.

Zu Beginn der Rekursion ist folglich die Anzahl  $\dot{\nu}$  der nicht passenden Kanten in  $\diamond_i$  kleiner als  $\frac{E}{2}$ . Da diese Anzahl laut Beweis zu Lemma 2.5 pro Iterationsschritt um mindestens zwei abnimmt, finden weniger als  $\frac{E}{4}$  Rekursionsschritte statt.  $\square$

Abschließend sei bemerkt, dass ein erstes T-System, wie in Fall 3 des Färbungsalgorithmus beschrieben, selten entsteht. Eine Rekursion mehrerer Farbwechsel tritt nur in sehr seltenen Fällen auf. Beliebige Quadrangulierungen werden mit Algorithmus 2.4 daher in linearer Komplexität gefärbt.

### 2.2.3 Klassen von Quadrangulierungen

Abhängig von der maximalen Kettenlänge definieren wir im Folgenden Klassen von Quadrangulierungen: Für  $\hat{l} \in \mathbb{N}$ ,  $\hat{l} \geq 2$ , bezeichne  $\mathcal{K}_{\hat{l}}$  die Menge aller durch Algorithmus 2.4 konstruierbaren Quadrangulierungen mit maximalen Kettenlängen  $\leq \hat{l}$ .

#### Die Klasse mit maximaler Kettenlänge 2

Im Folgenden zeigen wir, dass die Klasse  $\mathcal{K}_2$  eine allgemeine Klasse von natürlichen Quadrangulierungen ist.

Wie im Folgenden beschrieben, lässt sich bei der Konstruktion einer Quadrangulierung  $\diamond \in \mathcal{K}_2$  durch ein geeignetes Anfügen von Kegeln erzielen, dass innere Eckpunkte von vorgegebenen Graden entstehen. Auf diese Weise ergibt sich, dass  $\mathcal{K}_2$  Quadrangulierungen unterschiedlichster Geometrie enthält.

Es werden dabei ausschließlich *natürliche Quadrangulierungen* konstruiert: Dabei handelt es sich, grob gesprochen, um Zerlegungen, in welchen weder nahezu degenerierte Vierecke noch Vierecke von stark unterschiedlicher Größe enthalten sind.

Mit folgender Konstruktionsregel erzeugen wir durch geeignetes Anfügen von Kegeln an eine beliebige Teilquadrangulierung  $\diamond_{i-1} \in \mathcal{K}_2$  einen inneren Eckpunkt von gegebenem Grad  $d$ , sodass die resultierende Quadrangulierung ebenfalls in  $\mathcal{K}_2$  liegt.

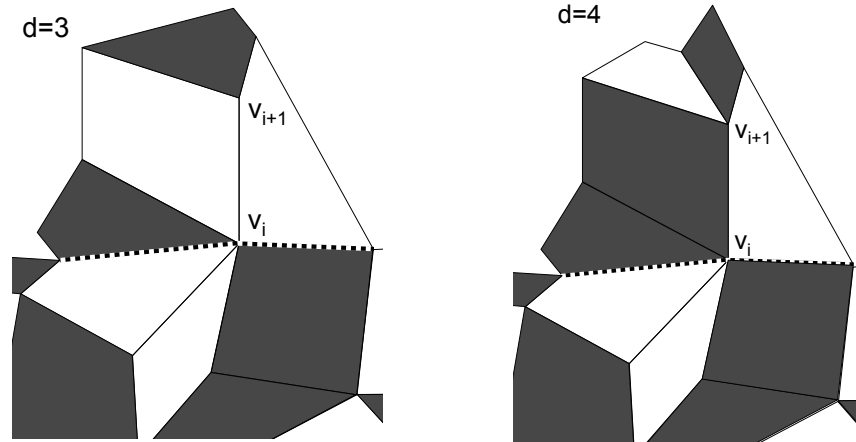


Abbildung 2.8: Erzeugung eines inneren Eckpunkts vom Grad 3 (links) bzw. Grad 4 (rechts).

1. Für die Konstruktion eines Eckpunkts vom Grad 3 kann wie folgt vorgegangen werden. Bei den Möglichkeiten 1.1 und 1.2 wird zu Randeckpunkten mit bestimmten Eigenschaften angefügt, bei Möglichkeit 1.3 zu beliebigen Randeckpunkten.

1.1 Existiert ein Randeckpunkt  $v_i$  von  $\diamond_{i-1}$  mit den folgenden Eigenschaften, so füge man einen Kegel  $K_i = \{Q\}$  zu  $v_i$  an:

- (a) Der zu  $v_i$  gehörende Innenwinkel von  $\diamond_{i-1}$  ist  $> 180^\circ$ .
- (b) Genau zwei Vierecke  $Q_0, Q_2 \in \diamond_{i-1}$  besitzen  $v_i$  als Eckpunkt, wobei mindestens eine der folgenden Bedingungen erfüllt ist:
  - (i)  $Q_0$  und  $Q_2$  sind gleichfarbig.
  - (ii)  $Q_0$  besitzt keinen gleichfarbigen Nachbarn in  $\diamond_{i-1}$ .
  - (iii)  $Q_2$  besitzt keinen gleichfarbigen Nachbarn in  $\diamond_{i-1}$ .

Ein solches Anfügen ist offensichtlich stets möglich und führt zu einer Färbung, sodass  $\diamond_i \in \mathcal{K}_2$  ist.

1.2 Falls ein Randeckpunkt  $v_i$  von  $\diamond_{i-1}$  existiert, welcher Eckpunkt eines einzigen Vierecks ist, füge man einen Kegel  $K_i = \{Q_1, Q_2\}$  mit zwei gemeinsamen Kanten zu  $v_i$  an. Durch Algorithmus 2.4 werden  $Q_1$  und  $Q_2$  so gefärbt, dass diese beiden Kanten passend sind. Auf diese Weise ist die resultierende Quadrangulierung in  $\mathcal{K}_2$ .

1.3 Man füge zunächst einen Kegel  $K_i$  von mindestens zwei Vierecken zu einem beliebigen Randeckpunkt  $\tilde{v}_i \in \mathcal{V}_B^{[\diamond_{i-1}]}$  mit zwei gemeinsamen Kanten an, sodass mindestens ein Randeckpunkt  $v_i \in \mathcal{V}_B^{[\diamond_i]}$  mit Eigenschaften wie in 1.1 entsteht. Offensichtlich ist eine solche Konstruktion stets möglich.

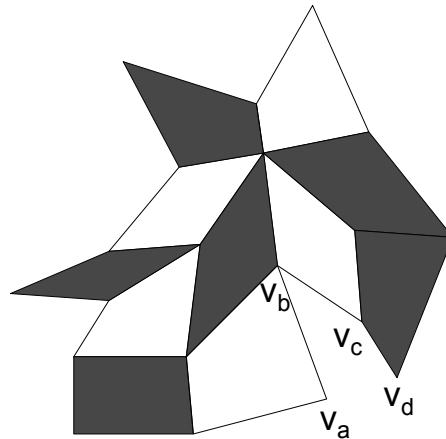


Abbildung 2.9: Fall zweier benachbarter Randeckpunkte  $v_b$  und  $v_c$  mit Innenwinkeln  $> 180^\circ$  in  $\diamond_{i-1}$

Die Vierecke aus  $K_i$  werden dann so gefärbt, dass die resultierende Quadrangulierung  $\diamond_i$  in  $\mathcal{K}_2$  ist. Im Anschluss füge man einen gefärbten Kegel  $K_{i+1} = \{Q\}$  zu  $v_i$  wie in 1.1 beschrieben an. Man erhält dadurch einen inneren Eckpunkt vom Grad drei, wobei die resultierende Quadrangulierung  $\diamond_{i+1}$  in  $\mathcal{K}_2$  liegt. Ein Beispiel zeigt Abbildung 2.8 (links).

2. Für die Konstruktion eines inneren Eckpunkts  $v_i$  vom Grad  $d \geq 4$  füge man entweder einen Kegel von mindestens zwei Vierecken zu einem geeigneten Eckpunkt  $v_i$  an, oder man füge ein einzelnes Viereck zu einem Eckpunkt  $v_i$  mit entsprechenden Eigenschaften wie in 1.1 an. In beiden Fällen ist die resultierende Quadrangulierung offensichtlich in  $\mathcal{K}_2$ .

Weiterhin gibt es die Möglichkeit, ähnlich wie in 1.3 vorzugehen: Zunächst füge man einen Kegel von mindestens zwei Vierecken zu einem Eckpunkt  $v_i$  an, sodass zu einem entstehenden Randeckpunkt  $v_{i+1}$  angefügt werden kann und der Grad von  $v_{i+1}$  nach dem Anfügen  $d$  beträgt. Auch hier wird offensichtlich so gefärbt, dass eine Quadrangulierung in  $\mathcal{K}_2$  resultiert. Ein Beispiel zeigt Abbildung 2.8 (rechts) für  $d = 4$ .

Auch in den unter 2. geschilderten Fällen entstehen Färbungen, in denen gleichfarbige Vierecke stets nicht geschlossene Ketten von maximaler Länge zwei bilden. Die Argumentation aus 1. ist hierbei analog anwendbar.

3. Es bleibt der Fall eines Anfügens mit drei gemeinsamen Kanten. In diesem Fall wird an einen von zwei benachbarten Eckpunkten angefügt, an denen die Teilquadrangulierung  $\diamond_{i-1}$  Innenwinkel  $> 180^\circ$  besitzt. Zwar treten bei der Konstruktion von  $\diamond$  nicht notwendigerweise solche benachbarte Paare von Randeckpunkten auf, doch auch in diesem Fall kann mit maximaler Kettenlänge zwei gefärbt werden. Bei einem Anfügen zu benachbarten Eckpunkten

mit Innenwinkeln  $> 180$  Grad sind die Farben dreier umgebender Vierecke zu beachten, siehe Abbildung 2.9.

Seien  $v_b$  und  $v_c$  zwei Randeckpunkte mit Innenwinkeln  $> 180^\circ$  in  $\diamond_{i-1}$ , und seien mit  $v_a$  und  $v_d$  die beiden zu  $\{v_b, v_c\}$  benachbarten Randeckpunkte bezeichnet, sodass  $v_a$  und  $v_b$  sowie  $v_c$  und  $v_d$  benachbart sind. Ferner sei mit  $Q_1$  das Viereck in  $\diamond_{i-1}$  mit Eckpunkten  $v_a$  und  $v_b$ , mit  $Q_2$  das Viereck mit Eckpunkten  $v_b$  und  $v_c$  und mit  $Q_3$  das Viereck mit Eckpunkten  $v_c$  und  $v_d$  bezeichnet. Da  $\diamond_{i-1}$  strikt konvex ist, sind diese drei Vierecke paarweise verschieden und eindeutig.

3.1 Besitzen  $Q_1$ ,  $Q_2$  sowie  $Q_3$  dieselbe Farbe, so füge man  $K_i = \{[v_a, v_b, v_c, v_d]\}$  zu  $v_b$  an und färbe  $K_i$  in der anderen Farbe.

3.2 Besitzen nicht alle Vierecke  $Q_1, Q_2, Q_3$  dieselbe Farbe, bezeichne  $Q$  dasjenige Viereck  $Q_i$ ,  $i \in \{1, 2, 3\}$ , das nicht die Farbe der anderen beiden Vierecke besitzt.

Besitzt  $Q$  dabei keine gleichfarbigen Nachbarn in  $\diamond_i$ , so füge man  $K_i$  wie in 3.1 definiert zu  $v_b$  an und färbe  $K_i$  in der Farbe von  $Q$ .

Andernfalls sei  $X$  die Farbe von  $Q$  und  $Y$  die andere Farbe. Man verfare wie folgt:

- Ist  $Q \in \{Q_1, Q_3\}$ , so betrachten wir zunächst den Fall  $Q = Q_3$ , siehe Abbildung 2.10 (links). Es sei  $v_e = \frac{1}{2}(v_a + v_d)$ . Außerdem sei  $v_f = v_c + 2 \cdot [\frac{1}{2}(v_d + v_e) - v_c]$ , d. h. man wählt  $v_f$ , sodass das Viereck  $Q_5 = [v_c, v_d, v_e, v_f]$  ein Trapez bildet.

Man füge zunächst  $Q_4 = [v_a, v_b, v_c, v_e]$  zu  $v_b$  an, und damit erhält  $Q_4$  gemäß Algorithmus 2.4 die Farbe  $X$ . Danach füge man  $\{Q_5\}$  zu  $v_c$  an, und  $Q_5$  erhält die Farbe  $Y$ .

Für  $Q = Q_1$  verfare man analog.

- Ist  $Q = Q_2$  so seien

$$v_e = \frac{2}{3}v_a + \frac{1}{3}v_d \quad \text{und} \quad v_f = \frac{1}{3}v_a + \frac{2}{3}v_d,$$

siehe Abbildung 2.10 (rechts). Nun sei  $g_a$  die Gerade durch  $v_b$  und  $\frac{1}{2}(v_a + v_e)$ . Entsprechend sei  $g_b$  die Gerade durch  $v_c$  und  $\frac{1}{2}(v_d + v_f)$ .

Wir bezeichnen nun mit  $v_g$  den Punkt auf  $g_a$  mit Abstand  $d = \frac{1}{2}(v_a + v_e)$  zum Schnittpunkt der Geraden  $g_a$  und  $(v_a, v_d)$ , der nicht im Dreieck  $[v_a, v_b, v_e]$  liegt. Ebenso bezeichnen wir mit  $v_h$  den Punkt auf  $g_b$  mit Abstand  $d$  zum Schnittpunkt der Geraden  $g_b$  und  $(v_a, v_d)$ , der nicht im Dreieck  $[v_c, v_d, v_f]$  liegt.

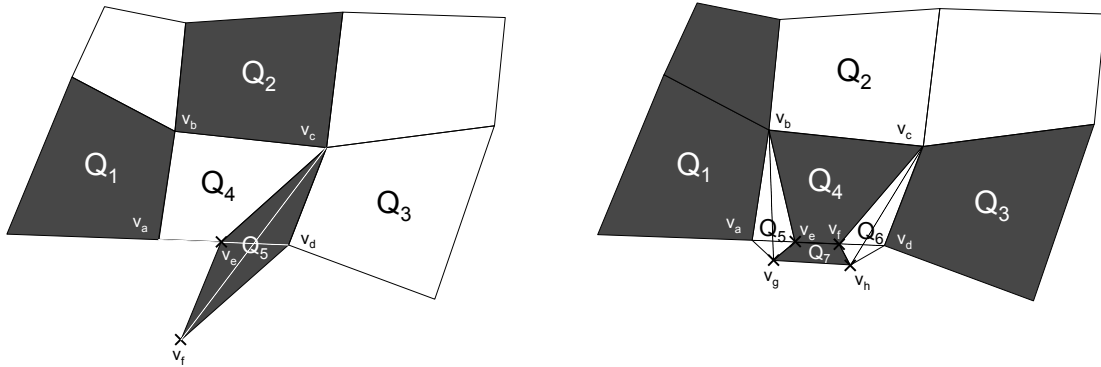


Abbildung 2.10: Anfügen von Kegeln an benachbarte Randeckpunkte mit Innenwinkeln  $> 180^\circ$  in  $\diamond_{i-1}$ , Fall 3.2; Dabei ist die Farbe  $X$  jeweils weiß,  $Y$  schwarz.

Es seien die Vierecke  $Q_4$ ,  $Q_5$ ,  $Q_6$  sowie  $Q_7$  wie folgt definiert:

$$Q_4 = [v_b, v_c, v_f, v_e]$$

$$Q_5 = [v_a, v_b, v_e, v_g]$$

$$Q_6 = [v_c, v_d, v_h, v_f]$$

$$Q_7 = [v_e, v_f, v_h, v_g]$$

Man füge  $\{Q_4, Q_5\}$  zu  $v_b$  an, wobei laut Färbungsalgorithmus  $Q_4$  die Farbe  $Y$  erhält und  $Q_5$  die Farbe  $X$ . Im Anschluss füge man  $\{Q_6\}$  zu  $v_c$  an, und  $Q_6$  erhält die Farbe  $X$ . Zuletzt füge man  $\{Q_7\}$  zu  $v_e$  an, und  $Q_7$  wird mit  $Y$  gefärbt. Beim letzten Anfügen existieren drei gemeinsame Kanten.

In allen Situationen von 3. ist die Konstruktion offensichtlich so gewählt, dass Quadrangulierungen mit maximaler Kettenlänge zwei entstehen.

Abbildung 2.11 gibt ein Beispiel für eine natürliche Quadrangulierung aus  $\mathcal{K}_2$  mit vielen Eckpunkten vom Grad drei.

Offensichtlich gilt für die Klassen  $\mathcal{K}_{\hat{l}}$ ,  $\hat{l} \geq 2$ , die folgende Relation:

$$\mathcal{K}_i \subseteq \mathcal{K}_j, \quad i, j \in \mathbb{N}, \quad 2 \leq i \leq j.$$

Je größer  $\hat{l}$  gewählt wird, desto größer ist die Anzahl der Quadrangulierungen, die sich in ähnlicher Weise wie in 1. bis 3. beschrieben färben lassen.

Somit führt der entwickelte Färbungsalgorithmus für zunehmende  $\hat{l}$  zu immer größeren Klassen von Quadrangulierungen, bei denen Ketten gleichfarbiger Vierecke maximal Länge  $\hat{l}$  besitzen. Insbesondere ergibt sich dabei wie gesehen bereits für  $\hat{l} = 2$  eine allgemeine Klasse.

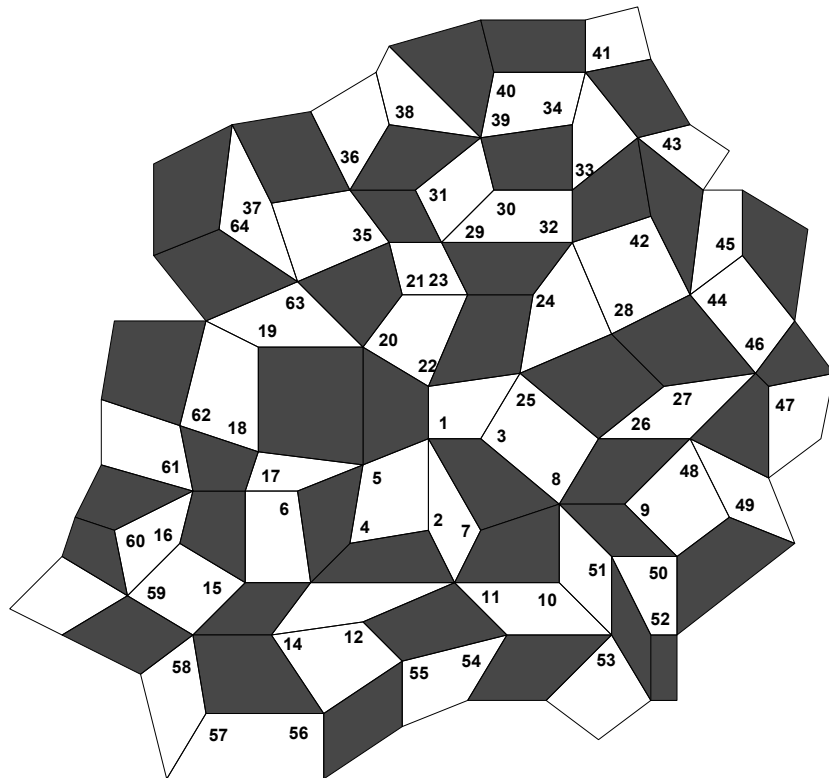


Abbildung 2.11: Eine Quadrangulierung aus  $\mathcal{K}_2$ ; Die Ziffern an den Eckpunkten kennzeichnen die Reihenfolge beim Anfügen der Kegel an die jeweiligen Teilquadrangulierungen wie in Algorithmus 2.1.

## 2.3 Interpolation und Färbung

Wir verwenden die entwickelten Färbungsverfahren nun für die lokale Lagrange-Interpolation mit bivariaten  $C^1$ -Splines auf Quadrangulierungen. Für kubische Interpolanten beschreiben Nürnberger, Schumaker und Zeilfelder [68] eine Interpolationsmethode, basierend auf einer Schwarzweißfärbung der Vierecke. Diese Methode stellen wir in Abschnitt 2.3.1 dar.

In Abschnitt 2.3.2 verallgemeinern wir die Methode auf Splines von beliebigem Grad  $\geq 3$ . Dabei ist die Wahl von Interpolationspunkten im Inneren der Vierecke wesentlich komplexer als im kubischen Fall. Zudem ergeben sich umfangreichere und komplexere Beweise für die Interpolationseigenschaft der konstruierten Mengen und die Stabilität der Methode. Schließlich zeigen wir die optimale Approximationsordnung des Verfahrens.

In Abschnitt 2.3.3 untersuchen wir den Einfluss verwendeter Färbungen auf Interpolationseigenschaften. Wir färben die betrachteten Quadrangulierungen  $\diamond$  mit Algorithmus 2.4 statt mit dem in [68] beschriebenen Färbungsverfahren. Dabei betrach-

ten wir sowohl beliebige Färbungen als auch Färbungen mit maximaler Kettenlänge zwei. In beiden Fällen vereinfacht sich die Methode, und man erhält Interpolanten mit stärkeren Lokalitätseigenschaften. Bei Quadrangulierungen aus  $\mathcal{K}_2$  erhält man darüber hinaus Splineräume von geringerer Dimension.

### 2.3.1 Interpolation mit kubischen Splines

Nürnberg, Schumaker und Zeilfelder [68] geben eine Lösung des folgenden lokalen Lagrange-Interpolationsproblems für kubische  $C^1$ -Splines auf Quadrangulierungen:

**Problem 2.8**

Gegeben sei eine Quadrangulierung  $\diamond$  mit Eckpunkten  $\mathcal{V}$ . Gesucht ist eine geeignete Triangulierung  $\blacklozenge$  von  $\diamond$  und eine Lagrange-Interpolationsmenge  $\mathcal{L}$  für den Spline-raum  $\mathcal{S}_3^1(\blacklozenge)$  mit der Eigenschaft  $\mathcal{V} \subset \mathcal{L}$ .

Die Vorgehensweise ist wie folgt: In einem ersten Schritt werden die Vierecke aus  $\diamond$  durch folgenden einfachen Algorithmus schwarzweiß gefärbt:

**Algorithmus 2.9 (Färbung nach Nürnberg, Schumaker & Zeilfelder)**

Man beginne mit einer beliebigen Schwarzweißfärbung von  $\diamond$ . Solange ein Viereck mit drei oder vier gleichfarbigen Nachbarn existiert, wechsele man die Farbe dieses Vierecks.

Mit ähnlichen Argumenten wie im Beweis zu Lemma 2.5 terminiert dieser Algorithmus in linearer Komplexität und führt zu einer Färbung, in welcher gleichfarbige benachbarte Vierecke stets in Ketten auftreten. Statt mit diesem Algorithmus können die Vierecke auch mit den in Abschnitt 2.2 entwickelten Methoden gefärbt werden.

Im Anschluss wird  $\diamond$  wie folgt trianguliert: In den schwarzen Ketten  $(Q_1, \dots, Q_n)$  fügt man in jedem Viereck mit ungeradem Index eine Diagonale hinzu. In allen verbleibenden Vierecken von  $\diamond$  fügt man beide Diagonalen hinzu. Die resultierende Triangulierung wird mit  $\blacklozenge$  bezeichnet.

Anschließend wird eine Teilmenge der Bézier-Bernstein-Punkte von  $\blacklozenge$  als Interpolationspunkte gewählt. Dabei wählt man zunächst Punkte auf den Kanten der Vierecke:

**Algorithmus 2.10**

Für  $l = 4, \dots, 1$  wiederhole: Solange ein Viereck  $Q \in \blacklozenge$  mit  $l$  nicht gewählten Eckpunkten existiert, wähle man alle Rand-Bézier-Bernstein-Punkte von  $Q$  ausgenommen der bereits gewählten Eckpunkte sowie deren Nachbarn. Die in der Iteration  $l$  durchlaufenen Vierecke seien als vom Typ  $l$  bezeichnet.



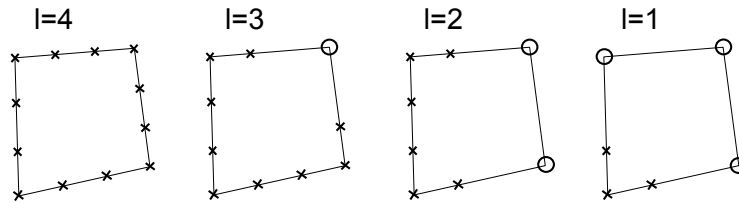


Abbildung 2.12: Die in Algorithmus 2.10 auf den Kanten der Vierecke gewählten Interpolationspunkte; Die bereits gewählten Eckpunkte sind mit „o“ gekennzeichnet.

Es werden in Algorithmus 2.10 also stets Vierecke mit einer größtmöglichen Anzahl  $l$  nicht gewählter Eckpunkte betrachtet, siehe Abbildung 2.12.  $\mathcal{L}_E$  bezeichnet die Menge der gewählten Punkte.

Im Anschluss werden weitere Interpolationspunkte im Inneren der Vierecke gewählt. Wieder handelt es sich bei dieser Wahl um eine Teilmenge der Bézier-Bernstein-Punkte von  $\diamond$ :

### Algorithmus 2.11

Man wähle eine Teilmenge der Bézier-Bernstein-Punkte von  $\diamond$  als Interpolationspunkte wie folgt:

1. Für jedes benachbarte Paar zweigeteilter schwarzer Vierecke wähle man das Baryzentrum von genau einem Mikrodreieck des Paares. Solche Paare treten in schwarzen geschlossenen Ketten von ungerader Länge auf.
2. Für jedes weitere zweigeteilte Viereck  $Q$  wähle man das Baryzentrum von genau einem Mikrodreieck von  $Q$ .
3. Man wähle zunächst für jedes viergeteilte schwarze Viereck, dann für jedes viergeteilte weiße Viereck mit geradem, zuletzt mit ungeradem Index Interpolationspunkte wie folgt:

Bezeichne  $k$  die Anzahl der Kanten von  $Q$ , die bereits durchlaufenen Nachbarvierecken angehören. Dann ist  $k \in \{1, 2\}$ , falls  $Q$  schwarz ist und  $k \in \{0, \dots, 4\}$ , falls  $Q$  weiß ist. Wir bezeichnen mit  $v_0$  den Schnittpunkt der Diagonalen von  $Q$  und mit  $v_1, \dots, v_4$  die Eckpunkte von  $Q$ , sodass  $[v_i, v_{i+1}]$  eine Kante von  $Q$  ist für alle  $i = 1, 2, 3$ . In Abhängigkeit von  $k$  wähle man folgende Interpolationspunkte:

$k=4$  : Man wähle keine Interpolationspunkte in  $Q$ .

$k=3$  : Man wähle  $v_0$ .

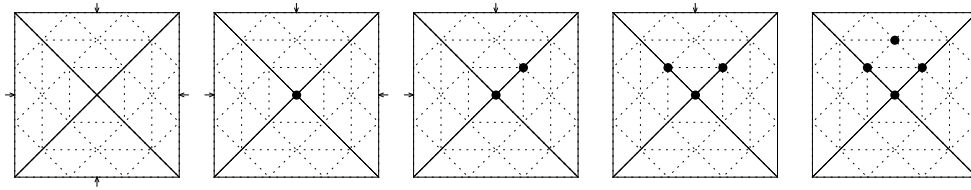


Abbildung 2.13: Die in Algorithmus 2.11 im Inneren der viergeteilten Vierecke gewählten Interpolationspunkte für  $k = 4, 3, 2, 1, 0$ ; Die Kanten, welche bereits durchlaufenen Nachbarvierecken angehören, sind durch Pfeile gekennzeichnet.

$k=2$  : Man wähle  $v_0$  und einen weiteren Punkt  $\frac{2}{3}v_0 + \frac{1}{3}v_i$ ,  $i \in \{1, 2, 3, 4\}$ .

Falls die beiden Kanten, welche bereits durchlaufenen Nachbarvierecken angehören, einen gemeinsamen Punkt  $v_j$ ,  $j \in \{1, 2, 3, 4\}$ , besitzen, ist  $i$  so zu wählen, dass  $|i - j| = 1$  gilt.

$k=1$  : In einem der Mikrodreiecke wähle man alle Bézier-Bernstein-Punkte mit Abstand  $\leq 1$  zu  $v_0$ .

$k=0$  : In einem der Mikrodreiecke wähle man alle Bézier-Bernstein-Punkte mit Abstand  $\leq 1$  zu  $v_0$  und außerdem das Baryzentrum dieses Mikrodreiecks.

Die Menge der in Algorithmus 2.11 gewählten Punkte wird mit  $\mathcal{L}_I$  bezeichnet. Abbildung 2.13 zeigt die im Inneren der viergeteilten Vierecke gewählten Punkte in Abhängigkeit zu bereits durchlaufenen Nachbarvierecken.

Nürnberg, Schumaker und Zeilfelder zeigen folgende Eigenschaften der Interpolationsmethode:

**Theorem 2.12 (Interpolation nach Nürnberg, Schumaker & Zeilfelder)**  
 $\mathcal{L} := \mathcal{L}_E \cup \mathcal{L}_I$  ist eine lokale Lagrange-Interpolationsmenge für den Raum  $\mathcal{S}_q^1(\diamond)$ . Außerdem ist die Interpolationsmethode stabil und besitzt optimale Approximationsordnung.

### 2.3.2 Interpolation mit Splines höheren Grades

Wir verallgemeinern dieses Verfahren nun für Spline-Interpolanten von beliebigem Grad  $q \geq 3$ . Die Wahl von Interpolationspunkten im Inneren der viergeteilten Vierecke ist dabei komplexer als bei einer Beschränkung auf den kubischen Fall. Wir zeigen die Interpolationseigenschaft für die konstruierten Mengen und weiterhin Lokalität und optimale Approximationsordnung der Methode. Die Interpolanten sind

stabil bestimmbar. Diese Beweise sind wesentlich komplexer und umfangreicher als für  $q = 3$ .

Sei eine Quadrangulierung  $\diamond$  wie in Problem 2.8 gegeben und wir lösen das analoge Problem für Splineräume  $\mathcal{S}_q^1(\diamond)$  von beliebigem Grad  $q \geq 3$ .

Wir färben die Vierecke zunächst wie in [68] oder nach dem in Abschnitt 2.2 entwickelten Verfahren. Anschließend triangulieren wir  $\diamond$  wie in Abschnitt 2.3.1 beschrieben und bezeichnen die resultierende Triangulierung mit  $\diamond$ .

Für  $q = 3$  konstruieren wir eine Interpolationsmenge durch die Algorithmen 2.10 und 2.11. Für  $q \geq 4$  wählen wir zunächst folgende Interpolationspunkte auf den Kanten von  $\diamond$ :

**Algorithmus 2.13**

Sei  $q \geq 4$ . Für  $l = 4, 3, 2, 1$  gehe man wie folgt vor: Solange ein Viereck  $Q \in \diamond$  mit  $l$  nicht gewählten Eckpunkten existiert, wähle man alle Rand-Bézier-Bernstein-Punkte von  $Q$  mit Abstand  $\geq 2$  zu den bereits gewählten Eckpunkten.

Nach diesem Schritt wähle man auf jeder Kante  $[a, b]$  von  $\diamond$ , die keinem dieser Vierecke angehört, alle Bézier-Bernstein-Punkte mit Abstand  $\geq 2$  zu  $a$  und  $b$ . Die in der Iteration  $l$  durchlaufenen Vierecke seien als vom Typ  $l$  bezeichnet.

Wie im kubischen Fall werden in Algorithmus 2.13 stets Vierecke mit möglichst vielen nicht gewählten Eckpunkten betrachtet. Wir bemerken, dass es sich bei der Menge  $\mathcal{L}_q^{[E]}$  der gewählten Punkte um eine Teilmenge der Bézier-Bernstein-Punkte von  $\diamond$  handelt.

Anschließend wählen wir weitere Bézier-Bernstein-Punkte im Inneren der Vierecke als Interpolationspunkte. Dabei stellen wir zunächst die auf den Mikrodreiecken der viergeteilten Vierecke zu wählenden Interpolationspunkte dar:

**Algorithmus 2.14**

Sei  $T = [v_0, v_1, v_2]$  ein Mikrodreieck, das entsteht, indem ein strikt konvexes Viereck  $Q = [v_1, v_2, v_3, v_4]$  durch Hinzufügen der Vierecksdiagonalen  $[v_1, v_3]$  und  $[v_2, v_4]$  unterteilt wird. Hierbei bezeichne  $v_0$  den Diagonalschnittpunkt. Sei ferner  $q \geq 4$ .

Man wähle auf  $T$  alle Bézier-Bernstein-Punkte mit Ausnahme von folgenden Punkten als Interpolationspunkte:

1. den Punkten  $\frac{q-1}{q}v_j + \frac{1}{q}v_0$ ,  $j = 1, 2$ , sowie allen Punkten auf  $[v_1, v_2]$ .
2. allen Punkten mit Abstand  $\leq 1$  zu denjenigen Kanten und Eckpunkten von  $T$ , welche bereits durchlaufenen Dreiecken angehören.
3. dem Punkt  $\frac{q-2}{q}v_0 + \frac{1}{q}v_1 + \frac{1}{q}v_2$ , falls  $v_0$  einem durchlaufenen Dreieck angehört.

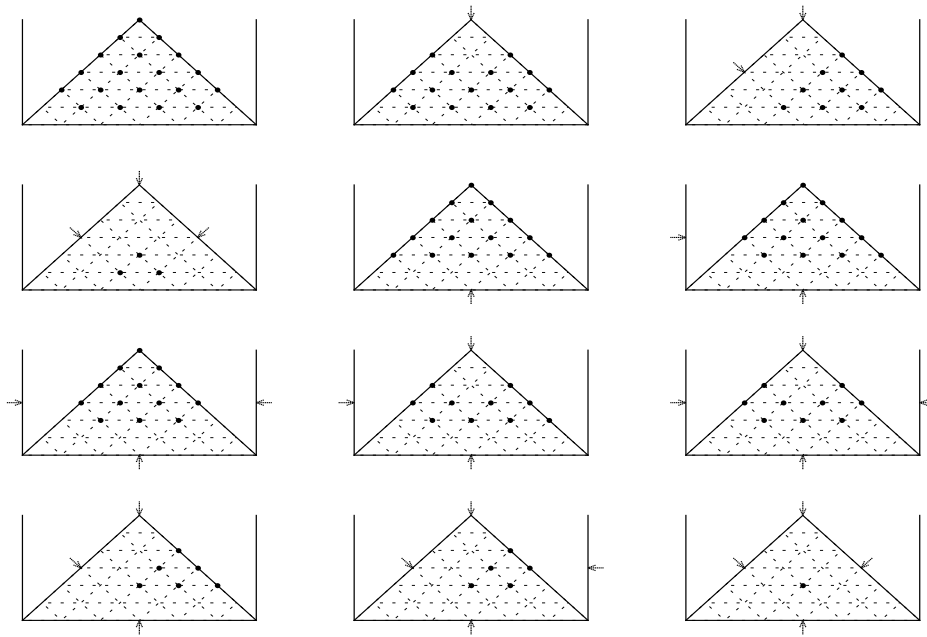


Abbildung 2.14: Interpolationspunkte auf einem Mikrodreieck eines viergeteilten Vierecks für den Fall  $q = 6$ ; Die Kanten und Eckpunkte, welchen bereits durchlaufenen Dreiecken angehören, sind durch Pfeile gekennzeichnet.

4. für  $j = 1, 2$  dem Punkt  $\frac{q-2}{q}v_j + \frac{2}{q}v_0$ , falls beide Kanten  $[v_1, v_2]$  und  $[v_j, v_{5-j}]$  durchlaufenen Dreiecken angehören.

Abbildung 2.14 stellt diese Punktmenge für den Fall  $q = 6$ , abhängig von den Eckpunkten und Kanten, die durchlaufenen Dreiecken angehören, dar.

Mit diesen Punktmenge lässt sich die Wahl weiterer Interpolationspunkte im Inneren der Vierecke in einfacher Weise formulieren:

**Algorithmus 2.15**

Sei  $q \geq 4$ . Man wähle eine Teilmenge der Bézier-Bernstein-Punkte von  $\diamond$  als Interpolationspunkte wie folgt. Dabei bezeichne man jedes betrachtete Mikrodreieck als *durchlaufen*.

1. In jedem zweigeteilten Viereck wähle man alle Bézier-Bernstein-Punkte auf der durchgezogenen Diagonalen  $[a, b]$  mit Abstand  $\geq 2$  zu  $a$  und  $b$ .
2. Für jedes einzelne zweigeteilte Viereck  $Q$  wähle man alle inneren Bézier-Bernstein-Punkte von genau einem Mikrodreieck von  $Q$ . Im zweiten Mikrodreieck wähle man alle inneren Bézier-Bernstein-Punkte mit Abstand  $\geq 2$  zur durchgezogenen Diagonalen von  $Q$ .

3. Für jedes Paar  $\{Q_1, Q_m\}$  zweigeteilter Vierecke mit gemeinsamer Kante  $f$  verfähre man mit Viereck  $Q_1$  wie in 2. Im Mikrodreieck von  $Q_m$  mit Kante  $f$  wähle man alle inneren Bézier-Bernstein-Punkte mit Abstand  $\geq 2$  zu  $f$ . Im zweiten Mikrodreieck von  $Q_m$  wähle man alle inneren Bézier-Bernstein-Punkte mit Abstand  $\geq 2$  zur durchgezogenen Diagonalen von  $Q_m$ . Wir bemerken, dass benachbarte Paare zweigeteilter Vierecke in schwarzen geschlossenen Ketten von ungerader Länge auftreten.
4. Zuletzt wähle man sukzessive Interpolationspunkte auf jedem viergeteilten Viereck  $Q$ . Man beginne mit den schwarzen Vierecken und fahre mit den weißen geraden und zuletzt mit den weißen ungeraden Vierecken fort.  
Für jedes Mikrodreieck  $T$  von  $Q$  wähle man Interpolationspunkte wie in Algorithmus 2.14 beschrieben. Falls Mikrodreiecke  $T$  von  $Q$  existieren, für welche die gemeinsame Kante von  $Q$  und  $T$  einem durchlaufenen Nachbardreieck angehört, so betrachte man diese Mikrodreiecke zuerst.

Wir bezeichnen die Menge der durch Algorithmus 2.15 gewählten Punkte mit  $\mathcal{L}_q^{[I]}$ . Durch die Betrachtung einzelner Mikrodreiecke anstelle der Vierecke verringert sich in den folgenden Beweisen die Anzahl der zu betrachtenden Fälle erheblich. Dabei ist von Bedeutung, dass zunächst diejenigen Mikrodreiecke  $T$  eines viergeteilten Vierecks  $Q$  mit gemeinsamer Kante  $T \cap Q$  zu einem durchlaufenen Nachbardreieck behandelt werden.

Für den Fall  $q = 3$  zeigen Nürnberger, Schumaker und Zeilfelder in einfacher Weise unter ausschließlicher Verwendung von Bézier-Bernstein-Techniken, dass  $\mathcal{L}$  eine Interpolationsmenge für den Raum  $\mathcal{S}_3^1(\diamond)$  ist. Bei den höhergradigen Interpolanten ist es hier nicht möglich, von den Bézier-Bernstein-Koeffizienten direkt auf die Interpolationswerte zu schließen.

Wir zeigen daher, dass das homogene Interpolationsproblem nur trivial lösbar ist, indem wir von den Polynomstücken sukzessive Geraden abspalten.

Zunächst benötigen wir die Dimension des Interpolationsraums, die wir über Bézier-Bernstein-Methoden ermitteln:

**Lemma 2.16**

Sei  $q \geq 4$ . Die Menge  $\mathcal{L}_q := \mathcal{L}_q^{[E]} \cup \mathcal{L}_q^{[I]}$  der gewählten Interpolationspunkte ist eine minimal bestimmende Menge für den Raum  $\mathcal{S}_q^1(\diamond)$ . Insbesondere ist  $\dim \mathcal{S}_q^1(\diamond) = |\mathcal{L}_q|$ .

**Beweis:** Wir untersuchen zunächst, welche Bézier-Bernstein-Koeffizienten eindeutig bestimmt sind, falls die zu  $\mathcal{L}_q^{[E]}$  gehörenden Koeffizienten vorgegeben sind. Hierfür werden alle  $C^1$ -Bedingungen über die Eckpunkte von  $\diamond$  berücksichtigt. Im Anschluss

zeigen wir, dass alle restlichen Koeffizienten in eindeutiger Weise bestimmt sind, falls die verbleibenden, in Algorithmus 2.15 gewählten Bézier-Bernstein-Koeffizienten vorgegeben sind. Dabei werden die noch nicht berücksichtigten  $C^1$ -Bedingungen verwendet.

Seien also die zu  $\mathcal{L}_q^{[E]}$  gehörenden Bézier-Bernstein-Koeffizienten vorgegeben.

**Behauptung 1:** Dann sind durch die  $C^1$ -Bedingungen über den Eckpunkten von  $\diamond$  alle Bézier-Bernstein-Koeffizienten auf den Kanten von  $\diamond$  und den  $C^1$ -Ringen in  $\diamond$  um die Knoten aus  $\diamond$  eindeutig bestimmt.

Wir zeigen diese Behauptung induktiv für alle Teilquadrangulierungen der durchlaufenen Vierecke. Dabei seien mit  $Q_1, \dots, Q_n \in \diamond$  die in Algorithmus 2.13 nacheinander durchlaufenen Vierecke bezeichnet. Beim Durchlauf von  $Q_1$  werden alle Rand-Bézier-Bernstein-Koeffizienten von  $Q_1$  gewählt. Insbesondere sind die Bézier-Bernstein-Koeffizienten der vier Eckpunkte von  $Q_1$  sowie jeweils zwei Nachbarpunkte eindeutig bestimmt, und die  $C^1$ -Bedingungen über diese vier Eckpunkte ziehen die eindeutige Bestimmtheit der  $C^1$ -Ringe in  $\diamond$  um diese Eckpunkte nach sich, d. h. die Behauptung gilt für  $i = 1$ . Gelte die Behauptung nun für die Teilquadrangulierung  $\diamond_{i-1}$ ,  $i - 1 \in \{1, \dots, n - 1\}$ . Vor dem Durchlauf des Vierecks  $Q_i$  sind genau die bereits gewählten Eckpunkte von  $Q_i$  Eckpunkte von  $\diamond_{i-1}$ , d. h. nach Induktionsannahme sind in  $Q_i$  genau die Bézier-Bernstein-Koeffizienten zu diesen Eckpunkten und deren Nachbarn bereits bestimmt. Gemäß Algorithmus 2.13 werden alle verbleibenden Rand-Bézier-Bernstein-Koeffizienten von  $Q_i$  gewählt. Insbesondere sind daher die Bézier-Bernstein-Koeffizienten zu jedem bislang noch nicht gewählten Eckpunkt von  $Q_i$  und zu jeweils zwei Nachbarpunkten bestimmt. Durch die  $C^1$ -Bedingungen über diese Eckpunkte sind daher die  $C^1$ -Ringe in  $\diamond$  um diese Eckpunkte eindeutig bestimmt. Also folgt die Behauptung für  $\diamond_i$ .

Ebenso sind auf den verbleibenden Kanten  $[a, b]$  von  $\diamond$ , die keinem Viereck aus  $\diamond$  angehören, die Bézier-Bernstein-Koeffizienten mit Abstand  $\leq 1$  zu  $a$  oder  $b$  bereits bestimmt. Die Punkte mit Abstand  $> 1$  werden gemäß Algorithmus 2.13 gewählt. Es folgt Behauptung 1.  $\square$

Seien nun alle zu den Interpolationspunkten aus  $\mathcal{L}_q$  gehörenden Bézier-Bernstein-Koeffizienten vorgegeben. Zunächst zeigen wir

**Behauptung 2:** Mit den verbleibenden  $C^1$ -Bedingungen über die durchgezogenen Diagonalen der zweigeteilten Vierecke und über die gemeinsamen Kanten von Paaren zweigeteilter Vierecke sind alle Bézier-Bernstein-Koeffizienten auf den zweigeteilten Vierecken in eindeutiger Weise bestimmt.

Wir betrachten zunächst ein einzelnes zweigeteiltes Viereck mit der durchgezogenen Diagonalen  $e$ . Es verbleiben die inneren Bézier-Bernstein-Koeffizienten der beiden Mikrodreiecke. In genau einem der Mikrodreiecke sind wegen Algorithmus 2.15, Nr. 2, alle inneren Bézier-Bernstein-Koeffizienten vorgegeben. Durch die verbleiben-

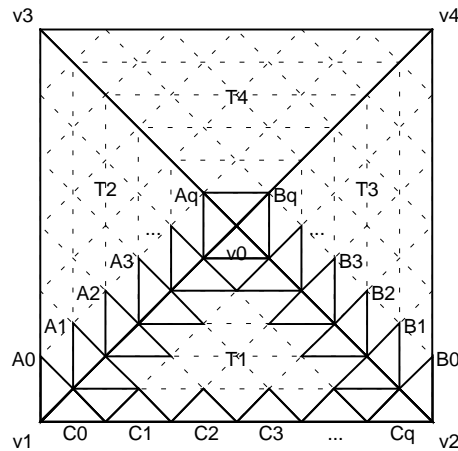


Abbildung 2.15: Notationen zu den Beweisen von Lemma 2.16, Lemma 2.17 und Theorem 2.18 ( $q=6$ )

den  $C^1$ -Bedingungen über  $e$  sind deshalb alle inneren Koeffizienten des zweiten Mikrodreiecks mit Abstand eins zu  $e$  in eindeutiger Weise bestimmt. Die verbleibenden Koeffizienten sind wegen Algorithmus 2.15, Nr. 2, vorgegeben.

Dieselbe Argumentation gilt für ein Viereck  $Q_1$  eines Paares  $\{Q_1, Q_m\}$  zweigeteilter Vierecke. Im Viereck  $Q_m$  sei  $T_{m,1}$  dasjenige Mikrodreieck mit  $T_{m,1} \cap Q_1 = a \neq \emptyset$ . Wegen den verbleibenden  $C^1$ -Bedingungen über die gemeinsame Kante  $a$  sind die inneren Bézier-Bernstein-Koeffizienten in  $T_{m,1}$  mit Abstand eins zu  $a$  eindeutig bestimmt, und die verbleibenden inneren Koeffizienten sind vorgegeben. Für das zweite Mikrodreieck  $T_{m,2}$  von  $Q_m$  gilt dieselbe Argumentation wie für das zweite Mikrodreieck eines einzelnen zweigeteilten Vierecks. Es folgt Behauptung 2.  $\square$

Es verbleiben die viergeteilten Vierecke. Wir betrachten den Durchlauf eines Mikrodreiecks  $T_1 = [v_0, v_1, v_2]$  eines viergeteilten Vierecks  $Q$  und verwenden die in Abbildung 2.15 dargestellte Notation. Ferner seien für  $t = 1, \dots, 4$  und  $i + j + k = q$ ,  $i, j, k \in \mathbb{N}_0$ , mit  $P_{ijk}^{[t]}$  der Bézier-Bernstein-Punkt und mit  $c_{ijk}^{[t]}$  der zugehörige Koeffizient im Dreieck  $T_t$  bezeichnet und weiterhin seien  $P_{ijk} = P_{ijk}^{[1]}$  sowie  $c_{ijk} = c_{ijk}^{[1]}$ .

Wir zeigen zunächst

**Behauptung 3:** Mit den verbleibenden  $C^1$ -Bedingungen über die durchlaufenen Nachbarn angehörenden Kanten und Eckpunkte von  $T_1$ , ausgenommen der Bedingungen über die Eckpunkte  $v_1$  und  $v_2$ , lassen sich alle verbleibenden Bézier-Bernstein-Koeffizienten auf  $T_1$  in eindeutiger Weise bestimmen, wobei die folgende Ausnahme gilt:

Gehören für  $j = 1, 2$  die Kanten  $[v_1, v_2]$  sowie  $[v_j, v_{j+2}]$  durchlaufenen Nachbarn an, die Kante  $[v_0, v_j]$  jedoch nicht, so lässt sich der zu  $\frac{q-2}{q}v_j + \frac{2}{q}v_0$  gehörende Koeffizient als einziger Koeffizient auf  $T_1$  nicht bestimmen. Er wird erst mit dem Durchlauf des Nachbardreiecks mit gemeinsamer Kante  $[v_0, v_j]$  bestimmt.

Es sei bemerkt, dass die  $C^1$ -Bedingungen über  $v_1$  und  $v_2$  schon verwendet wurden. Außerdem sind die Koeffizienten auf  $[v_1, v_2]$  und des Weiteren die Koeffizienten  $c_{1,q-1,0}$  sowie  $c_{1,0,q-1}$  bereits bestimmt. Für eine Darstellung der Situation, siehe Abbildung 2.15

Gehört nun  $v_0$  keinem durchlaufenen Nachbarn an, so handelt es sich um das erste durchlaufene Dreieck in  $Q$ , und es ist auch  $[v_0, v_j]$  nicht bestimmt für alle  $j = 1, \dots, 4$ . Es werden dann in  $T_1$  folgende Punkte gewählt:

$$P_{q00}, \quad P_{q-1,1,0}, \quad P_{q-1,0,1}, \quad P_{q-2,1,1}$$

Durch die  $C^1$ -Bedingung über  $v_0$  sind in eindeutiger Weise die Koeffizienten  $c_{q-1,1,0}^{[4]}$  und  $c_{q-1,0,1}^{[4]}$  bestimmt, und wir betrachten ferner die  $C^1$ -Bedingungen  $A_{q-1}, B_{q-1}$  sowie die entsprechenden Bedingungen auf den Kanten  $[v_0, v_3]$  und  $[v_0, v_4]$ . Mit diesen Bedingungen sind die Koeffizienten  $c_{q-2,1,1}^{[t]}$ ,  $t = 2, 3, 4$ , in eindeutiger Weise bestimmt. Da im Eckpunkt  $v_0$  jede Kante degeneriert ist, liegt keine Überbestimmung vor, siehe Lemma 3.3,  $l = 4$ , in [68]. Insgesamt sind also alle neun Koeffizienten

$$\mathcal{C}_1 := \left\{ c_{q00}^{[t]}, c_{q-1,1,0}^{[t]}, c_{q-1,0,1}^{[t]}, c_{q-2,1,1}^{[t]} \quad : \quad t = 1, \dots, 4 \right\}$$

in eindeutiger Weise bestimmt. Insbesondere sind die in  $T_1$  liegenden Koeffizienten aus  $\mathcal{C}_1$  bestimmt.

Gehört  $v_0$  dagegen einem durchlaufenen Nachbarn an, so handelt es sich nicht um das erste durchlaufene Dreieck in  $Q$ , d. h. die in  $T_1$  liegenden Koeffizienten aus  $\mathcal{C}_1$  sind bereits bestimmt.

Wir untersuchen als Nächstes die Koeffizienten

$$\mathcal{C}_2 = \{ c_{1,i,q-i-1} \quad : \quad i = 1, \dots, q-2 \}.$$

Gehört  $[v_1, v_2]$  einem durchlaufenen Nachbarn an, dann sind diese Koeffizienten durch die Bedingungen  $C_1, \dots, C_{q-1}$  bestimmt, andernfalls werden die zugehörigen Punkte gewählt.

Würde eine der Kanten  $[v_1, v_3]$  und  $[v_2, v_4]$  einem durchlaufenen Nachbarn angehören, nicht aber  $[v_1, v_2]$ , so gäbe es durch die Bedingung  $A_1$  bzw.  $B_1$  eine Überbestimmung des Koeffizienten  $c_{1,q-2,1}$  bzw.  $c_{1,1,q-2}$ . Da gemäß Algorithmus 2.15 jedoch zunächst diejenigen Mikrodreiecke  $T_1$  mit einem bereits durchlaufenen Nachbarn angehörender Kante  $[v_1, v_2]$  durchlaufen werden, ist dieser Fall ausgeschlossen.



Für  $j = 1, 2$  untersuchen wir den Bézier-Bernstein-Punkt  $P_j := \frac{q-2}{q}v_j + \frac{2}{q}v_0$  und den zugehörigen Koeffizienten  $c_j$ . Zunächst nehmen wir an, dass die Kanten  $[v_1, v_2]$  und  $[v_j, v_{j+2}]$  nicht beide zugleich durchlaufenen Nachbarn angehören. Dann ist  $c_j$  eindeutig bestimmt, falls  $[v_0, v_j]$  einem bereits durchlaufenen Nachbarn angehört. Gehört  $[v_0, v_j]$  keinem durchlaufenen Nachbarn an, so wird  $P_j$  gewählt. Seien im Folgenden also beide Kanten  $[v_1, v_2]$  sowie  $[v_j, v_{j+2}]$  Kanten von bereits durchlaufenen Nachbarn. Gehört außerdem auch  $[v_0, v_j]$  einem durchlaufenen Nachbarn an, so betrachte man die Bedingung  $A_1$  für  $j = 1$  und  $B_1$  für  $j = 2$ : Durch diese Bedingung ist der Koeffizient  $c_j$  in eindeutiger Weise bestimmt, da die Nachbarkoeffizienten von  $c_j$  im Inneren der Mikrodreiecke bestimmt sind. Ist jedoch  $[v_0, v_j]$  nicht bestimmt, so ist  $c_j$  noch nicht bestimmt, siehe Behauptung). Wir bezeichnen  $\mathcal{C}_j = \{c_{j-2}\}$  für  $j = 3, 4$ .

Verbleiben für  $j = 1, 2$  Koeffizienten auf der Kante  $[v_0, v_j]$ , so werden die zugehörigen Punkte genau dann gewählt, falls die Kante keinem durchlaufenen Nachbarn angehört.

Für  $q \geq 5$  fahren wir fort mit den Koeffizienten

$$\mathcal{C}_5 := \{c_{i, q-i-1, 1} : i = 2, \dots, q-3\}.$$

Gehört  $[v_0, v_1]$  einem durchlaufenen Nachbarn an, so sind diese Koeffizienten durch die Bedingungen  $A_2, \dots, A_{q-2}$  in eindeutiger Weise bestimmt. Andernfalls werden die zugehörigen Bézier-Bernstein-Punkte gewählt.

Analog verhält es sich für  $q \geq 5$  mit den Koeffizienten

$$\mathcal{C}_6 := \{c_{i, 1, q-i-1} : i = 2, \dots, q-3\}:$$

Für  $q \geq 6$  verbleiben die Koeffizienten

$$\mathcal{C}_7 := \{c_{i, j, k} : i + j + k = q, i, j, k \in \{2, \dots, q-2\}\}.$$

Die zugehörigen Punkte werden gewählt. Es folgt Behauptung 3.  $\square$

Wie gesehen sind die Bézier-Bernstein-Koeffizienten im Inneren der zweigeteilten Vierecke, auf den Kanten von  $\diamond$  und den  $C^1$ -Ringen in  $\diamond$  um Eckpunkte aus  $\diamond$  eindeutig bestimmt. Mit Behauptung 3 zeigen wir per Induktion, dass die verbleibenden  $C^1$ -Bedingungen alle verbleibenden Koeffizienten von  $s$  in eindeutiger Weise bestimmen und keine Überbestimmung vorliegt.

Zunächst zeigen wir

**Behauptung 4:** Nach dem Durchlauf aller Mikrodreiecke eines viergeteilten Vierecks  $Q$  sind durch die  $C^1$ -Bedingungen über die mit durchlaufenen Nachbarn gemeinsamen Kanten und über die durchgezogenen Diagonalen alle Koeffizienten auf  $Q$  in eindeutiger Weise bestimmt.

Mit dem Durchlauf eines Mikrodreiecks  $T_1$  von  $Q$  werden gemäß Behauptung 3 alle verbleibenden Koeffizienten auf  $T_1$  in eindeutiger Weise bestimmt. Ist für  $j = 1, 2$  der zu  $\frac{q-2}{q}v_j + \frac{2}{q}v_0$  gehörende Koeffizient noch nicht bestimmt, so geschieht diese Bestimmung mit dem Durchlauf des entsprechenden Nachbardreiecks  $\tilde{T}_1$ , welches ebenfalls Mikrodreieck von  $Q$  ist. Insgesamt sind also alle Koeffizienten auf  $Q$  eindeutig bestimmt. Somit gilt Behauptung 4.  $\square$

Mit dieser Behauptung folgt induktiv, dass alle Bézier-Bernstein-Koeffizienten von  $s$  unter Verwendung aller  $C^1$ -Bedingungen eindeutig bestimmt sind und dass keine Überbestimmung vorliegt: Mit dem Durchlauf des ersten viergeteilten Vierecks  $Q$  sind alle Koeffizienten auf  $Q$  eindeutig bestimmt. Selbiges gilt für jedes weiter durchlaufene viergeteilte Viereck. Nach dem Durchlauf des letzten viergeteilten Vierecks sind unter Berücksichtigung aller  $C^1$ -Bedingungen sämtliche Bézier-Bernstein-Koeffizienten von  $s$  in eindeutiger Weise bestimmt.

Dies beweist Lemma 2.16.  $\square$

Wir zeigen nun, dass  $\mathcal{L}_q$ ,  $q \geq 4$ , eine Lagrange-Interpolationsmenge für den Raum  $\mathcal{S}_q^1(\diamond)$  ist. Hierfür verwenden wir folgendes

**Lemma 2.17**

Sei  $q \geq 4$ . Weiterhin seien auf einem Mikrodreieck  $T_1$  eines viergeteilten Vierecks  $Q$  folgende Bedingungen erfüllt:

(i)  $s(z) = 0$  für alle  $z \in \mathcal{L}_q \cap T_1$

(ii)  $s|_{\delta Q} \equiv 0$

(iii)  $s|_T \equiv 0$  für alle durchlaufenen angrenzenden Dreiecke  $T$ .

Dann folgt  $s|_{T_1} \equiv 0$ .

**Beweis:** Sei  $T_1$  ein Mikrodreieck von  $Q$ , für welches die Bedingungen (i) bis (iii) erfüllt sind. Wir verwenden im Folgenden die Notation aus dem Beweis zu Lemma 2.16 und bezeichnen außerdem mit  $e_{ij}$  den Einheitsvektor in Richtung der Kante  $[v_i, v_j]$  für alle  $i, j \in \{0, \dots, 4\}$ ,  $i \neq j$ . Zuerst zeigen wir, dass der Interpolant auf den Kanten von  $T_1$  gleich Null ist. Dabei ist für  $[v_1, v_2]$  wegen (ii) nichts zu zeigen.

Gehöre  $v_0$  zunächst keinem durchlaufenen Dreieck an. Dann gehören auch die Kanten  $[v_0, v_j]$ ,  $j = 1, 2$ , keinem durchlaufenen Dreieck an, und wir zeigen zunächst  $s|_{[v_0, v_1]} \equiv 0$ . In diesem Fall werden die Bézier-Bernstein-Punkte  $\{P_{i, q-i, 0} : i = 3, \dots, q\}$  als Interpolationspunkte gewählt. Wegen (ii) ist außerdem  $s(v_1) = 0$

und  $\text{grad } s(v_1) \equiv 0$ . Gehören beide Kanten  $[v_1, v_2]$  und  $[v_1, v_3]$  durchlaufenen Nachbarn an, so folgt mit (iii)  $s_{e_{10}e_{12}}(v_1) = 0$  und  $s_{e_{10}e_{13}}(v_1) = 0$ . Insbesondere ist

$$s_{e_{10}^2}(v_1) = 0. \quad (2.2)$$

Dies lässt sich auch mit Hilfe von Bézier-Bernstein-Techniken zeigen, siehe Abbildung 2.15 auf Seite 54: Mit (ii) folgt, dass alle Bézier-Bernstein-Koeffizienten auf  $\delta Q$  gleich Null sind. Mit (iii) folgt, dass die auf den bezüglich  $[v_1, v_2]$  und  $[v_1, v_3]$  benachbarten Dreiecken liegenden Koeffizienten gleich Null sind. Somit sind alle Bézier-Bernstein-Koeffizienten der  $C^2$ -Scheibe um  $v_1$  in  $T_1$  und  $T_2$  mit Ausnahme von  $c_{\frac{2}{q}v_0 + \frac{q-2}{q}v_1}$  gleich Null sind. Wegen der Bedingung  $A_1$  ist dann auch dieser Koeffizient gleich Null und mit Satz 1.8 folgt insbesondere (2.2).

Also folgt mit insgesamt  $q+1$  auf der Kante  $[v_0, v_1]$  liegenden Hermite-Interpolationsbedingungen  $s|_{[v_0, v_1]} \equiv 0$ . Gehören andererseits nicht beide Kanten  $[v_1, v_2]$  und  $[v_1, v_3]$  durchlaufenen Nachbarn an, so wird außerdem der Punkt  $P_{2, q-2, 0}$  gewählt, und wir erhalten ebenfalls mit  $q+1$  Bedingungen auf der Kante  $[v_1, v_2]$  ein Hermite-Schema. Auch in diesem Fall ist also  $s|_{[v_0, v_1]} \equiv 0$ . Da die Situation für die Kante  $[v_0, v_2]$  analog ist, folgt ebenso  $s|_{[v_0, v_2]} \equiv 0$ .

Gehöre nun  $v_0$  einem durchlaufenen Nachbarn an. Dann ist  $s$  an den Stellen  $v_j$ ,  $j = 0, 1, 2$ , wegen (ii) und (iii) in Funktionswert und Gradient gleich Null. Gehört die Kante  $[v_0, v_1]$  einem durchlaufenen Nachbarn an, so ist  $s|_{[v_0, v_1]} \equiv 0$ . Andernfalls werden für  $q \geq 5$  die  $q-4$  Bézier-Bernstein-Punkte  $\{P_{i, q-i, 0} : i = 3, \dots, q-2\}$  gewählt. Wie oben beschrieben, wird außerdem entweder der Punkt  $P_{2, q-2, 0}$  gewählt, oder es ist  $s_{e_{10}^2}(v_1) = 0$ . In beiden Fällen liegen also insgesamt  $q+1$  Hermite-Bedingungen auf der Kante  $[v_0, v_1]$  vor, und es folgt  $s|_{[v_0, v_1]} \equiv 0$ . Die gleichen Argumente gelten für die Kante  $[v_0, v_2]$ . Also folgt in allen Fällen  $s|_{\delta T_1} \equiv 0$ .

Wir bezeichnen nun die Kanten von  $T_1$  mit  $e_1 = [v_1, v_2]$ ,  $e_2 = [v_1, v_0]$  und  $e_3 = [v_2, v_0]$ . Für  $i = 1, 2, 3$  sei

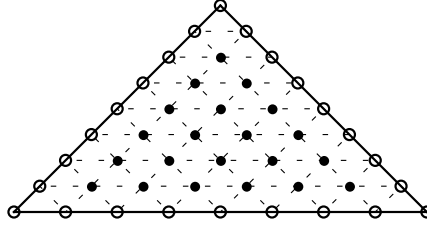
$$l_i : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad l_i \not\equiv 0, \quad \text{definiert durch } l_i(z) = a_i \cdot z + b_i, \quad a_i \in \mathbb{R}^2, \quad b_i \in \mathbb{R},$$

eine Ebene im Raum, welche  $e_i$  enthält. Wegen  $l_i \not\equiv 0$  folgt aus der Linearität von  $l_i$  insbesondere  $l_i(z) \neq 0$  für alle im Inneren des Dreiecks  $T_1$  liegenden Punkte  $z$ . Wir wählen für  $i = 1, 2, 3$  weiterhin

$$d_i = \begin{cases} 1 & \text{falls die Kante } e_i \text{ einem durchlaufenen Nachbarn angehört,} \\ 0 & \text{sonst.} \end{cases}$$

Mit (iii) und  $s|_{\delta T_1} \equiv 0$  folgt für alle  $i = 1, 2, 3$

$$D^w s(z) = 0, \quad z \in e_i, \quad 0 \leq w \leq d_i.$$


 Abbildung 2.16: Die inneren Bézier-Bernstein-Punkte von  $T_1$  ( $q = 8$ )

Aufgrund dieser Bedingung können wir jede Gerade  $e_i$ ,  $i = 1, 2, 3$ , zweimal vom Polynom  $s|_{T_1}$  abspalten, falls die Kante  $e_i$  einem durchlaufenen Nachbarn angehört, sonst können wir sie einmal abspalten. Wir wählen

$$d = q - 3 - \sum_{i=1}^3 d_i.$$

Ist  $d < 0$ , so folgt aus den Differenzierbarkeitsbedingungen über den Kanten direkt  $s|_{T_1} \equiv 0$ , d. h. es gilt die Behauptung. Für  $d \geq 0$  existiert ein Polynom  $p \in \mathcal{P}_d^2$ , sodass sich  $s|_{T_1}$  schreiben lässt als

$$s|_{T_1}(z) = l_1^{d_1+1}(z) l_2^{d_2+1}(z) l_3^{d_3+1}(z) \cdot p(z), \quad z \in T_1. \quad (2.3)$$

Wir betrachten die im Inneren des Dreiecks  $T_1$  gewählten Interpolationspunkte. Abbildung 2.17 zeigt alle möglichen Fälle. Gehöre  $v_0$  zunächst keinem durchlaufenen Nachbarn an, oder es gehöre mindestens eine der Kanten  $[v_0, v_i]$ ,  $i = 1, 2$ , einem durchlaufenen Nachbarn an (Fälle 2 bis 4 und 6 bis 10).

Die Anzahl aller Bézier-Bernstein-Punkte von  $T_1$  ist  $\sum_{i=1}^{q+1} i$ , siehe Abbildung 2.16. Somit beträgt die Anzahl aller im Inneren von  $T_1$  liegenden Bézier-Bernstein-Punkte  $(q-2) + (q-3) + \dots + 1$ . Gehört keine der Kanten von  $T_1$  einem durchlaufenen Nachbarn an, so werden alle inneren Punkte in  $T_1$  gewählt. Ferner ist in diesem Fall  $d = q - 3$ , und mit  $\dim \mathcal{P}_d^2 = \sum_{i=1}^{q-2} i$  folgt

$$|\mathcal{L}_q^{[I]} \cap T_1| = \dim \mathcal{P}_d^2,$$

Bei dieser Menge der im Inneren von  $T_1$  liegenden Interpolationspunkte handelt es sich um die Standardlage von Interpolationspunkten.

Gehört genau eine der Kanten von  $T_1$  einem durchlaufenen Nachbarn an, so reduziert sich die Anzahl  $|\mathcal{L}_q^{[I]} \cap T_1|$  dieser Interpolationspunkte um  $q - 2$ , d. h. es folgt auch hier

$$|\mathcal{L}_q^{[I]} \cap T_1| = (q-3) + (q-4) + \dots + 1 = \dim \mathcal{P}_d^2.$$

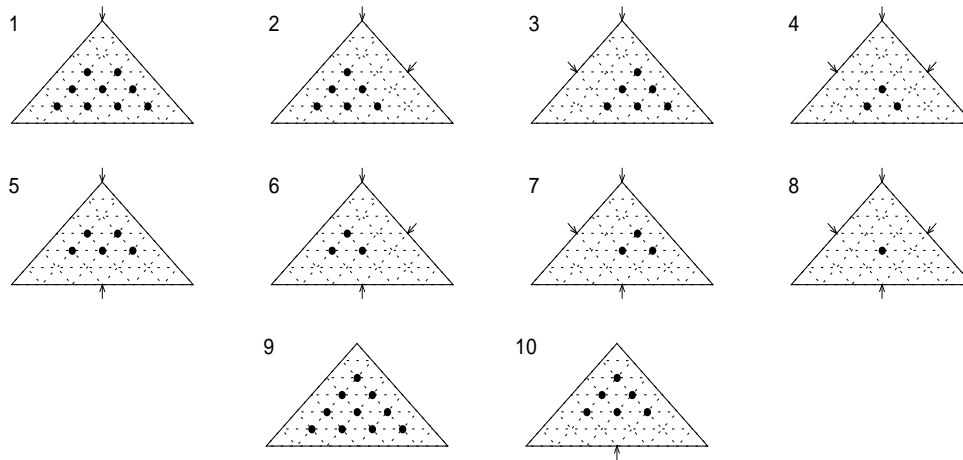


Abbildung 2.17: Die im Inneren von  $T_1$  gewählten Punkte in Abhängigkeit von den gemeinsamen Kanten und Eckpunkten mit bereits durchlaufenen Dreiecken ( $q = 6$ )

Gehört noch genau eine zweite der Kanten von  $T_1$  einem durchlaufenen Nachbarn an, so werden weitere  $q - 3$  Punkte weniger gewählt, und wieder folgt

$$|\mathcal{L}_q^{[l]} \cap T_1| = (q - 4) + (q - 5) + \dots + 1 = \dim \mathcal{P}_d^2.$$

Gehört schließlich auch die dritte Kante einem durchlaufenen Nachbarn an, so gilt analog:

$$|\mathcal{L}_q^{[l]} \cap T_1| = (q - 5) + (q - 6) + \dots + 1 = \dim \mathcal{P}_d^2$$

Zu allen Gleichungen sei bemerkt, dass wegen  $d \geq 0$  jede Summe strikt größer Null ist. Bei den gewählten Punkten  $\mathcal{L}_q^{[l]} \cap T_1$  handelt es sich in allen Fällen um die Standardlage (1.6) von Interpolationspunkten, und es folgt  $p \equiv 0$  und mit (2.3) schließlich

$$s|_{T_1} \equiv 0.$$

Es verbleibt die Situation, dass  $v_0$  einem durchlaufenen Nachbarn angehört, die Kanten  $[v_0, v_1]$  und  $[v_0, v_2]$  jedoch nicht. Für  $q = 6$  zeigt diese Situation Abbildung 2.17, Fälle 1 und 5. Wir spalten nun sukzessive Parallelen zu  $e_1$  von  $T_1$  ab. Für alle  $j = 0, \dots, q - 1$  seien

$$e_1^{[j]} = [P_{j, q-j, 0}, P_{j, 0, q-j}]$$

und

$$l_1^{[j]} : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad l_1^{[j]} \neq 0, \quad \text{eine die Kante } e_1^{[j]} \text{ enthaltende Ebene im Raum.}$$

Gehöre  $e_1$  zunächst keinem durchlaufenen Nachbarn an (Abbildung 2.17, Fall 1). Dann ist  $p \in \mathcal{P}_{q-3}^2$ , und auf der Kante  $e_1^{[1]}$  liegen  $q - 2$  Interpolationspunkte. Wegen

Gleichung (2.3) ist  $p$  Null auf diesen Punkten. Mit univariater Lagrange-Interpolation folgt  $p|_{e_1^{[1]}} \equiv 0$ , und es existiert ein Polynom  $p_1 \in \mathcal{P}_{q-4}^2$ , sodass sich  $p$  schreiben lässt als

$$p \equiv l_1^{[1]} p_1. \quad (2.4)$$

Gehört  $e_1$  dagegen einem durchlaufenen Nachbarn an (Abbildung 2.17, Fall 5), so setzen wir  $p_1 = p$ . In beiden Fällen ist also  $p_1 \in \mathcal{P}_{q-4}^2$ , und wir fahren wie folgt fort:

Für  $q \geq 5$  spalten wir von  $p_1$  nacheinander die Geraden

$$e_1^{[2]}, \dots, e_1^{[q-3]}$$

ab und erhalten somit Polynome

$$p_2 \in \mathcal{P}_{q-5}^2, \dots, p_{q-3} \in \mathcal{P}_{q-q}^2.$$

Jedes Abspalten erfolgt analog wie das Abspalten der Gerade  $e_1^{[1]}$ , siehe Gleichung (2.4). Man sieht, dass für alle  $j = 2, \dots, q-3$  die Anzahl der Interpolationspunkte auf der abgespaltenen Geraden  $e_1^{[j]}$  mit der Dimension des zu  $p_{j-1}|_{e_1^{[j]}}$  gehörenden Polynomraums übereinstimmt: Im Inneren von  $T_1$  befinden sich  $q-j-1$  Interpolationspunkte auf der Geraden  $e_1^{[j]}$ , und es ist  $p_{j-1}|_{e_1^{[j]}} \in \mathcal{P}_{q-j-2}$  mit  $\dim \mathcal{P}_{q-j-2} = q-j-1$ . Also folgt

$$p_{j-1} \equiv l_1^{[j]} p_j, \quad j = 2, \dots, q-3. \quad (2.5)$$

Wir erhalten ein Restpolynom  $p_{q-3} \in \mathcal{P}_0^2$ , und es bleibt  $p_{q-3} \equiv 0$  zu zeigen, für die Notationen siehe Abbildung 2.15 auf Seite 54. Da für  $T_1$  Fall 1 oder 5 gilt, ist  $T_4$  bereits durchlaufen,  $T_2$  und  $T_3$  jedoch nicht. Beim Durchlauf von  $T_4$  trat insbesondere weder Fall 1 noch Fall 5 auf, denn  $T_1$  war zu diesem Zeitpunkt noch nicht durchlaufen. Mit obigen Argumenten ist daher  $s|_{T_4} \equiv 0$ . Also gilt  $c_{ijk}^{[T_4]} = 0$ ,  $i+j+k = q$ . Durch die  $C^1$ -Bedingungen über die Kante  $[v_0, v_3]$  ergibt sich insbesondere  $c_{q-2,1,1}^{[T_2]} = 0$ . Für  $i = 1, 2$  folgt aus  $[v_0, v_i] \equiv 0$  auch  $c_{\frac{j}{q}v_0 + \frac{q-j}{q}v_i} = 0$ ,  $j = 0, \dots, q$ . Mit der  $C^1$ -Bedingung  $A_{q-1}$  folgt, dass alle Bézier-Bernstein-Koeffizienten von  $s|_{T_1}$  in der  $C^2$ -Scheibe um  $v_0$  gleich Null sind. Wegen Satz 1.8 ist daher  $D^2(s|_{T_2}) \equiv 0$ , und es folgt  $p_{q-3} \equiv 0$  für das Restpolynom. Mit den Gleichungen (2.2) bis (2.4) folgt schließlich

$$s|_{T_1} \equiv 0.$$

Dies beweist Lemma 2.17. □

Mit diesem Lemma zeigen wir nun

### Theorem 2.18

Sei  $q \geq 4$ . Dann ist  $\mathcal{L}_q$  eine Lagrange-Interpolationsmenge für den Raum  $\mathcal{S}_q^1(\diamond)$ .

**Beweis:** Wegen Lemma 2.16 genügt es, die Eindeutigkeit des homogenen Problems zu zeigen. Es sei also  $s \in \mathcal{S}_q^1(\diamond)$  mit  $s|_{\mathcal{L}_q} \equiv 0$ , und wir zeigen  $s \equiv 0$ .

**Behauptung 1:** Für jeden Punkt  $z$  auf den Kanten von  $\diamond$  und auf den durchgezogenen Diagonalen der zweigeteilten Vierecke ist  $s(z) = 0$ .

In Algorithmus 2.13 werden für  $l = 4$  auf jeder Kante eines durchlaufenen Vierecks  $Q$  alle Bézier-Bernstein-Punkte gewählt, und es folgt mit univariater Lagrange-Interpolation auf jeweils  $q + 1$  Stellen, dass  $s$  auf diesen Kanten Null ist. Da kein Viereck degeneriert ist, folgt für jeden Eckpunkt  $v$  von  $Q$ , dass der Gradient  $\text{grad } s(v) \equiv 0$  ist.

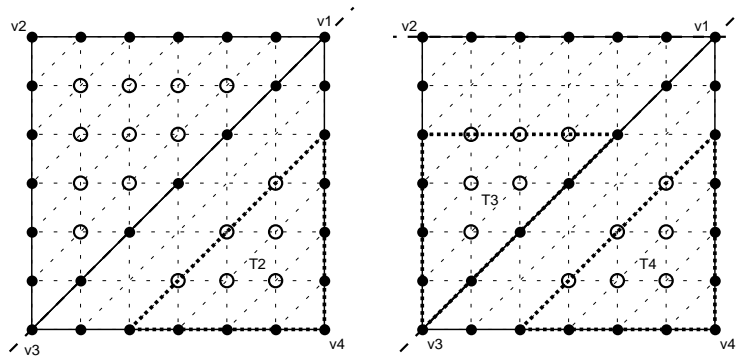
Wir betrachten nun den Fall  $l = 3$ , also zunächst das erste Viereck  $Q$  der Iteration mit genau drei nicht gewählten Eckpunkten. Für genau zwei Kanten von  $Q$  gilt obige Argumentation. Sei  $e$  eine der verbleibenden beiden Kanten. Laut Algorithmus 2.13 ist der Interpolant auf  $e$  an  $q$  Stellen gleich Null. An einer dieser Stellen, nämlich am bereits gewählten Eckpunkt  $v$ , ist  $s$  auch im Gradienten, also insbesondere in der Richtungsableitung in Richtung  $e$  gleich Null. Demnach liegt für  $s|_e$  ein eindeutiges Hermite-Interpolationsschema vor, und es folgt  $s|_e \equiv 0$ . Da wiederum für jeden im Iterationsschritt gewählten Eckpunkt  $v$  der Gradient  $\text{grad } s(v) \equiv 0$  ist, gilt dies analog für alle verbleibenden Vierecke mit genau drei nicht gewählten Eckpunkten.

Wir betrachten die unbestimmten Kanten  $e_1$ ,  $e_2$  und  $e_3$  des ersten durchlaufenen Vierecks  $Q$  mit genau zwei nicht gewählten Eckpunkten: Auf einer dieser Kanten liegen  $q + 1$  Lagrange-Bedingungen vor, auf den beiden übrigen ist wie oben ein eindeutiges Hermite-Schema gegeben. Also folgt  $s|_{\{e_1, e_2, e_3\}} \equiv 0$ , und wieder ist  $\text{grad } s(v) \equiv 0$  für alle Eckpunkte von  $Q$ , sodass für die verbleibenden Vierecke mit  $l = 2$  analog argumentiert werden kann.

Für ein Viereck  $Q$  mit genau einem nicht gewählten Eckpunkt sind genau zwei Kanten  $e_1, e_2$  zu bestimmen, und mit zwei Hermite-Interpolationsschemata folgt ebenfalls  $s|_{\{e_1, e_2\}} \equiv 0$ . Auch hier ist mit obigem Argument  $\text{grad } s(v) \equiv 0$  für alle Eckpunkte von  $Q$ .

Auf jeder verbleibenden Kante  $[a, b]$  von  $\diamond$  liegen aufgrund der Punkte in  $\mathcal{L}_q^{[E]}$  genau  $q - 1$  Lagrange-Bedingungen vor, und es ist  $\text{grad } s(a) \equiv \text{grad } s(b) \equiv 0$ . Mit Hermite-Interpolation folgt  $s|_{[a, b]} \equiv 0$ . Ebenso ist wegen Algorithmus 2.15, Nr. 1, der Interpolant auf der Diagonalen jedes zweigeteilten Vierecks in  $\diamond$  gleich Null. Es folgt Behauptung 1.  $\square$

Wir betrachten nun das Innere der Vierecke aus  $\diamond$  und beginnen mit einem einzelnen zweigeteilten Viereck  $Q = [v_1, v_2, v_3, v_4]$ . Abbildung 2.18 (links) stellt die Situation für  $q = 6$  dar. O. B. d. A. seien  $[v_1, v_3]$  die durchgezogene Diagonale des Vierecks und  $T_1 = [v_1, v_2, v_3]$  dasjenige Mikrodreieck, für welches alle inneren Bézier-Bernstein-Punkte in  $\mathcal{L}_q$  enthalten sind. Wegen  $s|_{\delta T_1} \equiv 0$  ist  $s$  insbesondere auf den Rand-Bézier-Bernstein-Punkten, also insgesamt auf allen Bézier-Bernstein-Punkten gleich


 Abbildung 2.18: Abspaltung von Geraden bei zweigeteilten Vierecken ( $q = 6$ )

Null. Da dies eine Standardlage von Interpolationspunkten für bivariate Polynome ist, folgt  $s|_{T_1} \equiv 0$ .

Wir fahren fort mit  $T_2 = [v_1, v_3, v_4]$ : Wegen der Differenzierbarkeitsbedingung über  $[v_1, v_3]$  ist  $D^w s(z) \equiv 0$  für alle  $w = 0, 1$  und  $z \in [v_1, v_3]$ . Deshalb lässt sich das Polynom  $s|_{T_2}$  schreiben als

$$s|_{T_2}(z) = l_1^2(z) \cdot p(z), \quad z \in T_2, \quad (2.6)$$

wobei  $p$  ein Polynom vom Grad  $q - 2$  ist und  $l_1 : \mathbb{R}^2 \rightarrow \mathbb{R}$ ,  $l_1 \not\equiv 0$ , definiert durch  $l_1(z) = a \cdot z + b$  mit  $a \in \mathbb{R}^2$ ,  $b \in \mathbb{R}$  eine Ebene, die  $[v_1, v_3]$  enthält. Mit  $l_1 \not\equiv 0$  folgt  $l_1|_{[v_1, v_3]} \equiv 0$  und  $l_1(z) \neq 0$  für alle  $z \in T_2 \setminus [v_1, v_3]$ .

Nun ist  $s|_{T_2}$  auf den Rand-Bézier-Bernstein-Punkten von  $T_2$  und auf den im Inneren von  $T_2$  gewählten Punkten gleich Null. Also ist  $s|_{T_2}$  insbesondere auf allen  $\frac{(q-1)q}{2}$  Bézier-Bernstein-Punkten

$$\mathcal{P} := \left\{ \frac{i}{q}v_1 + \frac{j}{q}v_3 + \frac{k}{q}v_4 : i + j + k = q, k \geq 2 \right\}$$

gleich Null. Da  $l_1(z) \neq 0$  ist für alle  $z \in \mathcal{P}$ , folgt mit Gleichung (2.6)  $p(z) = 0$  für alle  $z \in \mathcal{P}$ . Da  $\mathcal{P}$  eine Lagrange-Interpolationsmenge für den Raum  $\mathcal{P}_{q-2}$  ist, folgt  $p \equiv 0$  und schließlich  $s|_{T_2} \equiv 0$ .

Sei nun  $\{Q_1, Q_2\}$  ein benachbartes Paar zweigeteilter Vierecke. Laut Algorithmus 2.15 liegt für genau eines der Vierecke dieselbe Situation vor wie für ein einzelnes zweigeteiltes Viereck, o. B. d. A. sei dies  $Q_1$ . Wie gesehen folgt  $s|_{Q_1} \equiv 0$ . In  $Q_2 = [v_1, v_2, v_3, v_4]$  seien  $T_3 = [v_1, v_2, v_3]$  und  $T_4 = [v_1, v_3, v_4]$  die beiden Mikrodreiecke, und o. B. d. A. sei  $[v_1, v_2]$  die zu  $Q_1$  gemeinsame Kante. Abbildung 2.18 (rechts) veranschaulicht die Situation für den Fall  $q = 6$ .

Wie gesehen ist  $s|_e \equiv 0$  auf jeder Kante  $e$  jedes Mikrodreiecks  $T_3$  und  $T_4$ . Durch die  $C^1$ -Bedingung über  $[v_1, v_2]$  ist  $D^w s(z) \equiv 0$  für alle  $w = 0, 1$  und  $z \in [v_1, v_2]$ . Also



lässt sich das Polynom  $s|_{T_3}$  schreiben als

$$s|_{T_3}(z) = l_2^2(z) \cdot p_2(z), \quad z \in T_3, \quad (2.7)$$

wobei  $p_2$  ein Polynom vom Grad  $q - 2$  ist und  $l_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$ ,  $l_1 \neq 0$ , definiert durch  $l_2(z) = a \cdot z + b \neq 0$  mit  $a \in \mathbb{R}^2$ ,  $b \in \mathbb{R}$  eine Ebene, die  $[v_1, v_2]$  enthält. Wieder folgt  $l_2(z) \neq 0$  für alle  $z \in T_3 \setminus [v_1, v_2]$ .

Da  $s(z) = 0$  ist auf jedem Rand-Bézier-Bernstein-Punkt  $z$  von  $T_3$  und ferner jeder innere Bézier-Bernstein-Punkt auf  $T_3$  mit Abstand  $\geq 2$  zu  $[v_1, v_2]$  gewählt wird, folgt für die Menge

$$\tilde{\mathcal{P}} := \left\{ \frac{i}{q}v_1 + \frac{j}{q}v_2 + \frac{k}{q}v_3 : i + j + k = q, k \geq 2 \right\}$$

$s|_{\tilde{\mathcal{P}}} \equiv 0$ . Mit  $l_2(z) \neq 0$  für alle  $z \in \tilde{\mathcal{P}}$  folgt  $p_2|_{\tilde{\mathcal{P}}} \equiv 0$ . Da  $\tilde{\mathcal{P}}$  eine Lagrange-Interpolationsmenge für den Raum  $\mathcal{P}_{q-2}$  ist, folgt  $p_2 \equiv 0$ , also  $s|_{T_1} \equiv 0$ . Für das Dreieck  $T_4$  liegt derselbe Fall vor wie für das zweite Mikrodreieck eines einzelnen zweigeteilten Vierecks. Mit obigen Argumenten folgt  $s|_{T_4} \equiv 0$ , d. h. es ist  $s|_{Q_2} \equiv 0$ .

Somit folgt induktiv  $s|_Q \equiv 0$  auf jedem zweigeteilten Viereck  $Q$ . Weiterhin folgt mit Lemma 2.17 induktiv  $s|_{T_5} \equiv 0$  für jedes durchlaufene Mikrodreieck  $T_5$  der viergeteilten Vierecke, d. h. schließlich ist  $s \equiv 0$ . Dies beweist Theorem 2.18.  $\square$

Wir zeigen ähnlich wie in [68], dass die Methode lokal ist. Weiterhin zeigen wir, dass die konstruierten Spline-Interpolanten stabil bestimmbar sind. Dabei sei  $q \geq 4$  und es bezeichnen  $f_z$ ,  $z \in \mathcal{L}_q$ , die Interpolationswerte.

### Theorem 2.19

(i) Für jeden Punkt  $z \in \mathcal{L}_q$  liegt der Träger des zugehörigen Basissplines  $\mathcal{B}_z$  mit  $\mathcal{B}_z(\tilde{z}) = \delta_{z\tilde{z}}$  für alle  $\tilde{z} \in \mathcal{L}_q$

(a) in einem  $\text{star}^5(Q)$ ,  $Q \in \diamond$ , falls  $z$  im Inneren eines Vierecks von  $\diamond$  liegt.

(b) in einem  $\text{star}^9(Q)$ ,  $Q \in \diamond$ , falls  $z$  auf den Kanten von  $\diamond$  liegt.

Insbesondere ist die Interpolationsmethode lokal.

(ii) Es existiert eine lediglich vom kleinsten Winkel  $\alpha$  von  $\diamond$  abhängende Konstante  $C$  mit der folgenden Eigenschaft: Für jeden Bézier-Bernstein-Koeffizienten  $c_P$ ,  $P \in \mathcal{P}_q(\diamond)$ , existiert eine beschränkte Teilmenge  $\mathcal{L}_P \subseteq \mathcal{L}_q$  von Interpolationspunkten mit

$$|c_P| \leq C \cdot \max_{z \in \mathcal{L}_P} |f_z|, \quad (2.8)$$

d. h. die Interpolationsmethode ist stabil.

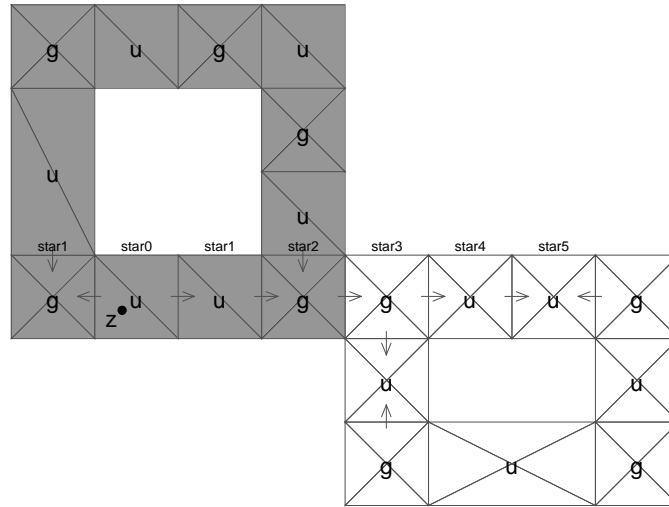


Abbildung 2.19: Maximale Trägergröße von  $\mathcal{B}_z$ , falls  $z$  im Innern eines Vierecks liegt; Gerade Vierecke sind mit „g“, ungerade mit „u“ gekennzeichnet

**Beweis:**

(i)

Sei  $z \in \mathcal{L}_q$ . Wir untersuchen die *maximale* Größe, welche die Träger der Basissplines  $\mathcal{B}_z$  besitzen können. Als Viereck  $Q$  des  $\text{star}^m(Q)$ ,  $m \in \mathbb{N}_0$ , wählen wir dasjenige den Punkt  $z$  enthaltende Viereck  $Q \in \diamond$ , welches im Kantenalgorithmus zuerst durchlaufen wurde. Falls kein solches Viereck existiert, wählen wir ein beliebiges, den Punkt  $z$  enthaltendes Viereck  $Q \in \diamond$ .

Fall 1:  $z$  liegt im Inneren eines Vierecks  $Q \in \diamond$

Laut Voraussetzung ist  $s|_{\mathcal{L}_q^{[E]}} \equiv 0$ . Somit folgt  $s|_{\mathcal{E}} \equiv 0$ , siehe Beweis zu Theorem 2.18. In den schwarzen Ketten werden, grob gesprochen, die Interpolantenstücke auf den viergeteilten Vierecken abhängig von den Interpolantenstücken auf den benachbarten zweigeteilten Vierecken bestimmt. Im ungünstigsten Fall liegt  $z$  deshalb in einem benachbarten Paar  $\{Q_5, Q_6\}$ ,  $Q_5 \cap Q_6 = f$ , zweigeteilter Vierecke, und zwar im Inneren desjenigen Mikrodreiecks, in dem alle inneren Bézier-Bernstein-Punkte gewählt sind. O. B. d. A. sei dies  $T_{5,1}$ . Es folgt  $s|_{T_{5,1}} \not\equiv 0$ . Durch die Differenzierbarkeitsbedingung über die durchgezogene Diagonale  $e_1$  von  $T_{1,1}$  ist im ungünstigsten Fall ferner  $s|_{T_{5,2}} \not\equiv 0$  auf dem zweiten Mikrodreieck von  $Q_5$ . Weiterhin ist wegen den Differenzierbarkeitsbedingungen über  $f$  und über die Diagonale  $e_2$  von  $Q_6$  der Interpolant im ungünstigsten Fall ungleich Null auf jedem Mikrodreieck  $T_{6,i}$ ,  $i = 1, 2$ , von  $Q_6$ .

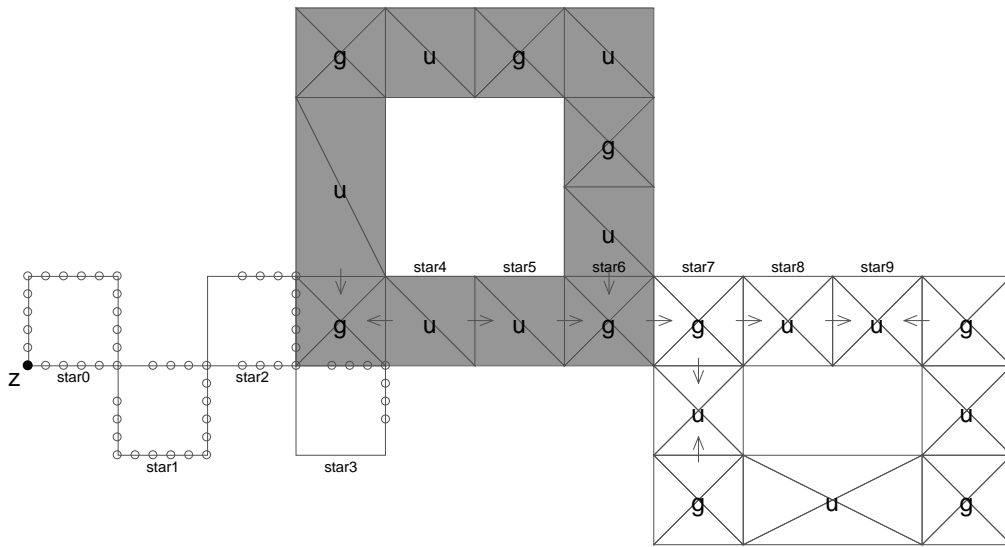


Abbildung 2.20: Maximale Trägergröße von  $\mathcal{B}_z$ , falls  $z$  auf einer Kante von  $\diamond$  liegt; gerade Vierecke sind mit „g“, ungerade mit „u“ gekennzeichnet

Nun hat das Paar  $\{Q_5, Q_6\}$  im ungünstigsten Fall genau zwei weitere schwarze Nachbarn  $\{Q_7, Q_8\}$ , und wegen der Ausbreitung über die zu  $Q_5$  bzw.  $Q_6$  gemeinsamen Kanten  $g$  und  $h$  ist der Interpolant auf diesen Vierecken im ungünstigsten Fall ebenfalls ungleich Null. Da sich nach den Argumenten aus dem Beweis zu Satz 2.18 die verbleibenden zweigeteilten schwarzen Vierecke unabhängig von allen bisher bestimmten Vierecken bestimmen lassen, folgt  $s|_Q \equiv 0$  auf allen verbleibenden schwarzen Vierecken  $Q$ .

Es bleibt noch die Ausbreitung in die weißen Vierecke zu untersuchen. Hierbei werden die ungeraden Vierecke abhängig von den jeweils benachbarten geraden Vierecken bestimmt. Im ungünstigsten Fall ist daher das schwarze Viereck  $Q_8$  mit einem geraden Viereck  $Q_9 = \tilde{Q}_{m-1}$  aus einer weißen geschlossenen Kette  $(\tilde{Q}_1, \dots, \tilde{Q}_m)$  ungerader Länge benachbart. Dann nämlich erhalten wir im ungünstigsten Fall zusätzliche Polynomstücke ungleich Null auf den Vierecken  $Q_{10} = \tilde{Q}_{m-1}$ ,  $Q_{11} = \tilde{Q}_m$  sowie  $Q_{12} = \tilde{Q}_1$  dieser Kette. Die Interpolantenstücke auf den verbleibenden weißen Vierecken werden unabhängig von den betrachteten Interpolantenstücken bestimmt. Daher ist der Träger von  $\mathcal{B}_z$  im ungünstigsten Fall im  $\text{star}^5(Q)$ , aber in keinem  $\text{star}^4(\tilde{Q})$ ,  $\tilde{Q} \in \diamond$ , enthalten, siehe Abbildung 2.19.

Fall 2:  $z$  liegt auf einer Kante von  $\diamond$

Die Bestimmung einer maximalen Anzahl von Kanten  $E$  mit  $s|_E \neq 0$  ist analog zum Fall  $q = 3$ . Die im Beweis zu Theorem 4.3 aus [68] verwendeten Argumente gelten entsprechend. Im ungünstigsten Fall liegt  $z$  auf einer Kante  $E$  eines Vierecks  $Q_1$  vom Typ 4, sodass der Interpolant  $\neq 0$  ist auf  $E$ , auf einer Kante eines  $Q_1$  berührenden Vierecks  $Q_2$  vom Typ 3, weiterhin auf einer Kante eines  $Q_2$  berührenden Vierecks  $Q_3$  vom Typ 2 und schließlich auf einer Kante eines  $Q_3$  berührenden Vierecks  $Q_4$  vom Typ 1. Eine solche Situation zeigt Abbildung 2.20. Zusammen mit den Argumenten aus Fall 1 folgt, dass der Träger von  $\mathcal{B}_z$  im ungünstigsten Fall im  $\text{star}^9(Q)$ , aber in keinem  $\text{star}^8(\tilde{Q})$ ,  $\tilde{Q} \in \diamond$ , enthalten ist.

(ii)

Wir verwenden die Notation aus (i) und bezeichnen mit  $T_{i,j}$ ,  $j = 1, \dots, 4$ , die Mikrodreiecke von  $Q_i$  für alle  $i = 7, \dots, 12$ . Mit den Lokalisierungsargumenten aus (i) zeigen wir, dass jeder Bézier-Bernstein-Koeffizient  $c_P$  aus einer Menge  $\mathcal{L}_P \subseteq \mathcal{L}_q \cap \text{star}^9_\diamond(Q)$  von Interpolationswerten in der Nähe von  $z$  bestimmt wird, sodass für eine geeignete, von der Wahl von  $z$  unabhängige Konstante  $C$  Abschätzung (2.8) gilt.  $C$  ist hierbei lediglich vom kleinsten Winkel  $\alpha$  von  $\diamond$  abhängig, insbesondere also nicht von Winkelsummen. Sei nun  $P \in \mathcal{P}_q(\diamond)$ .

Fall 1:  $z$  liegt auf einer Kante  $E$  von  $\diamond$ .

Angenommen es existiert ein  $z$  enthaltendes Viereck  $Q$  aus  $\diamond$ , bei dessen Durchlauf Punkte auf  $E$  gewählt werden. Dann betrachten wir zunächst den Fall, dass  $Q$  vom Typ 4 ist: Auf  $E$  befinden sich  $q + 1$  Interpolationspunkte  $z_i$ ,  $i = 0, \dots, q$ , sodass  $s|_E$  durch die zugehörigen Interpolationswerte  $f_{z_i}$ ,  $i = 0, \dots, q$ , eindeutig bestimmt ist. Insbesondere ist durch

$$f_{z_j} = \sum_{i=0}^q c_i B_i(z_j), \quad j = 0, \dots, q, \quad (2.9)$$

auch  $c_P$  eindeutig bestimmt. Hierbei bezeichnen  $B_i : E \rightarrow \mathbb{R}$ ,  $i = 0, \dots, q$ , die univariaten Bernsteinpolynome auf  $E$  und  $c_i$  die Bézier-Bernstein-Koeffizienten von  $s$  auf  $E$ . (2.9) lässt sich darstellen als  $\vec{f}_z = M_0 \vec{c}$  mit  $M_0 = (B_i(z_j))_{i,j=0,\dots,q}$ , und es ist bekannt, dass sich  $M^{-1}$  stabil berechnen lässt. Mit  $C_0 = \|M_0^{-1}\|$  folgt Gleichung (2.8) für  $c_P$ .

Seien nun  $Q$  vom Typ 3 und  $v$  der zum angrenzenden Viereck  $\tilde{Q}$  vom Typ 4 gemeinsame Eckpunkt. Liegt  $P$  auf einer Kante mit  $q + 1$  Punkten aus  $\mathcal{L}_q$ , so ist  $c_P$  wie im Fall  $l = 4$  stabil bestimmt. Andernfalls sind wie gesehen die Bézier-Bernstein-Koeffizienten auf den Kanten von  $\tilde{Q}$  stabil bestimmt. Wegen Korollar 1.5 existiert somit eine lediglich von  $\alpha$  abhängende Konstante  $C_1$ , sodass (2.8) für die beiden auf

dem Rand von  $Q$  liegenden Bézier-Bernstein-Koeffizienten mit Abstand eins zu  $v$  erfüllt ist. Auf den entsprechenden Kanten sind also jeweils zwei Bézier-Bernstein-Koeffizienten und weitere  $q - 1$  Interpolationswerte bekannt. Wegen den Beweisen zu Lemma 2.16, Lemma 2.17 und Theorem 2.18 folgt mit Gleichung (2.9), dass die verbleibenden Bézier-Bernstein-Koeffizienten in eindeutiger Weise bestimmt werden können. Dabei existiert eine Konstante  $C_2$ , sodass für sämtliche Koeffizienten auf den Kanten von  $Q$  mit Eckpunkt  $v$  Abschätzung (2.8) gilt.

Ist  $Q$  vom Typ 2, so kommen drei Kanten von  $Q$  in Betracht: Auf einer davon liegen  $q + 1$  Interpolationswerte, sodass die Argumente von Fall  $l = 4$  analog gelten. Auf den beiden anderen Kanten ist die Situation analog zum Fall  $l = 3$ , denn wegen den Lokalisierungsargumenten aus (i), Fall 2, wissen wir, dass genau zwei  $Q$  berührende Vierecke vom Typ 4 oder 3 sind. Bei den Vierecken  $Q$  vom Typ 1 kommen zwei Kanten in Betracht. Auch hier sind aufgrund der Lokalisierungseigenschaften die Fälle  $l = 3$  bzw.  $l = 2$  analog anwendbar.

Es verbleibt der Fall, dass  $z$  auf keinem Viereck aus  $\diamond$  liegt, bei dessen Durchlauf Punkte auf  $E = [a, b]$  gewählt werden. Dann sind die Koeffizienten  $c_a$  und  $c_b$  stabil bestimmt. Wegen Korollar 1.5 und den Lokalisierungsargumenten aus (i), Fall 2, existiert eine Konstante  $C_3$ , sodass Abschätzung (2.8) für die beiden zu  $z_a$  und  $z_b$  benachbarten Bézier-Bernstein-Koeffizienten auf  $E$  gilt. Des Weiteren liegen  $q - 3$  Interpolationswerte auf  $E$ . Wegen den Beweisen zu Lemma 2.16, Lemma 2.17 und Theorem 2.18 liefert dann Gleichung (2.9) eine stabile Berechnung der  $q - 3$  verbleibenden Bézier-Bernstein-Koeffizienten für eine lediglich von  $\alpha$  abhängende Konstante  $C_4$ .

Fall 2:  $z$  liegt im Inneren eines Vierecks  $Q = [v_1, \dots, v_4] \in \diamond$ .

Wegen Fall 1 wissen wir, dass die Koeffizienten auf den Kanten von  $Q$  stabil bestimmt sind. Sei  $Q$  nun zunächst ein Viereck vom Typ  $Q_5$  wie in (i) und sei  $[v_1, v_3]$  die durchgezogene Diagonale von  $Q$ . Dann existiert eine lediglich von  $\alpha$  abhängende Konstante  $C_5$ , sodass für  $z_{(q-1)v_1+v_3}$  und  $z_{v_1+(q-1)v_3}$  Abschätzung (2.8) gilt. Somit liegen vier stabil bestimmte Bézier-Bernstein-Koeffizienten und weiterhin  $q - 3$  Interpolationsbedingungen auf  $[v_1, v_3]$ . Daher folgt mit Gleichung (2.9) wegen den Beweisen zu Lemma 2.16, Lemma 2.17 und Theorem 2.18, dass eine lediglich von  $\alpha$  abhängende Konstante  $C_6$  existiert, sodass auch für die verbleibenden Bézier-Bernstein-Koeffizienten auf  $[v_1, v_3]$  Abschätzung (2.8) gilt. Liege  $z$  nun im Inneren von  $T_{5,1}$ . Auf allen inneren Bézier-Bernstein-Punkten von  $T_{5,1}$  sind Interpolationswerte gegeben, und die Rand-Bézier-Bernstein-Koeffizienten sind stabil bestimmt. Nun führt der Zusammenhang

$$f_{z_{ijk}} = \sum_{i'+j'+k'=q} c_{i'j'k'} B_{i'j'k'}(z_{ijk}), \quad i + j + k = q, \quad (2.10)$$

zwischen den auf  $T_{5,1}$  liegenden Interpolationswerten  $\vec{f}_z := (f_{z_{ijk}})_{i+j+k=q}$  und Bézier-

Bernstein-Koeffizienten  $\vec{c} := (c_{ijk})_{i+j+k=q}$  auf ein lineares Gleichungssystem der Form  $\vec{f}_z = M_2 \vec{c}$  mit  $M_2 = (B_{i'j'k'}(z_{ijk}))_{i+j+k=q, i'+j'+k'=q}$ . Aus diesem Zusammenhang folgt wegen den Beweisen zu Lemma 2.16, Lemma 2.17 und Theorem 2.18 die Existenz einer lediglich von  $\alpha$  abhängenden Konstanten  $C_7$ , sodass Abschätzung (2.8) für die im Inneren von  $T_{5,1}$  liegenden Bézier-Bernstein-Koeffizienten gilt. Liege  $z$  schließlich im Inneren von  $T_{5,2}$ . Wegen Theorem 1.4 folgt die Existenz einer lediglich vom kleinsten Winkel von  $Q$  abhängenden Konstanten  $C_8$ , welche für alle im Inneren von  $T_{5,2}$  liegenden Bézier-Bernstein-Koeffizienten mit Abstand eins zu  $[v_1, v_3]$  Abschätzung (2.8) erfüllt. Es verbleiben die  $(q-1)(q-2)/2$  noch nicht bestimmten Bézier-Bernstein-Koeffizienten auf  $T_{5,2}$ , auf welchen Lagrange-Bedingungen vorliegen. Wegen Gleichung (2.10) und den Beweisen zu Lemma 2.16, Lemma 2.17 sowie Theorem 2.18 existiert eine Konstante  $C_9$ , welche Abschätzung (2.8) für diese Koeffizienten erfüllt.

Nun betrachten wir den Fall, dass  $Q$  vom Typ  $Q_6$  wie in (i) ist. Sei hierbei  $[v_1, v_2]$  die mit  $Q_5$  gemeinsame Kante. Wie im Fall von  $Q_5$  sind auch hier die Bézier-Bernstein-Koeffizienten auf der durchgezogenen Diagonalen  $[v_1, v_3]$  aus den Interpolationswerten lokal und stabil bestimmt. Des Weiteren folgt mit Theorem 1.4 die Existenz einer lediglich von  $\alpha$  abhängenden Konstanten  $C_{10}$ , welche für die im Inneren von  $T_{6,1}$  liegenden Koeffizienten mit Abstand eins zu  $[v_1, v_2]$  Abschätzung (2.8) erfüllt. Analog zum Fall  $T_{5,2}$  gilt dies auch für die in  $T_{6,1}$  verbleibenden Bézier-Bernstein-Koeffizienten. Für das Dreieck  $T_{6,2}$  gelten dieselben Argumente wie im Fall  $T_{5,2}$ .

Es verbleiben die viergeteilten Vierecke. Wegen den Lokalisierungsargumenten aus (i) genügt es zu zeigen, dass alle auf einem viergeteilten Viereck  $Q$  verbleibenden Bézier-Bernstein-Koeffizienten mit den auf  $Q$  liegenden Interpolationswerten sowie den  $C^1$ -Bedingungen über die Kanten der Mikrodreiecke und das Baryzentrum von  $Q$  in stabiler Weise berechnet werden können. Wir verwenden hierbei Argumente und Notationen aus dem Beweis zu Lemma 2.16 und untersuchen die für jedes Mikrodreieck von  $Q$  möglichen Fälle, siehe Abbildung 2.13 auf Seite 49.

Sei  $Q$  ein viergeteiltes Viereck. Wir zeigen zunächst, dass sich alle Bézier-Bernstein-Koeffizienten auf den Diagonalen von  $Q$  stabil berechnen lassen und beginnen auf einer Kante  $[v_0, v_1]$ , sodass  $P_{1,q-1,0} \in \mathcal{L}_q$  ist. Wegen den Beweisen zu Lemma 2.16, Lemma 2.17 und Theorem 2.18 genügt es zu zeigen, dass auf jedem Bézier-Bernstein-Punkt von  $[v_0, v_1]$  der entsprechende Koeffizient oder eine Lagrange-Bedingung vorliegt. Mit (2.9) folgt daraus die stabile Bestimmung der auf  $[v_0, v_1]$  liegenden Bézier-Bernstein-Koeffizienten. Der Koeffizient  $c_{v_1}$  ist stabil bestimmt. Für den Koeffizienten  $c_{1,q-1,0}$  existiert wegen der  $C^1$ -Bedingung über  $v_1$  eine lediglich von  $\alpha$  abhängenden Konstante, sodass Abschätzung (2.8) erfüllt ist. Hinsichtlich  $P_{2,q-2,0}$  sind zwei Fälle möglich: Entweder ist dieser Punkt in  $\mathcal{L}_q$ , oder die Nachbarn bezüglich der Kanten  $[v_1, v_2]$  und  $[v_1, v_3]$  wurden algorithmisch zuerst durchlaufen. In letzterem Fall existiert wegen den  $C^1$ -Bedingungen über diese Kanten für die Koeffizienten  $c_{1,q-2,1}$  sowie  $c_{1,q-2,1}^{(2)}$  eine lediglich von  $\alpha$  abhängende Konstante, sodass Gleichung (2.8)

erfüllt ist. Wegen  $|c_{2,q-2,0}| \leq \max \left\{ |c_{1,q-2,1}|, |c_{1,q-2,q}^{(2)}| \right\}$  ist auch  $c_{1,q-2,0}$  stabil bestimmt. Ist  $q \geq 5$ , so liegt jeder Bézier-Bernstein-Punkt  $P_{i,q-i,0}$ ,  $i = 3, \dots, q-2$  in  $\mathcal{L}_q$ . Laut Voraussetzung ist dies auch für  $P_{q-1,1,0}$  der Fall. Schließlich ist  $v_0 \in \mathcal{L}_q$ . Mit Gleichung (2.9) lassen sich nun alle verbleibenden Bézier-Bernstein-Koeffizienten auf  $[v_0, v_1]$  berechnen, wobei eine lediglich von  $\alpha$  abhängende Konstante existiert, sodass (2.8) erfüllt ist. Wir betrachten die zweite Kante  $[v_0, v_i]$ ,  $i = 2, 3, 4$ , mit  $\frac{q-1}{q}v_0 + \frac{1}{q}v_i \in \mathcal{L}_q$ . Die Bézier-Bernstein-Koeffizienten auf dieser Kante sind aufgrund analoger Argumente stabil bestimmt. Insbesondere sind somit genau drei Bézier-Bernstein-Koeffizienten in der  $C^1$ -Scheibe um  $v_0$  stabil bestimmt. Folglich sind gemäß Korollar 1.5 die beiden verbleibenden Koeffizienten in dieser Scheibe durch die  $C^1$ -Bedingung über  $v_0$  stabil bestimmt. Mit obigen Argumenten sind also alle Bézier-Bernstein-Koeffizienten auf den Diagonalen von  $Q$  stabil bestimmt.

Es verbleiben die im Inneren der Mikrodreiecke  $T_1$  der viergeteilten Vierecke liegenden Bézier-Bernstein-Koeffizienten. Wir betrachten die Mikrodreiecke in derselben Reihenfolge wie im Beweis zu Lemma 2.16. Zunächst betrachten wir den Punkt  $P_{q-2,1,1}$ . Entweder ist ein zu  $T_1$  benachbartes Mikrodreieck von  $Q$  bereits bestimmt, und  $c_{q-2,1,1}$  lässt sich stabil durch die  $C^1$ -Bedingungen über die den Dreiecken gemeinsamen Kanten bestimmen, andernfalls ist  $P_{q-2,1,1} \in \mathcal{L}_q$ . Ähnlich ist die Situation für die Punkte  $P_{1,i,q-i-1}$ ,  $i = 1, \dots, q-2$ : Entweder liegen diese in  $\mathcal{L}_q$ , oder die Bézier-Bernstein-Koeffizienten auf dem zu  $T_1$  benachbarten Dreieck mit gemeinsamer Kante  $[v_1, v_2]$  sind bereits bestimmt, d. h. die Koeffizienten  $c_{q-2,1,1}$ ,  $i = 1, \dots, q-2$ , sind durch die  $C^1$ -Bedingungen über  $[v_1, v_2]$  bestimmt. Dasselbe gilt für die im Fall  $q \geq 5$  verbleibenden Bézier-Bernstein-Koeffizienten  $c_{i,q-i-1,1}$ ,  $c_{i,1,q-i-1}$ ,  $i = 2, \dots, q-3$ : Entweder sind die entsprechenden Punkte in der Menge  $\mathcal{L}_q$  enthalten, oder sie sind durch die  $C^1$ -Bedingungen über die Kante  $[v_0, v_1]$  bzw.  $[v_0, v_2]$  stabil bestimmbar. Ist  $q \geq 6$ , so liegen die noch verbleibenden Punkte  $P_{ijk}$ ,  $i + j + k = q$ ,  $i, j, k \in \{2, \dots, q-2\}$ , in  $\mathcal{L}_q$ . Folglich ist für jeden der  $(q+1)(q+2)/2$  Bézier-Bernstein-Punkte auf  $T_1$  der Interpolationswert oder der Bézier-Bernstein-Koeffizient in stabiler Weise bestimmt. Wegen den Beweisen zu Lemma 2.16, Lemma 2.17 sowie Theorem 2.18 und Gleichung (2.10) folgt daher die Existenz einer lediglich von  $\alpha$  abhängenden Konstanten  $C_{11}$ , sodass Abschätzung (2.8) für alle verbleibenden Bézier-Bernstein-Koeffizienten auf  $T_1$  erfüllt ist.

Mit den Lokalisierungsargumenten aus (i) folgt, dass eine lediglich von  $\alpha$  abhängende Konstante  $C$  existiert, mit welcher Gleichung (2.8) für alle  $z \in \mathcal{L}_q$  erfüllt ist. Dabei ist  $\mathcal{L}_P$  stets in einem geeigneten  $\text{star}^9(Q)$ ,  $Q \in \diamond$ , enthalten. Dies beweist Theorem 2.19.  $\square$

Ein bekanntes Ergebnis von Lai und Schumaker [57] besagt, dass aus der Lokalität und der Stabilität von Interpolationsverfahren mit Splines unter gewissen Voraussetzungen optimale Approximationsordnung folgt. Wir zeigen nun in ähnlicher Weise,

dass die dargestellten Methoden optimale Approximationsordnung besitzen (siehe auch [69]).

Im Wesentlichen verwenden wir die in den vorigen Abschnitten eingeführten Notationen. Bezeichne zudem  $W_\infty^{q+1}$  den Sobolevraum und  $If \in \mathcal{S}_q^r(\diamond)$  den Interpolanten einer Abbildung  $f \in W_\infty^{q+1}(\Omega)$ . Außerdem sei

$$|f|_{m,M} := \sum_{\alpha+\beta=m} \|D_x^\alpha D_y^\beta f\|_M, \quad 0 \leq m \leq q+1, \quad (2.11)$$

für jede kompakte Teilmenge  $M \subseteq \Omega$ .

**Theorem 2.20**

Seien  $q \geq 4$  und  $f \in W_\infty^{q+1}(\Omega)$ . Dann existiert eine lediglich vom kleinsten in  $\diamond$  auftretenden Winkel  $\alpha$  abhängende Konstante  $K$ , sodass die Bedingung

$$\|D_x^\alpha D_y^\beta (f - If)\|_\Omega \leq K h^{q+1-\alpha-\beta} |f|_{q+1,\Omega} \quad (2.12)$$

für alle  $0 \leq \alpha + \beta \leq q$  erfüllt ist, d. h. die Interpolationsmethode besitzt optimale Approximationsordnung. Dabei bezeichne  $h$  den größten Dreiecksumkreis in  $\Delta$ .

**Beweis:** Es seien  $f \in W_\infty^{q+1}(\Omega)$  und  $0 \leq \alpha + \beta \leq q$ . Ferner seien  $T$  ein Dreieck aus  $\diamond$  und  $Q$  dasjenige Viereck aus  $\diamond$ , welches  $T$  enthält. Weiterhin setzen wir  $\Omega_Q = \text{star}_\diamond^Q(Q)$  als Träger mit maximaler Größe für Basissplines, wie sie nach Theorem 2.19 in der Interpolationsmethode auftreten können.

Wir schätzen den Term  $\|D_x^\alpha D_y^\beta (f - If)\|_T$  zunächst nur für  $T$  und dann für die gesamte Quadrangulierung. Bekanntermaßen (vgl. Lai und Schumaker [57], Lemma 4.6) existiert ein Polynom  $p \in \mathcal{P}_q^2$  mit der Eigenschaft

$$\|D_x^\alpha D_y^\beta (f - p)\|_{\Omega_Q} \leq K_0 \Sigma_{\Omega_Q}^{q+1-\alpha-\beta} |f|_{q+1,\Omega_Q} \quad (2.13)$$

für alle  $0 \leq \alpha + \beta \leq q$ , wobei  $\Sigma_{\Omega_Q}$  den Durchmesser von  $\Omega_Q$  bezeichne und  $K_0$  absolut konstant ist. Wegen der Wahl von  $\Omega_Q$  ist  $19h$  eine obere Schranke für  $\Sigma_{\Omega_Q}$ , und es folgt

$$\|D_x^\alpha D_y^\beta (f - p)\|_{\Omega_Q} \leq K_1 h^{q+1-\alpha-\beta} |f|_{q+1,\Omega_Q}, \quad (2.14)$$

wobei  $K_1$  als  $19^{q+1}K_0$  gewählt wird. Da  $I$  eine Projektion und  $\mathcal{P}_q^2 \subset \mathcal{S}_q^1(\diamond)$  ist, folgt  $Ip = p$ , und mit der Dreiecksungleichung gilt:

$$\|D_x^\alpha D_y^\beta (f - If)\|_T \leq \|D_x^\alpha D_y^\beta (f - p)\|_T + \|D_x^\alpha D_y^\beta I(f - p)\|_T.$$

Für den ersten Summanden liegt mit Ungleichung (2.14) bereits eine obere Schranke vor. Folglich genügt die Abschätzung des zweiten Terms. Die Markov-Ungleichung (vgl. Wilhelmsen [87]) liefert

$$\|D_x^\alpha D_y^\beta I(f - p)\|_T \leq K_2 h^{-\alpha-\beta} \|I(f - p)\|_T, \quad (2.15)$$



wobei  $K_2$  eine geeignete Konstante ist, welche nur von  $\alpha$  abhängt. Also bleibt der Term  $\|I(f - p)\|_T$  abzuschätzen. Sei  $I(f - p)|_T \equiv \sum_{i+j+k=q} c_{ijk} B_{ijk}^q$ ,  $c_{ijk} \in \mathbb{R}$ ,  $i+j+k = q$ . Mit den Eigenschaften der Bézier-Bernstein-Polynome, siehe Kapitel 1, folgt

$$\|I(f - p)\|_T \leq \max_{i+j+k=q} |c_{ijk}|. \quad (2.16)$$

Wegen Theorem 2.19, (ii) existiert eine nur von  $\alpha$  abhängende Konstante  $C$ , sodass für alle  $i+j+k = q$  die Abschätzung

$$|c_{ijk}| \leq C \max_{\tilde{z} \in \mathcal{L}_q \cap \Omega_Q} |(f - p)(\tilde{z})| \leq K_3 \|f - p\|_{\Omega_Q} \quad (2.17)$$

gilt, wobei auch die Konstante  $K_3$  lediglich von  $\alpha$  abhängt. Somit lässt sich die Abschätzung (2.17) auf Ungleichung (2.16) anwenden, und mit (2.13) bis (2.15) erhält man

$$\|D_x^\alpha D_y^\beta (f - If)\|_T \leq K_4 h^{q+1-\alpha-\beta} |f|_{q+1, \Omega_Q} \quad (2.18)$$

für eine geeignete, lediglich von  $\alpha$  abhängende Konstante  $K_4$ .

Bei Anwendung dieser Ungleichung auf jedes Dreieck  $T \in \diamond$  ergibt sich ein Maximum, und somit folgt Ungleichung (2.12) mit einer Konstanten  $K_5$ , welche lediglich von  $\alpha$  abhängt.  $\square$

### 2.3.3 Interpolation bei spezieller Färbung

Wir betrachten nun die Interpolation mit Färbungen wie in [68] und mit Färbungen des in Abschnitt 2.2 entwickelten Algorithmus. Dabei betrachten wir einerseits beliebige Quadrangulierungen und andererseits Quadrangulierungen mit maximaler Kettenlänge zwei. Eine Zusammenfassung der Unterschiede hinsichtlich wesentlicher Interpolationseigenschaften geben wir in Abschnitt 2.5.

Wir untersuchen zunächst den Einfluss von Färbungen auf die Lokalität bei der Bestimmung der Interpolanten: Wie gesehen, sind die Träger der jeweiligen Lagrange-Basis-splines maximal in einem  $\text{star}^9(Q)$ ,  $Q \in \diamond$ , enthalten, falls gleichfarbige Vierecke stets in Ketten auftreten. Bei der in Abschnitt 2.2 entwickelten Färbungsmethode treten darüber hinaus keine geschlossenen Ketten auf. Mit ähnlichen Argumenten wie in den Beweisen zu Theorem 4.3 in [68] und zu Theorem 2.19 folgt auf beliebigen Quadrangulierungen  $\text{star}^7(Q)$ ,  $Q \in \diamond$ , für die maximale Trägergröße der Basis-splines. Für den folgenden Beweis verwenden wir die Notationen aus Theorem 2.19.

#### Theorem 2.21

*Sei  $q \geq 3$ . Treten bei der Interpolation keine geschlossenen Ketten gleicher Farbe in  $\diamond$  auf, so gilt die folgende Aussage: Für jeden Punkt  $z \in \mathcal{L}_q$  ist der Träger des zugehörigen Basis-splines  $\mathcal{B}_z$*

1. *in einem  $\text{star}^3(Q)$ ,  $Q \in \diamond$ , enthalten, falls  $z$  im Inneren eines Vierecks von  $\diamond$  liegt.*
2. *in einem  $\text{star}^7(Q)$ ,  $Q \in \diamond$ , enthalten, falls  $z$  auf den Kanten von  $\diamond$  liegt.*

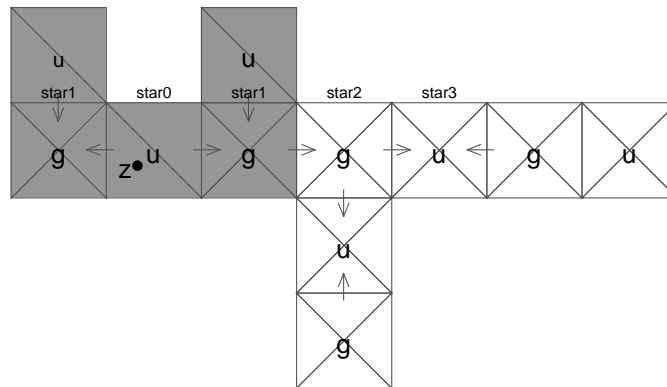


Abbildung 2.21: Maximale Trägergröße von  $\mathcal{B}_z$ , falls  $z$  im Innern eines Vierecks liegt; gerade Vierecke sind mit „g“, ungerade mit „u“ gekennzeichnet

**Beweis:** Als Viereck  $Q$  des  $\text{star}^m(Q)$ ,  $m \in \mathbb{N}_0$ , wählen wir möglichst dasjenige den Punkt  $z$  enthaltende Viereck  $Q \in \diamond$ , welches im Kantenalgorithmus zuerst durchlaufen wurde, ansonsten wählen wir ein beliebiges Viereck, welches  $z$  enthält.

Wir wenden die Argumente aus dem Beweis zu Theorem 2.19 auf den Fall an, dass auftretende Ketten gleicher Farbe nicht geschlossen sind. Dann existieren keine Paare ungerader Vierecke innerhalb der weißen und schwarzen Ketten, und der ungünstigste Fall lautet wie folgt:

Liege wieder  $z$  zunächst im Inneren eines Vierecks  $Q \in \diamond$ . Im ungünstigsten Fall ist dies ein ungerades schwarzes Viereck mit benachbartem schwarzen Viereck  $Q'$ , sodass  $\mathcal{B}_z|_{Q'} \neq 0$  ist und  $Q'$  einen geraden weißen Nachbarn  $Q''$  mit  $\mathcal{B}_z|_{Q''} \neq 0$  besitzt. Im ungünstigsten Fall besitzt dann  $Q''$  einen weißen ungeraden Nachbarn  $Q'''$  mit  $\mathcal{B}_z|_{Q'''} \neq 0$ . Abbildung 2.21 zeigt die maximale Ausbreitung, falls  $z$  im Inneren eines Vierecks liegt. Der Träger von  $\mathcal{B}_z$  ist im ungünstigsten Fall folglich im  $\text{star}^3(Q)$ , aber in keinem  $\text{star}^2(\tilde{Q})$ ,  $\tilde{Q} \in \diamond$ , enthalten.

Wir betrachten nun den Fall, dass  $z$  auf den Kanten von  $\diamond$  liegt. Diese Situation ist analog wie im Beweis zu Theorem 2.19. Insgesamt ist der Träger von  $\mathcal{B}_z$  daher im ungünstigsten Fall im  $\text{star}^7(Q)$ , aber in keinem  $\text{star}^6(\tilde{Q})$ ,  $\tilde{Q} \in \diamond$ , enthalten. Diese Situation zeigt Abbildung 2.22.  $\square$

Für die Anwendung von Approximationsverfahren spielt die zur Darstellung eines Interpolanten benötigte Datenmenge eine wesentliche Rolle. Bei der Spline-Interpolation ist die benötigte Datenmenge proportional zur Dimension des Raumes  $\mathcal{S}_q^r(\diamond)$ . Daher ist bei den dargestellten Interpolationsmethoden von Vorteil, wenn möglichst wenige Vierecke aus  $\diamond$  in vier Mikrodreiecke unterteilt werden und möglichst viele

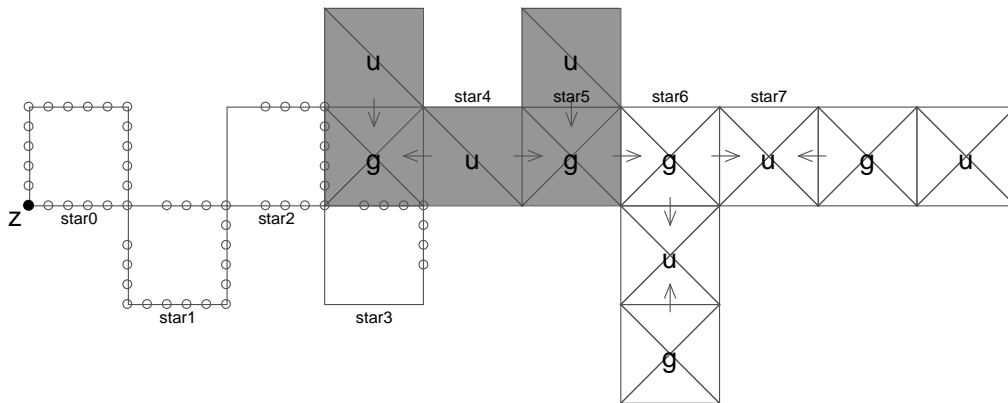


Abbildung 2.22: Maximale Trägergröße von  $\mathcal{B}_z$ , falls  $z$  auf einer Kante von  $\diamond$  liegt; Gerade Vierecke sind mit „g“, ungerade mit „u“ gekennzeichnet.

in zwei. Seien im Folgenden  $n_4$  die Anzahl der viergeteilten sowie  $n_2$  die Anzahl der zweigeteilten Vierecke einer Quadrangulierung  $\diamond$ , und sei  $n = |\diamond|$ . Wir betrachten die Verhältnisse  $n_2/n$  sowie  $n_4/n$ .

Sowohl die Färbungsmethode von Nürnberger, Schumaker und Zeilfelder [68] als auch die von uns entwickelten Verfahren machen keinen Unterschied zwischen den Farben schwarz und weiß. Daher ist bei beliebigen Quadrangulierungen durchschnittlich die Hälfte der Vierecke weiß. Wegen der Tatsache, dass jedes weiße Viereck viergeteilt wird, ist im Schnitt mindestens jedes zweite Viereck viergeteilt. Insbesondere ist für den Fall, dass alle schwarzen Vierecke zweigeteilt werden, also stets einzeln, d. h. ohne schwarze Nachbarn auftreten, im Schnitt genau die Hälfte der Vierecke viergeteilt. Andererseits wird der Anteil der viergeteilten Vierecke maximal, falls jedes schwarze Viereck in einer gleichfarbigen Kette von gerader Länge liegt. In diesem Fall wird im Schnitt genau die Hälfte der schwarzen Vierecke zweigeteilt, d. h. es sind drei Viertel der Vierecke viergeteilt. Für den Vergleich mit den anderen Methoden nehmen wir im Folgenden an, dass im Schnitt zwei Drittel der schwarzen Vierecke zweigeteilt sind.

Sei nun  $n_2$  die Anzahl der zweigeteilten,  $n_4$  die Anzahl der viergeteilten Vierecke. Dann folgt für die Anzahl  $N$  der in  $\diamond$  enthaltenen Dreiecke  $N = 2n_2 + 4n_4$ , wir erhalten mit obiger Schätzung also

$$N \approx 2 \cdot \frac{1}{3}n + 4 \cdot \frac{2}{3}n = 3\frac{1}{3}n. \tag{2.19}$$

Dieser Wert und somit die von Interpolanten benötigte Datenmenge lässt sich durch eine geringfügige Modifikation der ursprünglichen Interpolationsmethode verringern:

Bei der Triangulierung von  $\diamond$  fügen wir für jedes Paar  $\{Q_1, Q_2\}$  schwarzer Vierecke ohne weitere schwarze Nachbarn jeweils nur eine Diagonale zu  $Q_1$  und  $Q_2$  hinzu. Abgesehen davon triangulieren wir wie oben beschrieben. Auf der resultierenden Triangulierung  $\diamond$  konstruieren wir eine Interpolationsmenge  $\mathcal{L}_q$  wie in den Abschnitten 2.3.1 und 2.3.2 beschrieben.

Es ergibt sich folgendes Interpolationsverfahren mit stärkeren Lokalitätseigenschaften:

**Algorithmus 2.22**

Seien  $q \geq 3$  und  $\diamond$  eine nach Algorithmus 2.4 gefärbte, strikt konvexe Quadrangulierung. In den schwarzen Ketten der Länge  $\neq 2$  füge man zu jedem Viereck mit geradem Index beide Diagonalen und zu jedem Viereck mit ungeraden Index eine Diagonale hinzu. Für jede schwarze Kette der Länge zwei füge man zu jedem Viereck eine Diagonale hinzu. Für jedes weiße Viereck füge man beide Diagonalen hinzu. Auf der resultierenden Triangulierung  $\diamond$  wähle man Interpolationspunkte wie in den Algorithmen 2.10 und 2.11 ( $q = 3$ ) bzw. den Algorithmen 2.13 und 2.15 ( $q \geq 4$ ) beschrieben.

Offensichtlich bleibt  $\mathcal{L}_q$  trotz dieser Modifikation eine Lagrange-Interpolationsmenge. Weiterhin sind die Interpolanten lokal und stabil bestimmbar, und das Verfahren besitzt optimale Approximationsordnung. Alle Voraussetzungen der entsprechenden Beweise, für  $q = 3$  auch in [68], sind erfüllt. Hinsichtlich der Lokalität des Verfahrens stellt sich die Frage nach einer Veränderung der maximalen Trägergröße von Basissplines. Zusätzlich zu den in Theorem 2.21 betrachteten Fällen ist die Situation zu berücksichtigen, dass eine Stelle  $\mathcal{B}_z(z_1) \neq 0$  im Inneren eines zweigeteilten Vierecks  $Q_1$  aus einer Kette  $(Q_1, Q_2)$  schwarzer Vierecke der Länge zwei auftritt. Zwar entstehen dadurch im ungünstigsten Fall Werte  $\mathcal{B}_z(z_2) \neq 0$  an Stellen  $z_2$  im Inneren des zweigeteilten Vierecks  $Q_2$ , allerdings bleiben die Splinestücke im Inneren aller verbleibenden schwarzen Vierecke davon unbeeinflusst, denn  $Q_1$  und  $Q_2$  haben laut Voraussetzung keine weiteren schwarzen Nachbarn in  $\diamond$ . Somit bleibt die maximale Trägergröße der Lagrange-Basissplines unverändert.

Daher ergeben sich für die gewählten Interpolationspunkte  $\mathcal{L}_q$  folgende Eigenschaften:

**Theorem 2.23**

*Sei  $q \geq 3$ .  $\mathcal{L}_q$  ist eine minimal bestimmende Menge und ferner eine Lagrange-Interpolationsmenge für den Raum  $\mathcal{S}_q^1(\diamond)$ . Die Interpolanten sind lokal und stabil bestimmbar und besitzen optimale Approximationsordnung. Für jeden Basisspline  $\mathcal{B}_z$ ,  $z \in \mathcal{L}_q$ , existiert ein Viereck  $Q \in \diamond$  mit der folgenden Eigenschaft:*

$$\text{supp}(\mathcal{B}_z) \subseteq \text{star}^7(Q)$$

Im Allgemeinen liefert Algorithmus 2.4 einen sehr geringen Anteil von Vierecken  $Q \in \diamond$  in Ketten der Länge  $\geq 3$ . Insbesondere treten in Quadrangulierungen der Klasse  $\mathcal{K}_2$  keine solche Ketten auf, d. h. jedes schwarze Viereck ist zweigeteilt. Folglich ist durchschnittlich genau die Hälfte der Vierecke viergeteilt. Für diese Klasse gilt daher im Schnitt

$$N \approx 3n. \quad (2.20)$$

## 2.4 Interpolation und Prioritätsprinzip

Wir entwickeln nun eine weitere Methode der lokalen Lagrange-Interpolation mit bivariaten  $C^1$ -Splines auf strikt konvexen triangulierten Quadrangulierungen  $\diamond$ . Im Unterschied zu den bisher behandelten Methoden ist hierbei keine Färbung der Vierecke nötig.

Wieder betrachten wir das Interpolationsproblem 2.8 bezüglich des Raums  $\mathcal{S}_q^1(\diamond)$  der  $C^1$ -Splines vom Grad  $q \geq 3$  auf einer triangulierten strikt konvexen Quadrangulierung  $\diamond$  des polygonalen Gebiets  $\Omega$ .

Ohne die Vierecke zu färben, wenden wir zunächst Kantenalgorithmus 2.10 ( $q = 3$ ) bzw. 2.13 ( $q \geq 4$ ) auf  $\diamond$  an. Die Menge der dabei gewählten Interpolationspunkte bezeichnen wir mit  $\mathcal{L}_q^{[E]}$ . Wie in Abschnitt 2.3 ausführlich dargestellt, ist der Interpolant  $s$  damit auf den Kanten von  $\diamond$  bestimmt. Im folgenden Algorithmus wird  $s$  durch die Wahl weiterer Interpolationspunkte auch im Inneren der Vierecke von  $\diamond$  in eindeutiger Weise bestimmt, ohne dass eine Färbung von  $\diamond$  benötigt wird. Die Vierecke von  $\diamond$  werden durch Hinzufügen von einer oder zwei Diagonalen unterteilt.

Wir beschreiben zunächst, welche Interpolationspunkte in den einzelnen Mikrodreiecken gewählt werden:

### Algorithmus 2.24

Seien  $q \geq 3$  und  $Q = [v_1, v_2, v_3, v_4]$  ein Viereck mit  $l \in \{0, \dots, 4\}$  bereits gewählten Nachbarvierecken. Dabei bezeichne  $v_0$  das Baryzentrum von  $Q$ . Die zu den gewählten Vierecken gemeinsamen Kanten von  $Q$  seien mit  $E_1, \dots, E_l$  bezeichnet. Für  $l \in \{0, 1\}$  füge man eine Diagonale  $[v_1, v_3]$ , für  $l \in \{2, 3, 4\}$  füge man beide Diagonalen zu  $Q$  hinzu. In der entstehenden Unterteilung  $\{T_1, T_2\}$  bzw.  $\{T_1, \dots, T_4\}$  wähle man folgende Interpolationspunkte:

$l = 0$ :

Auf der durchgezogenen Diagonalen von  $Q$  wähle man alle Bézier-Bernstein-Punkte mit Abstand  $\geq 2$  zu den Eckpunkten von  $Q$ . In  $T_1$  wähle man alle inneren Bézier-Bernstein-Punkte. In  $T_2$  wähle man alle inneren Bézier-Bernstein-Punkte mit Abstand  $\geq 2$  zu  $[v_1, v_3]$ .

$l = 1$ :

Man wähle alle Punkte wie in Fall  $l = 0$  beschrieben, ausgenommen der Bézier-Bernstein-Punkte mit Abstand eins zu  $E_1$ .

$l \in \{2, 3, 4\}, q = 3$ :

Für  $l = 4$  wähle man keine Punkte. Ist  $l \in \{2, 3\}$ , so wähle man  $v_0$ . Im Fall  $l = 2$  wähle man außerdem einen Bézier-Bernstein-Punkt mit Abstand eins zu  $v_0$ . Besitzen  $E_1$  und  $E_2$  einen gemeinsamen Eckpunkt  $v_j$  von  $Q$ , so ist ein Bézier-Bernstein-Punkt von  $Q$  auf der Diagonalen  $e$  mit der Eigenschaft  $v_j \notin e$  zu wählen.

$l \in \{2, 3, 4\}, q \geq 4$ :

Für jedes Mikrodreieck  $T$  von  $Q$  wähle man Interpolationspunkte wie in Algorithmus 2.14 beschrieben. Man beginne hierbei mit denjenigen Mikrodreiecken, welche Kanten aus  $\{E_1, \dots, E_l\}$  besitzen.

Wir wählen nun Interpolationspunkte im Inneren der Vierecke aus  $\diamond$ :

### Algorithmus 2.25

Sei  $q \geq 3$ . Für  $l = 0, \dots, 4$  gehe man wie folgt vor: Solange ein noch nicht markiertes Viereck  $Q \in \diamond$  existiert, welches genau  $l$  bereits gewählte Nachbarvierecke aus  $\diamond$  besitzt, wähle man auf  $Q$  die in Algorithmus 2.24 beschriebenen Punkte und bezeichne das Viereck als markiert.

Die Menge aller im Inneren der Vierecke gewählten Punkte nennen wir  $\mathcal{L}_q^{[l]}$ . Weiterhin bezeichnen wir für  $l = 0, \dots, 4$  mit  $\mathcal{K}_l$  die Menge aller in der Iteration  $l$  gewählten Vierecke.

Durch diese Vorgehensweise werden auf den unterteilten Vierecken offensichtlich gleiche Punkte gewählt wie bei der Interpolation auf gefärbten Quadrangulierungen in Abschnitt 2.3. Da die Vierecke nach dem Prioritätsprinzip in abweichender Reihenfolge durchlaufen werden, ergeben sich jedoch im Allgemeinen andere Interpolationsmengen.

Wir zeigen zunächst, dass die Menge  $\mathcal{L}_q$ ,  $q \geq 3$ , der Interpolationspunkte eine minimal bestimmende Menge ist. Dabei sei bemerkt, dass es sich bei sämtlichen in  $\mathcal{L}_q$  enthaltenen Punkten um Bézier-Bernstein-Punkte handelt.

### Lemma 2.26

Sei  $q \geq 3$ . Die Menge  $\mathcal{L}_q := \mathcal{L}_q^{[E]} \cup \mathcal{L}_q^{[I]}$  ist eine minimal bestimmende Menge für den Raum  $\mathcal{S}_q^1(\diamond)$ .

**Beweis:** Wir zeigen, dass durch die Wahl der zu  $\mathcal{L}_q$  gehörenden Bézier-Bernstein-Koeffizienten der Interpolant  $s$  in eindeutiger Weise bestimmt ist und insbesondere keine Überbestimmung von Koeffizienten vorliegt. Seien also alle Bézier-Bernstein-Koeffizienten der Punkte aus  $\mathcal{L}_q$  vorgegeben.

Insbesondere sind dadurch die zu  $\mathcal{L}_q^{[E]}$  gehörenden Koeffizienten vorgegeben. Wegen der hier analog geltenden Aussage 1 aus dem Beweis zu Lemma 2.16 sind durch die  $C^1$ -Bedingungen über die Eckpunkte von  $\diamond$  genau die folgenden Bézier-Bernstein-Koeffizienten von  $s$  eindeutig bestimmt:

1. alle Bézier-Bernstein-Koeffizienten auf den Kanten von  $\diamond$
2. alle verbleibenden Bézier-Bernstein-Koeffizienten auf den Kanten von  $\diamond$  mit Abstand eins zu einem Eckpunkt von  $\diamond$

Dies gilt analog für den Fall  $q = 3$ . Wir betrachten zunächst die durch eine Diagonale unterteilten Vierecke:

**Behauptung 1:** Mit den verbleibenden  $C^1$ -Bedingungen über die durchgezogenen Diagonalen der zweigeteilten Vierecke lassen sich alle Koeffizienten auf den zweigeteilten Vierecke in eindeutiger Weise bestimmen.

Sei  $Q \in \diamond$  ein zweigeteiltes Viereck. Es gilt Behauptung 2 aus dem Beweis zu Lemma 2.16 analog. Dabei entspricht der Fall  $l = 0$  der Situation eines einzelnen zweigeteilten Vierecks, der Fall  $l = 1$  entspricht dem Viereck  $Q_2$  aus einem benachbarten Paar zweigeteilter Vierecke. Somit folgt Behauptung 1. Wieder gelten die Argumente analog für den Fall  $q = 3$ .  $\square$

Es verbleiben die viergeteilten Vierecke. Im Fall  $q = 3$  lassen sich die Argumente von Nürnberger, Schumaker und Zeilfelder [68], Theorem 3.4 analog anwenden. Im Fall  $q \geq 4$  gelten die Behauptungen 3 und 4 aus dem Beweis zu Lemma 2.16, denn die im Inneren von  $Q$  liegenden Punkte aus  $\mathcal{L}_q^{[I]}$  werden gemäß Algorithmus 2.14 gewählt.

Induktiv folgt, dass mit jedem durchlaufenen Viereck  $Q$  alle Koeffizienten auf  $Q$  unter Verwendung sämtlicher  $C^1$ -Bedingungen über den Kanten zu bereits durchlaufenen Vierecken in eindeutiger Weise bestimmt sind. Somit ist  $s$  unter Verwendung aller  $C^1$ -Bedingungen in eindeutiger Weise bestimmt.  $\square$

### Theorem 2.27

Sei  $q \geq 3$ .  $\mathcal{L}_q$  ist eine Lagrange-Interpolationsmenge für den Raum  $\mathcal{S}_q^1(\diamond)$ .

**Beweis:** Wegen Lemma 2.26 ist  $|\mathcal{L}_q| = \dim \mathcal{S}_q^1(\diamond)$ , und es genügt, die Eindeutigkeit des homogenen Interpolationsproblems zu zeigen. Sei also  $s(z) = 0$  auf allen Punkten  $z \in \mathcal{L}_q$ .

Da auf den Kanten von  $\diamond$  dieselben Interpolationspunkte gewählt werden wie in den Algorithmen 2.10 und 2.13, gilt Behauptung 1 aus dem Beweis zu Lemma 2.18 analog, d. h. für jeden Punkt  $z$  auf den Kanten von  $\diamond$  und auf den durchgezogenen Diagonalen der zweigeteilten Vierecke ist  $s(z) = 0$ .

Auch für jedes zweigeteilte Viereck  $Q \in \diamond$  lassen sich die dortigen Argumente analog anwenden: Abbildung 2.18 auf Seite 63 zeigt, dass sowohl im Fall  $l = 0$  (Abbildung links) als auch im Fall  $l = 1$  (Abbildung rechts) durch die Wahl der Punkte  $\mathcal{L}_q^{[l]}|_Q$  folgt, dass das Splinstück  $s|_Q \equiv 0$  ist. Dies gilt in gleicher Weise für den Fall  $q = 3$ .

Es folgt  $s|_{\Omega_{\mathcal{M}}} \equiv 0$  auf der Menge  $\mathcal{M}$  aller zweigeteilten Vierecke. Da auf den viergeteilten Vierecken dieselben Mengen von Lagrange-Punkten gewählt werden wie bei der Interpolationsmethode auf gefärbten Vierecken (vgl. Algorithmus 2.25), gilt Lemma 2.17 hier völlig analog. Es folgt  $s \equiv 0$ . Im kubischen Fall gelten wiederum die Argumente aus [68], Beweis zu Theorem 3.4.  $\square$

Auch dieses Interpolationsverfahren ist lokal. Der Träger jedes Lagrange-Basis-splines ist in einem  $\text{star}^8(Q)$ ,  $Q \in \diamond$ , enthalten:

**Theorem 2.28**

Sei  $q \geq 3$ . Für jeden Punkt  $z \in \mathcal{L}_q$  ist der Träger des zugehörigen Basis-splines  $\mathcal{B}_z$  mit  $\mathcal{B}_z(\tilde{z}) = \delta_{z\tilde{z}}$  für alle  $\tilde{z} \in \mathcal{L}_q$

1. in einem  $\text{star}^4(Q)$ ,  $Q \in \diamond$ , enthalten, falls  $z$  im Inneren eines Vierecks  $Q \in \diamond$  liegt.
2. in einem  $\text{star}^8(Q)$ ,  $Q \in \diamond$ , enthalten, falls  $z$  auf den Kanten von  $\diamond$  liegt.

*Inbesondere ist die Interpolationsmethode lokal.*

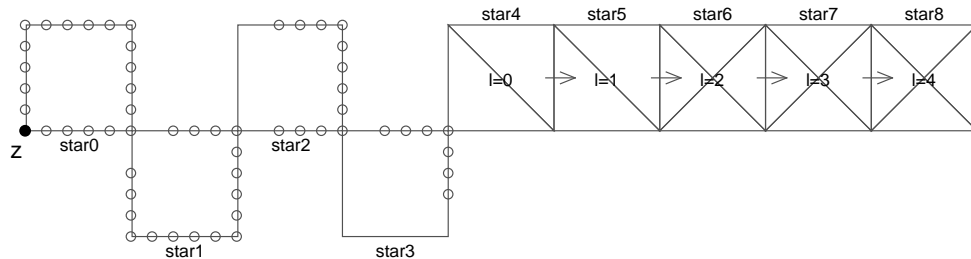
**Beweis:** Sei  $z \in \mathcal{L}_q$ . Wir zeigen

$$\text{supp}(\mathcal{B}_z) \subseteq \text{star}^k(Q),$$

wobei  $k = 4$  für den Fall 1 bzw.  $k = 8$  für den Fall 2 ist. Weiterhin sei  $Q$  dasjenige den Punkt  $z$  enthaltende Viereck aus  $\diamond$ , welches in Algorithmus 2.25 zuerst durchlaufen wurde.

Wir untersuchen die *maximale* Größe, wie sie für Träger der Basis-splines  $\mathcal{B}_z$  auftreten kann:




 Abbildung 2.23: Maximale Trägergröße von  $\mathcal{B}_z$ 

Fall 1:  $z$  liegt im Inneren eines Vierecks  $Q \in \diamond$

Wir zeigen zunächst folgende

**Behauptung 1:** Keine der Klassen  $\mathcal{K}_i$ ,  $i = 0, \dots, 4$ , enthält benachbarte Vierecke.

Alle folgenden Argumente stützen sich auf die Vorgehensweise in Algorithmus 2.25. Wir beweisen durch Widerspruch: Für  $i = 0, \dots, 4$  sei  $\{Q_i, \tilde{Q}_i\} \in \mathcal{K}_i \times \mathcal{K}_i$  ein Paar von Vierecken aus  $\mathcal{K}_i$ , wobei o. B. d. A.  $Q_i$  jeweils das vor  $\tilde{Q}_i$  gewählte Viereck sei. Wir nehmen nun an, dass  $Q_i$  und  $\tilde{Q}_i$  benachbart sind.

Da  $\tilde{Q}_0$  keine gemeinsamen Kanten mit zuvor gewählten Vierecken hat, ist es insbesondere nicht mit  $Q_0$  benachbart, und wir erhalten einen Widerspruch. Also existieren keine benachbarten Vierecke in  $\mathcal{K}_0$ . Das Viereck  $\tilde{Q}_1 \in \mathcal{K}_1$  besitzt genau ein zuvor gewähltes und benachbartes Viereck, nämlich  $Q_1$ . Wegen  $Q_1 \in \mathcal{K}_1$  wäre  $\tilde{Q}_1$  jedoch zu  $\mathcal{K}_0$  hinzugefügt worden, und wir erhalten einen Widerspruch. Für die Klasse  $\mathcal{K}_2$  lässt sich analog argumentieren: Das Viereck  $\tilde{Q}_2$  besitzt außer  $Q_2$  genau einen zuvor gewählten Nachbarn, d. h.  $\tilde{Q}_2$  wäre zu  $\mathcal{K}_1 \neq \mathcal{K}_2$  hinzugefügt worden. Ebenso besitzt  $\tilde{Q}_3$  außer  $Q_3$  genau zwei zuvor gewählte Nachbarn, und es folgt  $\tilde{Q}_3 \in \mathcal{K}_2 \neq \mathcal{K}_3$ . Schließlich besitzt  $\tilde{Q}_4$  außer  $Q_4$  genau drei zuvor gewählte Nachbarn, und es folgt  $\tilde{Q}_4 \in \mathcal{K}_3 \neq \mathcal{K}_4$ .  $\square$

Analog zum Beweis von Theorem 2.27 ist  $\mathcal{B}_z|_{\mathcal{E}} \equiv 0$  auf den Kanten  $\mathcal{E}$  von  $\diamond$ . Wie der Beweis zu Theorem 2.27 zeigt, kann ein Interpolantenstück  $\mathcal{B}_z|_Q$  über einem Viereck  $Q$  deshalb nur dann Funktionswerte ungleich Null besitzen, falls ein Interpolantenstück eines gemäß Prioritätsprinzip zuvor gewählten Nachbarvierecks von  $Q$  Funktionswerte ungleich Null besitzt. Mit dieser Überlegung, zusammen mit Behauptung 1, ergibt sich der Fall einer maximalen Trägergröße für  $\mathcal{B}_z$  wie folgt:

Der Punkt  $z$  liegt im Inneren eines Vierecks  $Q_0 \in \mathcal{K}_0$ , d. h. es ist  $\mathcal{B}_z|_{Q_0} \neq 0$ . Im ungünstigsten Fall enthält  $\diamond$  eine (nicht geschlossene) Kette  $\{Q_0, \dots, Q_4\}$  von Vierecken, wobei  $Q_i \in \mathcal{K}_i$  ist für alle  $i = 0, \dots, 4$ . Sei für  $i = 0, \dots, 3$  mit  $E_i$  ferner die gemeinsame Kante  $Q_i \cap Q_{i+1}$  bezeichnet. Algorithmisch besteht die Möglichkeit, dass  $E_i \cap E_j = \emptyset$  ist für alle Paare  $j \neq i$ ,  $i, j = 0, \dots, 3$ . Abbildung 2.23 veranschaulicht diese Situation. Die Vierecke  $Q_i$  sind, gemäß den zugehörigen Klassen mit „ $l = i$ “ bezeichnet.

Durch die Differenzierbarkeitsbedingungen über die Kanten  $E_0, E_1, E_2$  sowie  $E_3$  folgt sukzessive, dass im ungünstigsten Fall  $\mathcal{B}_z|_{Q_i} \neq 0$  ist für  $i = 1, 2, 3, 4$ . Die Menge  $\cup_{i=0}^4 Q_i$  ist dann im  $\text{star}^4(Q)$ ,  $Q \in \diamond$ , enthalten, aber in keinem  $\text{star}^3(\tilde{Q})$ ,  $\tilde{Q} \in \diamond$ . Wegen Behauptung 1 und der Reihenfolge der gewählten Vierecke ist  $\mathcal{B}_z|_{\diamond \setminus \text{star}^4_\diamond(Q)} \equiv 0$ .

Fall 2:  $z$  liegt auf einer Kante eines Vierecks  $Q \in \diamond$

Liege  $z$  nun auf den Kanten von  $\diamond$ . Wie im gefärbten Fall liegt bei der maximal möglichen Trägergröße der Punkt  $z$  auf der Kante eines Vierecks  $Q$ , welches im Kantentalgorithmus in der Iteration  $l = 4$  ausgewählt wird. In diesem Fall können Eckpunkte  $v$  von  $\diamond$  mit Gradienten ungleich Null maximal im  $\text{star}^3(Q)$  auftreten. Sei  $v_0$  ein solcher Eckpunkt auf dem Rand von  $\text{star}^3_\diamond(Q)$ . Folglich liegt in der hinsichtlich der Trägergröße ungünstigsten Situation das Viereck  $Q$  aus Fall 1 in der Menge  $\text{star}^4_\diamond(Q) \setminus \text{star}^3_\diamond(Q)$ , siehe Abbildung 2.23. Somit ist der Träger von  $\mathcal{B}_z$  im ungünstigsten Fall im  $\text{star}^8(Q)$ , aber in keinem  $\text{star}^7(\tilde{Q})$ ,  $\tilde{Q} \in \diamond$ , enthalten.  $\square$

Damit folgen in völlig analoger Weise zum gefärbten Fall Stabilität und optimale Approximationsordnung der Methode.

Wir untersuchen schließlich die Dimension der Interpolationsräume. Die Erfahrung zeigt, dass bei großen Quadrangulierungen im Schnitt ungefähr die Hälfte der Vierecke zweigeteilt ist und die andere Hälfte viergeteilt. Als einfaches Beispiel betrachte man eine Quadrangulierung, in welcher jeder Eckpunkt von geradem Grad ist. Vom Rand abgesehen befindet sich dann bei bestimmter Klassifizierung wie in Algorithmus 2.25 die Hälfte der Vierecke in der Klasse  $\mathcal{K}_0$  und die verbleibende Hälfte in der Klasse  $\mathcal{K}_4$ . Für beliebige Quadrangulierungen  $\diamond$  enthält  $\mathcal{K}_0$  im Schnitt zwar einen geringeren Anteil der Vierecke, allerdings enthält  $\mathcal{K}_0 \cup \mathcal{K}_1$  in den meisten Fällen wieder ungefähr die Hälfte der Vierecke.

Daher gehen wir im Folgenden davon aus, dass im Schnitt die Hälfte der Vierecke zweigeteilt wird, d. h. wir wählen

$$N \approx \frac{2+4}{2}n = 3n. \quad (2.21)$$

für die Anzahl  $N$  der Dreiecke in  $\diamond$ .

	<i>ursprüngliches Verfahren wie in [68]</i>	<i>Färbung nach Algorithmus 2.4</i>	<i>Interpolation unter Verwendung des Prioritätsprinzips</i>
<i>Lokalität</i>	Träger der Basis-splines stets in einem $\text{star}^9(Q)$ , $Q \in \diamond$ , enthalten	Träger der Basis-splines stets in einem $\text{star}^7(Q)$ , $Q \in \diamond$ , enthalten	Träger der Basis-splines stets in einem $\text{star}^8(Q)$ , $Q \in \diamond$ , enthalten
<i>Effizienz</i>	lineare Komplexität für Färbungs- und Interpolationsalgorithmus	lineare Komplexität für Färbungs- und Interpolationsalgorithmus	keine Färbung erforderlich, Interpolation in linearer Komplexität
<i>Anzahl der Mikrodreiecke</i>	$ \diamond  \approx 3\frac{1}{3} \cdot  \diamond $	$ \diamond  \approx 3 \cdot  \diamond $	$ \diamond  \approx 3 \cdot  \diamond $

Abbildung 2.24: Vergleich der Interpolationsmethoden

## 2.5 Vergleich der Interpolationsmethoden

Im folgenden Abschnitt vergleichen wir die dargestellten Interpolationsverfahren hinsichtlich Lokalitätseigenschaften, Dimension der Splineräume sowie algorithmischer Effizienz.

Wir betrachten dabei einerseits die auf dem Prioritätsprinzip basierende Methode, andererseits mit Algorithmus 2.4 und Algorithmus 2.9 gefärbte Quadrangulierungen. Tabelle 2.24 fasst die Ergebnisse des Vergleichs zusammen.

### Lokalität

Durch das Färbungsverfahren aus [68] werden Interpolanten erzeugt, deren zugehörige Basissplines maximal in einem  $\text{star}^9(Q)$ ,  $Q \in \diamond$ , enthalten sind. Für die in Abschnitt 2.2 entwickelte Färbungsmethode ergibt sich eine maximale Trägergröße eines  $\text{star}^7(Q)$ ,  $Q \in \diamond$ . Bei Verwendung des Prioritätsprinzips aus Abschnitt 2.4 erhält man einen  $\text{star}^8(Q)$ ,  $Q \in \diamond$ .

### Effizienz

Ein geringfügiger Effizienzvorteil des Prioritätsprinzips ist, dass die Vierecke der zugrunde liegenden Quadrangulierung nicht gefärbt werden. Auch die Konstruktion von Lagrange-Interpolationsmengen erfolgt bei dieser Methode in effizienter und einfacher Weise. Bei der Verwendung von gefärbten Quadrangulierungen sind gegebenenfalls auch längere Ketten von Vierecken algorithmisch zu erfassen. Hierbei muss geprüft werden, welche Vierecke Endpunkte sind bzw. ob eine geschlossene Kette gleichfarbiger Vierecke vorliegt. Da bei den weiter oben entwickelten Färbungsmethoden nur selten Ketten der Länge  $\geq 3$  und auch keine geschlossenen Ketten auftreten, ist die Verwaltung der Ketten und somit die Konstruktion der Interpolationsmengen im Allgemeinen mit weniger Rechenaufwand verbunden als bei der Verwendung von Färbungen wie in [68]. Weiterhin vereinfacht sich die Konstruktion der Interpolationsmengen, da der Fall für geschlossene Ketten nicht auftritt.

Was den Aufwand für die Färbung der Vierecke nach der ursprünglichen Methode angeht, so müssen zunächst alle Vierecke  $Q \in \diamond$  durchlaufen und auf die Anzahl gleichfarbiger Nachbarn überprüft werden. Nach jedem Farbwechsel eines Vierecks  $Q$  müssen wiederum die neuen gleichfarbigen Nachbarn von  $Q$  in dieser Weise geprüft werden. Bei der oben entwickelten Färbungsmethode hingegen müssen die Vierecke kegelweise in geeigneter Reihenfolge bearbeitet werden. Beide Verfahren besitzen lineare Komplexität und sind effizient implementierbar.

### Dimension resultierender Interpolationsräume

Aufgrund der unterschiedlichen Geometrie verschiedener Quadrangulierung kann die Dimension der resultierenden Splineräume in Abhängigkeit von der Anzahl der Vierecke in  $\diamond$ , wie in den Abschnitten 2.3 und 2.4 dargestellt, im Allgemeinen nicht bestimmt werden. Jedoch sind mit (2.19), (2.20) und (2.21) Abschätzungen für typische Quadrangulierungen gegeben.

Im Schnitt erhält man für die Methode auf ungefärbten Quadrangulierung und auch bei einer Färbung durch Algorithmus 2.4 daher Splineräume von geringerer Dimension als in [68].

# Kapitel 3

## 3D-Splineoberflächen auf Quadrangulierungen

In diesem Kapitel entwickeln wir die erste Methode zur Interpolation mit kubischen 3D-Splineoberflächen auf Quadrangulierungen. In der Literatur finden sich bislang fast ausschließlich Verfahren für stückweise lineare 3D-Oberflächen, siehe beispielsweise [30, 40, 42]. Das Konzept geglätteter 3D-Splines wurde erst kürzlich von Kohlmüller, Nürnberger und Zeilfelder entworfen, die in zwei Arbeiten [47, 48] die bislang einzige Interpolationsmethode für solche Splines entwickelten. Die dort enthaltenen Konzepte dienen als Grundlage für das in diesem Kapitel entwickelte Verfahren.

Gegeben sei eine feine Triangulierung  $\Delta_f$  im Raum. Wie in [47] interpolieren wir  $\Delta_f$  zunächst durch eine stetige kubische Splineoberfläche, definiert auf einer (groben) 3D-Triangulierung. Anschließend fassen wir Paare von Dreiecken dieser Triangulierung zu Vierecken zusammen, wobei zusammengefasste Vierecke im Allgemeinen nicht in einer Ebene liegen. Wir entwickeln hierfür einen effizienten Algorithmus, so dass fast alle Dreiecke zusammengefasst werden können. Dabei ist insbesondere von Bedeutung, dass die Dreiecke in einer bestimmten Reihenfolge betrachtet werden. Auf diese Weise wird erreicht, dass nur in speziellen Fällen innere Dreiecke einzeln bleiben. Wir geben eine ausführliche Darstellung dieser Fälle. Die auf diese Weise konstruierte Zerlegung nennen wir *quadrangulierte 3D-Triangulierung*  $\diamond$ .

Basierend auf Färbungs- und Prioritätsprinzipien ähnlich wie in Kapitel 2 modifizieren wir den stetigen Interpolanten, um glatte Splineoberflächen zu erhalten. Dabei wird der Interpolant zuerst über den Eckpunkten, dann über den Kanten von  $\diamond$  geglättet. Für die geglätteten Oberflächen erhält man Differenzierbarkeit auf den Eckpunkten und Mittelpunkten der Kanten von  $\diamond$ .

Wir zeigen die Lokalität der Methode. Entspricht die 3D-Triangulierung einer ebenen Triangulierung, so ist es uns möglich, Ergebnisse für die Approximationsordnung

zu geben. Für diesen Fall zeigen wir, dass die Methode optimale Approximationsordnung besitzt.

### 3.1 Grundlagen

Wir verwenden der besseren Lesbarkeit wegen für 3D-Symbole fettgedruckte Buchstaben, z. B.  $\mathbf{v} := (x, y, z) \in \mathbb{R}^3$ . Zunächst stellen wir die Bézier-Bernstein-Darstellung für Polynome in drei Komponenten dar. Bezüglich eines nichtdegenerierten Dreiecks  $T := [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] \subset \mathbb{R}^3$  im Raum existiert für jeden Punkt in der durch  $T$  gegebenen Ebene  $(\phi_1, \phi_2, \phi_3)$  eine eindeutige Darstellung, genannt *baryzentrischer Koordinatenvektor*  $\phi = (\phi_1, \phi_2, \phi_3) : (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) \rightarrow \mathbb{R}^3$ , mit folgender Eigenschaft:

$$\mathbf{z} = \sum_{i=1}^3 \phi_i(\mathbf{z}) \mathbf{v}_i, \quad \text{wobei} \quad \phi_1(\mathbf{z}) + \phi_2(\mathbf{z}) + \phi_3(\mathbf{z}) = 1 \quad (3.1)$$

Wie bei den baryzentrischen Koordinaten von Dreiecken der Ebene erfüllt auch die Darstellung für Dreiecke im Raum die Interpolationseigenschaft  $\phi_i(\mathbf{v}_j) = \delta_{ij}$  für alle  $i, j = 1, 2, 3$ . Ferner ist genau dann  $\mathbf{z} \in T$ , wenn  $\phi_i \in [0, 1]$  ist für alle  $i = 1, 2, 3$ .

Unter Verwendung von (3.1) sind die *polynomialen 3D-Oberflächen über  $T$*  wie folgt definiert: Es handelt sich um Abbildungen  $\mathbf{p} : T \rightarrow \mathbb{R}^3$ , welche in jeder Komponente  $p_\tau$ ,  $\tau = 1, 2, 3$ , ein bivariates Polynom von totalem Grad  $q$  sind. Die drei Komponenten von  $\mathbf{p}$  werden in Bézier-Bernstein-Schreibweise dargestellt. Die *Bézier-Bernstein-Basispolynome*  $B_{i,j,k}^{q,T} : T \rightarrow \mathbb{R}$ ,  $i + j + k = q$ , über  $T \subset \mathbb{R}^3$  sind hierbei entsprechend dem bivariaten Fall definiert durch

$$B_{i,j,k}^{q,T}(\mathbf{z}) = \frac{q!}{i!j!k!} \phi_1(\mathbf{z})^i \phi_2(\mathbf{z})^j \phi_3(\mathbf{z})^k, \quad \mathbf{z} \in T, \quad (3.2)$$

und die *Bézier-Bernstein-Darstellung von  $\mathbf{p}$*  mit zugehörigen *Bézier-Bernstein-Koeffizientenvektoren*  $\mathbf{c}_{i,j,k} = [(c_1)_{i,j,k}, (c_2)_{i,j,k}, (c_3)_{i,j,k}] \in \mathbb{R}^3$ ,  $i + j + k = q$ , lautet

$$\mathbf{p}(\mathbf{z}) = \sum_{i+j+k=q} B_{i,j,k}^{q,T}(\mathbf{z}) \mathbf{c}_{i,j,k}, \quad \mathbf{z} \in T. \quad (3.3)$$

Dabei entspricht die Menge der  $q$ -gradigen bivariaten Polynome über  $T$  einer Teilmenge der polynomialen 3D-Oberflächen: Jeder Bézier-Bernstein-Koeffizient  $\mathbf{c}_{i,j,k}$  liegt für den Fall, dass  $\mathbf{p}$  einem bivariaten Polynom entspricht, auf der Normalen zu  $T$ , die durch den Punkt  $\frac{i}{q} \mathbf{v}_1 + \frac{j}{q} \mathbf{v}_2 + \frac{k}{q} \mathbf{v}_3$  geht.  $\mathbf{p}(\mathbf{z})$  lässt sich in diesem Fall darstellen als

$$\begin{aligned} \mathbf{p}(\mathbf{z}) &= \sum_{i+j+k=q} B_{i,j,k}^{q,T}(\mathbf{z}) (\mathbf{b}_{i,j,k} + \mathbf{c}_{i,j,k}^\perp) \\ &= \sum_{i+j+k=q} B_{i,j,k}^{q,T}(\mathbf{z}) \mathbf{b}_{i,j,k} + \sum_{i+j+k=q} B_{i,j,k}^{q,T}(\mathbf{z}) \mathbf{c}_{i,j,k}^\perp, \quad \mathbf{z} \in T, \end{aligned}$$

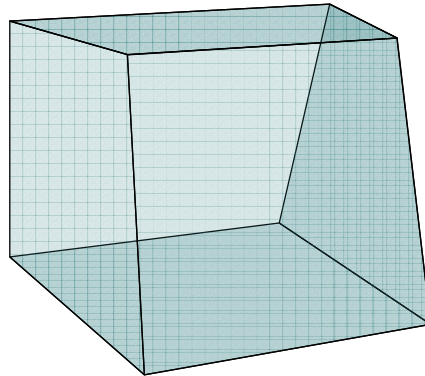


Abbildung 3.1: 3D-Quadrangulierung, für welche Algorithmus 2.4 nicht direkt anwendbar ist.

wobei  $\mathbf{b}_{i,j,k}$ ,  $i+j+k$ , die Vektoren zu den jeweiligen Bézier-Bernstein-Punkten auf  $T$  und  $\mathbf{c}_{i,j,k}^\perp$  die jeweiligen, orthogonal zu  $T$  liegenden Differenzvektoren bezeichnen. Das entsprechende bivariate Polynom  $p : T \rightarrow \mathbb{R}$  hat dann die Bézier-Bernstein-Koeffizienten  $c_{ijk} = \|\mathbf{c}_{ijk}^\perp\|_2$ .

Das *Bézier-Bernstein-Netz* einer polynomialen 3D-Oberfläche ist direkt durch die Koeffizienten  $\mathbf{c}_{i,j,k}$  gegeben, wobei auch hier Netz- und Polynomoberfläche an den Eckpunkten von  $T$  übereinstimmen.

Wir behandeln nun den Begriff der 3D-Splineoberflächen. Eine *Triangulierung*  $\Delta$  definiert als eine Menge  $\{T_1, \dots, T_N\}$  von Dreiecken, falls die analogen Bedingungen wie im zweidimensionalen Fall erfüllt sind. Ebenso ist eine Quadrangulierung  $\diamond$  von  $\Omega$  analog zur Quadrangulierung im  $\mathbb{R}^2$  definiert.

Im Gegensatz zur Zerlegung in der Ebene ist das zugehörige Gebiet  $\Omega$  zwar zusammenhängend, jedoch nicht notwendigerweise einfach zusammenhängend. Insbesondere bemerken wir, dass Lemma 2.2 nicht analog für Quadrangulierungen im Raum gilt und der in Abschnitt 2.2 entwickelte Färbungsalgorithmus nicht direkt übertragbar ist. Selbst bei einfach zusammenhängenden Quadrangulierungen ist der Färbungsalgorithmus nicht analog anwendbar: Man betrachte beispielsweise die sehr einfache, in Abbildung 3.1 dargestellte Quadrangulierung  $\diamond$ , bestehend aus sechs Vierecken  $Q_1, \dots, Q_6$ . Obwohl bei einem sukzessiven Anfügen von Kegeln alle Teilquadrangulierungen und auch  $\diamond$  selbst einfache zusammenhängende Punktmengen bilden, liegen bei der Färbung des letzten Vierecks stets vier bereits gefärbte Nachbarn vor.

Offensichtlich lässt sich jede Triangulierung  $\Delta$  im Raum als dreidimensionale polynomiale Oberfläche vom Grad eins interpretieren: Auf einer geeigneten Triangulierung  $\tilde{\Delta}$ , im einfachsten Fall  $\Delta$  selbst, wählt man die Koeffizienten aus (3.3), sodass

sie mit den Eckpunkten von  $\Delta$  übereinstimmen. Diese Interpretation wird wie folgt auf höhergradige dreidimensionale polynomiale Oberflächen übertragen: Eine Abbildung

$$\mathbf{s} = (s_1, s_2, s_3) : \Omega \rightarrow \mathbb{R}^3 \quad (3.4)$$

heißt *Spline vom Grad  $q$  in drei Komponenten* bezüglich einer 3D-Triangulierung  $\Delta$  einer Oberfläche  $\Omega$ , falls  $s_\tau|_T \in \mathcal{P}_q^2$  ist für alle  $\tau = 1, 2, 3$  und alle  $T \in \Delta$ . Die zugehörige *3D-Splineoberfläche* ist gegeben durch die folgende Menge von Punkten:

$$\mathbf{s}(\Omega) = \bigcup_{T \in \Delta} \left( \bigcup_{\mathbf{z} \in T} \mathbf{s}|_T(\mathbf{z}) \right) \quad (3.5)$$

Die Menge  $\mathcal{P}_q(\Delta)$  der *Bézier-Bernstein-Punkte* ist, analog zum bivariaten Fall, definiert durch

$$\mathcal{P}_q(\Delta) := \bigcup_{[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] \in \Delta} \left( \bigcup_{i+j+k=q} i \mathbf{v}_1 + j \mathbf{v}_2 + k \mathbf{v}_3 \right). \quad (3.6)$$

Das *Bézier-Bernstein-Netz*  $\mathcal{N}_s(\Delta)$  ergibt sich als Vereinigung aller Bézier-Bernstein-Koeffizienten  $\mathbf{c}_{ijk}^{[T]}$ :

$$\mathcal{N}_s(\Delta) := \bigcup_{T \in \Delta} \left( \bigcup_{i+j+k=q} \mathbf{c}_{ijk}^{[T]} \right) \quad (3.7)$$

Lokalität einer Methode ist wie folgt definiert: Gegeben ein Interpolant  $\mathbf{s} : \Omega_\Delta \rightarrow \mathbb{R}^3$  einer feinen Oberfläche  $\Omega_f$  und ein Interpolant  $\tilde{\mathbf{s}} : \Omega_\Delta \rightarrow \mathbb{R}^3$ , welcher sich durch die Modifikation eines einzelnen Interpolationswerts  $\mathbf{f}_z$ ,  $\mathbf{z} \in \Omega_\Delta$ , ergibt. Dann heißt das entsprechende Interpolationsverfahren *lokal*, falls ein von der Lage von  $\mathbf{z}$  und von  $|\Delta|$  unabhängiges  $l \in \mathbb{N}_0$  existiert, sodass für ein bestimmtes Dreieck  $T \in \Delta$  die folgende Bedingung erfüllt ist:

$$\mathbf{s}(\tilde{\mathbf{z}}) = \tilde{\mathbf{s}}(\tilde{\mathbf{z}}), \quad \tilde{\mathbf{z}} \in \Omega_\Delta \setminus \text{star}^l(T) \quad (3.8)$$

Der Differenzierbarkeitsbegriff lässt sich nicht direkt auf 3D-Splines übertragen, siehe [36]. Es ist nur möglich, stetige Differenzierbarkeit für jede Komponente  $s_\tau$ ,  $\tau = 1, 2, 3$  von  $\mathbf{s}$  zu fordern, falls die betrachteten Dreiecke in einer Ebene liegen. Hierbei ist von Bedeutung, dass eine polynomiale 3D-Oberfläche  $G(\mathbf{s}|_T)$  unabhängig von Lage und Geometrie des entsprechenden Dreiecks  $T \in \Delta$  ist. Somit ist  $\mathbf{s}$  über eine gemeinsame Kante zweier Dreiecke differenzierbar, falls zwei benachbarte ebene Dreiecke existieren, sodass Theorem 1.4 für jede Komponente  $s_\tau$ ,  $\tau = 1, 2, 3$ , über den Dreiecken der Ebene erfüllt ist.



Wir fordern die *Regularität* von  $\mathbf{s}$ . In diesem Fall existiert in jedem Punkt  $\mathbf{z} \in \Omega$  ein Einheitsnormalenvektor, d. h.

$$\mathbf{n}(\mathbf{s})(\mathbf{z}) = \frac{\mathbf{s}_x(\mathbf{z}) \times \mathbf{s}_y(\mathbf{z})}{\|\mathbf{s}_x(\mathbf{z}) \times \mathbf{s}_y(\mathbf{z})\|_2} \quad (3.9)$$

ist wohldefiniert für alle  $\mathbf{z} \in \Omega$ .

Es ist eine bekannte Tatsache (vgl. [36], S. 120ff.), dass die Forderung stetiger Differenzierbarkeit im Allgemeinen zu streng und deshalb in der Anwendung nicht sinnvoll ist. Daher gibt es verschiedene Ansätze für weniger strenge Glatttheitsbedingungen [34, 74]. Die entsprechenden Bedingungen sind dabei unabhängig von Lage und Geometrie der Grunddreiecke. Wir verwenden im Folgenden die gleiche Glatttheitsbedingung, die in der Methode von Kohlmüller, Nürnberger und Zeilfelder [48] verwendet wird. Bei der Glättung über einem Eckpunkt  $\mathbf{v} \in \mathcal{V}^{[\Delta]}$  projiziert man, grob gesprochen, alle Koeffizienten des  $C^1$ -Rings um  $\mathbf{v}$  in die gemittelte Tangentialebene der Splineoberfläche  $\mathbf{s}$  und erhält dadurch Differenzierbarkeit des geglätteten Splines in  $\mathbf{v}$ . Bei der Glättung über einer Kante  $[\mathbf{u}, \mathbf{v}]$ , die zwei Dreiecken  $T_1$  und  $T_2$  aus  $\Delta$  angehört, geht man, grob gesprochen, wie folgt vor: Man mittelt die Tangentialebenen in  $\mathbf{m} = 1/2(\mathbf{u} + \mathbf{v})$  und modifiziert die beiden Koeffizienten  $\mathbf{c}_{111}^{[T_l]}$ ,  $l = 1, 2$ , sodass die zu  $[\mathbf{u}, \mathbf{v}]$  senkrechten Ableitungen von  $\mathbf{s}|_{T_l}$  in  $\mathbf{m}$  übereinstimmen. Auf diese Weise erhält man einen geglätteten Spline, der an den Punkten  $\mathbf{u}$ ,  $\mathbf{v}$  und  $\mathbf{m}$  differenzierbar ist. Darüber hinaus weichen die Normalenvektoren der Polynomstücke auf allen übrigen Punkten der Kante nur wenig voneinander ab, siehe Theorem 3 in [48].

Wir geben nun eine detaillierte Beschreibung dieser Modifikation von Bézier-Bernstein-Koeffizienten. Zunächst stellen wir die Glättung über einem Eckpunkt  $\mathbf{v} \in \mathcal{V}^{[\Delta]}$  dar:

### Algorithmus 3.1

Sei  $\mathbf{s} : \Omega \rightarrow \mathbb{R}^3$  ein stetiger Spline in drei Komponenten  $(s_1, s_2, s_3)$  bezüglich der 3D-Triangulierung  $\Delta$ , ferner sei  $\mathbf{v} \in \mathcal{V}^{[\Delta]}$  ein Eckpunkt, welcher mindestens zwei Dreiecken aus  $\Delta$  angehört. Die Dreiecke, welche  $\mathbf{v}$  enthalten, seien mit  $T_l \in \Delta$ ,  $l = 1, \dots, n$ , und die zugehörigen Bézier-Bernstein-Koeffizienten mit  $\mathbf{c}_{ijk}^{[T_l]}$  bezeichnet. Die Nummerierung der Eckpunkte sei für alle  $l = 1, \dots, n$ , sodass  $\mathbf{c}_{300}^{[T_l]}$  zu  $\mathbf{v}$  gehöre. Bezeichne  $\mathbf{n}_l(\mathbf{v})$  den Normalenvektor der durch die Koeffizienten  $\mathbf{c}_{300}^{[T_l]}$ ,  $\mathbf{c}_{210}^{[T_l]}$  und  $\mathbf{c}_{201}^{[T_l]}$  aufgespannten Ebene für alle  $l = 1, \dots, n$ . Der Normalenvektor der gemittelten Ebene berechnet sich dann wie folgt:

$$\mathbf{n}(\mathbf{v}) = \frac{\sum_{l=1}^n \mathbf{n}_l(\mathbf{v})}{\|\sum_{l=1}^n \mathbf{n}_l(\mathbf{v})\|_2} \quad (3.10)$$

Man projiziere alle Koeffizienten mit Abstand eins zu  $\mathbf{v}$  in die gemittelte Tangentialebene, verschiebe sie also in Richtung  $\mathbf{n}(\mathbf{v})$ , bis sie in dieser Ebene liegen. Eine solche Vorgehensweise nennen wir *Glättung von  $\mathbf{s}$  auf  $\mathbf{v}$*  oder kurz *Glättung von  $\mathbf{v}$* .

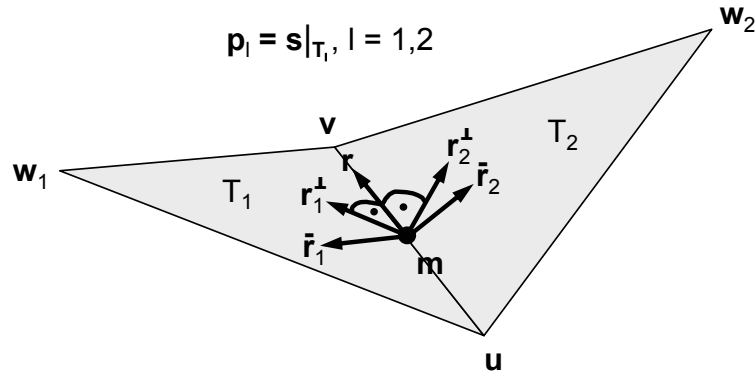


Abbildung 3.2: Notation bei der Glättung einer gemeinsamen Kante

Mit dieser Modifikation erhalten wir das folgende Ergebnis:

**Lemma 3.2**

Ein wie in Algorithmus 3.1 über einem Eckpunkt  $v$  von  $\Delta$  geglätteter Spline  $s$  ist differenzierbar in  $v$ .

**Beweis:** Da die Bézier-Bernstein-Koeffizienten  $c_l$ ,  $l = 1, \dots, n + 1$ , von  $s$  der  $C^1$ -Scheibe um  $v$  in einer Ebene liegen, stimmt diese Ebene mit der Tangentialebene jedes Polynomstücks  $s|_{T_l}$  der zu  $v$  gehörenden Dreiecke  $T_l$ ,  $l = 1, \dots, n$ , in  $v$  überein, siehe Beweis zu Theorem 3 in [48]. Betrachtet man die ebene Zelle von  $n$  Dreiecken, welche diese Koeffizienten als Eckpunkte besitzen, so ist jede Komponente  $s_\tau$ ,  $\tau = 1, 2, 3$ , über dieser Zelle differenzierbar in  $c_v$ .  $\square$

Wir beschreiben nun die Glättung von  $s$  über einer Kante  $[u, v]$  der Triangulierung:

**Algorithmus 3.3**

Sei  $s : \Omega \rightarrow \mathbb{R}^3$  ein stetiger Spline in drei Komponenten  $(s_1, s_2, s_3)$  bezüglich der 3D-Triangulierung  $\Delta$ , ferner seien  $T_l := [u, v, w_l]$ ,  $l = 1, 2$ , zwei benachbarte Dreiecke aus  $\Delta$  mit zugehörigen Bézier-Bernstein-Koeffizienten  $c_{ijk}^{[T_l]}$ ,  $l = 1, 2$ .

Wir bezeichnen als *Glättung von  $s$  über  $[u, v]$* , oder kurz Glättung von  $[u, v]$ , folgende Modifikation einer Teilmenge dieser Koeffizienten:

Zunächst berechne man die Tangentialebenen der Polynomstücke  $p_1 := s|_{T_1}$  und  $p_2 := s|_{T_2}$  im Kantenmittelpunkt  $m := \frac{u+v}{2}$ . Seien mit  $r$  der Einheitsvektor  $r := \frac{v-u}{\|v-u\|_2}$  in Richtung gemeinsamer Kante  $[u, v]$  und mit  $r_l^\perp$ ,  $l = 1, 2$ , die dazu orthogonalen Einheitsvektoren in den Dreiecken  $T_l$  bezeichnet. Dann liefert der Algorithmus von de Casteljau die Richtungsableitung von  $s$  im Punkt  $m$  in Richtung  $r$ :

$$\mathbf{s}_r(\mathbf{m}) = \frac{3}{4 \|\mathbf{v} - \mathbf{u}\|_2} \left( \mathbf{c}_{300}^{[T_1]} + \mathbf{c}_{210}^{[T_1]} - \mathbf{c}_{120}^{[T_1]} - \mathbf{c}_{030}^{[T_1]} \right) \quad (3.11)$$

Man beachte, dass  $\mathbf{c}_{ij0}^{[T_1]} = \mathbf{c}_{ij0}^{[T_2]}$  ist für alle  $i + j = 3$  und somit  $\mathbf{p}_1|_{[\mathbf{u},\mathbf{v}]} = \mathbf{p}_2|_{[\mathbf{u},\mathbf{v}]}$  gilt.

Die orthogonalen Vektoren  $\mathbf{r}_l^\perp$ ,  $l = 1, 2$ , sind aufgrund der linearen Abhängigkeit zu den Kantenvektoren der Dreiecke  $T_l$  durch Linearfaktoren  $\alpha_l$  und  $\beta_l$  wie folgt darstellbar:

$$\mathbf{r}_l^\perp = \frac{(\mathbf{w}_l - \mathbf{v}) - \beta_l(\mathbf{v} - \mathbf{u})}{\alpha_l}, \quad l = 1, 2$$

Als Nächstes werden die orthogonalen Ableitungen von  $\mathbf{p}_1$  und  $\mathbf{p}_2$  in  $\mathbf{m}$  benötigt, also diejenigen Ableitungsvektoren, welche orthogonal zum Vektor  $\mathbf{s}_r(\mathbf{m})$  stehen. Diese Ableitungen seien mit  $(\mathbf{p}_l)_{\bar{\mathbf{r}}_l}(\mathbf{m})$ ,  $l = 1, 2$ , bezeichnet. Die zugehörigen Richtungsvektoren  $\bar{\mathbf{r}}_l$ ,  $l = 1, 2$ , liegen in den Dreiecken  $T_l$ , sind im Allgemeinen nicht orthogonal zu  $\mathbf{r}$  und lassen sich in eindeutiger Weise darstellen als

$$\bar{\mathbf{r}}_l = \bar{\alpha}_l \mathbf{r} + \bar{\beta}_l \mathbf{r}_l^\perp, \quad l = 1, 2. \quad (3.12)$$

Die beiden Ableitungsvektoren sind somit mit denselben Koeffizienten wie folgt darstellbar:

$$(\mathbf{p}_l)_{\bar{\mathbf{r}}_l}(\mathbf{m}) = \bar{\alpha}_l \mathbf{s}_r(\mathbf{m}) + \bar{\beta}_l (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}), \quad l = 1, 2 \quad (3.13)$$

Man setze nun

$$\bar{\alpha}_l := -\frac{\langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \rangle}{\|\mathbf{s}_r(\mathbf{m})\|_2^2}, \quad \bar{\beta}_l := 1, \quad l = 1, 2. \quad (3.14)$$

Durch diese Wahl von  $\bar{\alpha}_l$  und  $\bar{\beta}_l$ ,  $l = 1, 2$ , erhält man die beiden zu  $\mathbf{s}_r(\mathbf{m})$  orthogonalen Ableitungsvektoren, denn für  $l = 1, 2$  folgt

$$\begin{aligned} \langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_l)_{\bar{\mathbf{r}}_l}(\mathbf{m}) \rangle &= \langle \mathbf{s}_r(\mathbf{m}), \bar{\alpha}_l \mathbf{s}_r(\mathbf{m}) + \bar{\beta}_l (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \rangle \\ &= \left\langle \mathbf{s}_r(\mathbf{m}), -\frac{\langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \rangle}{\|\mathbf{s}_r(\mathbf{m})\|_2^2} \mathbf{s}_r(\mathbf{m}) + (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \right\rangle \\ &= \left\langle \mathbf{s}_r(\mathbf{m}), -\frac{\langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \rangle}{\|\mathbf{s}_r(\mathbf{m})\|_2^2} \mathbf{s}_r(\mathbf{m}) \right\rangle + \langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \rangle \\ &= -\frac{\langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \rangle}{\|\mathbf{s}_r(\mathbf{m})\|_2^2} \langle \mathbf{s}_r(\mathbf{m}), \mathbf{s}_r(\mathbf{m}) \rangle + \langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \rangle \\ &= -\frac{\langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \rangle}{\|\mathbf{s}_r(\mathbf{m})\|_2^2} \|\mathbf{s}_r(\mathbf{m})\|_2^2 + \langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \rangle \\ &= 0. \end{aligned} \quad (3.15)$$

Man bestimme nun die beiden Normalenvektoren in  $\mathbf{m}$  wie folgt:

$$\mathbf{n}_l(\mathbf{m}) := \frac{(\mathbf{p}_l)_{\bar{\mathbf{r}}_l}(\mathbf{m}) \times \mathbf{s}_r(\mathbf{m})}{\|\mathbf{s}_r(\mathbf{m})\|_2^2}, \quad l = 1, 2 \quad (3.16)$$

Aufgrund der Definition von  $\mathbf{n}_l(\mathbf{m})$ ,  $l = 1, 2$ , und (3.15) bilden die Vektoren

$$\{ \mathbf{n}_l(\mathbf{m}), (\mathbf{p}_l)_{\bar{\mathbf{r}}_l}(\mathbf{m}), \mathbf{s}_r(\mathbf{m}) \} \quad (3.17)$$

für  $l = 1, 2$  Orthogonalsysteme.

An dieser Stelle unterscheiden wir zwei Fälle: In Fall 1, genannt *Glättung von  $\mathbf{s}$  über  $[\mathbf{u}, \mathbf{v}]$* , ermittelt man den gemittelten Normalenvektor  $\mathbf{n}(\mathbf{m})$  und projiziert die Bézier-Bernstein-Koeffizienten  $\mathbf{c}_{111}^{[T_l]}$ ,  $l = 1, 2$ , beider Dreiecke in die gemittelte Ebene. Dagegen lässt man in Fall 2, genannt *Glättung von  $\mathbf{s}$  über  $T_2$  bezüglich des Nachbardreiecks  $T_1$* , das Polynomstück  $\mathbf{p}_1$  unverändert und projiziert lediglich den Koeffizienten  $\mathbf{c}_{111}^{[T_2]}$  in die zu  $\mathbf{p}_1$  gehörende Ebene. Die Kante  $[\mathbf{u}, \mathbf{v}]$  wird nach Beendigung der Projektion in beiden Fällen als *geglättet* bezeichnet.

Fall 1: Glättung von  $[u, v]$

Mit Hilfe der gemittelten Tangentialebene lässt sich die modifizierte orthogonale Ableitung  $\mathbf{s}_{\bar{\mathbf{r}}^*}(\mathbf{m})$  bezüglich des neuen Richtungsvektors  $\bar{\mathbf{r}}^*$  in  $\mathbf{m}$  wie folgt berechnen:

$$\mathbf{s}_{\bar{\mathbf{r}}^*}(\mathbf{m}) := \mathbf{n}(\mathbf{m}) \times \mathbf{s}_r(\mathbf{m}) \quad \text{mit} \quad \mathbf{n}(\mathbf{m}) := \frac{1}{2} \sum_{l=1}^2 \mathbf{n}_l(\mathbf{m}) \quad (3.18)$$

Es sei bemerkt, dass der dadurch bestimmte Vektor  $\mathbf{s}_{\bar{\mathbf{r}}^*}(\mathbf{m})$  ungleich Null ist, denn es gilt:

- (i)  $\langle \mathbf{n}(\mathbf{m}), \mathbf{s}_r(\mathbf{m}) \rangle = \frac{1}{2} \sum_{l=1}^2 2 \langle \mathbf{n}_l(\mathbf{m}), \mathbf{s}_r(\mathbf{m}) \rangle = 0$
- (ii)  $\mathbf{s}_r(\mathbf{m}) \neq \mathbf{0}$

Aufgrund der Zusammenhänge (3.12) bis (3.14) folgt für die (modifizierten) Ableitungen der Polynomstücke in Richtung der orthogonalen Richtungsableitungen

$$(\mathbf{p}_l^*)_{\mathbf{r}_l^\perp}(\mathbf{m}) = (\mathbf{n}(\mathbf{m}) \times \mathbf{s}_r(\mathbf{m})) + \frac{\langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_l)_{\mathbf{r}_l^\perp}(\mathbf{m}) \rangle}{\|\mathbf{s}_r(\mathbf{m})\|_2^2} \cdot \mathbf{s}_r(\mathbf{m}), \quad l = 1, 2.$$

Da die auf  $[\mathbf{u}, \mathbf{v}]$  liegenden Bézier-Bernstein-Koeffizienten der Polynomstücke übereinstimmen, sind die Koeffizienten  $\mathbf{c}_{111}^{[T_l]}$ ,  $l = 1, 2$ , aus den orthogonalen Ableitungen  $(\mathbf{p}_l^*)_{\mathbf{r}_l^\perp}(\mathbf{m})$ ,  $l = 1, 2$ , bestimmbar. Diese Tatsache ergibt sich durch Anwendung von Satz 1.8 auf jede Komponente  $\mathbf{s}_\tau$ ,  $\tau = 1, 2, 3$ .

### Fall 2: Glättung von $T_2$ bezüglich $T_1$

Ohne Beschränkung der Allgemeinheit sei  $\mathbf{p}_2$  das zu verändernde und  $\mathbf{p}_1$  das zu belassende Polynomstück. Analog zur Berechnung (3.18) bestimme man für  $\mathbf{p}_2$  die neue orthogonale Ableitung  $(\mathbf{p}_2)_{\bar{\mathbf{r}}_2^*}(\mathbf{m})$  bezüglich des neuen Richtungsvektors  $\bar{\mathbf{r}}_2^* := \bar{\mathbf{r}}_1$  wie folgt:

$$(\mathbf{p}_2)_{\bar{\mathbf{r}}_2^*}(\mathbf{m}) := \mathbf{n}_1(\mathbf{m}) \times \mathbf{s}_r(\mathbf{m}) \quad (3.19)$$

Für die (modifizierten) Ableitungen von  $\mathbf{p}_2$  in Richtung der orthogonalen Richtungsableitung ergibt sich mit den Zusammenhängen (3.12) bis (3.14)

$$(\mathbf{p}_2^*)_{\mathbf{r}_t^\perp}(\mathbf{m}) = (\mathbf{n}_1(\mathbf{m}) \times \mathbf{s}_r(\mathbf{m})) + \frac{\langle \mathbf{s}_r(\mathbf{m}), (\mathbf{p}_2)_{\mathbf{r}_t^\perp}(\mathbf{m}) \rangle}{\|\mathbf{s}_r(\mathbf{m})\|_2^2} \cdot \mathbf{s}_r(\mathbf{m}).$$

Wieder lässt sich der modifizierte Koeffizient  $\mathbf{c}_{111}^{[T_2]}$  aus  $(\mathbf{p}_2^*)_{\mathbf{r}_t^\perp}(\mathbf{m})$  durch eine Anwendung von Satz 1.8 auf die Komponenten  $\tau = 1, 2, 3$  von  $\mathbf{s}$  bestimmen.

Auf diese Weise erhält man Differenzierbarkeit über dem Mittelpunkt der gemeinsamen Kante:

#### Lemma 3.4

Ein wie in Algorithmus 3.3 über einer Kante  $[\mathbf{u}, \mathbf{v}]$  von  $\Delta$  geglätteter Spline  $\mathbf{s}$  ist differenzierbar in  $1/2(\mathbf{u} + \mathbf{v})$ .

**Beweis:** Die Polynomstücke  $\mathbf{s}|_{T_1}$  und  $\mathbf{s}|_{T_2}$  auf den zu  $[\mathbf{u}, \mathbf{v}]$  gehörenden Dreiecken  $T_1$  und  $T_2$  stimmen auf der Kante überein, besitzen im Mittelpunkt  $\mathbf{m}$  also insbesondere dieselbe Ableitung in Richtung  $[\mathbf{u}, \mathbf{v}]$ . Weiterhin stimmen sie in den modifizierten orthogonalen Ableitungen im Kantenmittelpunkt überein. Betrachtet man zu  $T_1$  und  $T_2$  ähnliche Dreiecke der Ebene mit gemeinsamer Kante  $[u, v]$ , so ist daher jede Komponente  $s_\tau$ ,  $\tau = 1, 2, 3$ , differenzierbar in  $1/2(u + v)$ .  $\square$

## 3.2 Berechnung des stetigen Interpolanten

Es sei ein zu interpolierendes 3D-Objekt in Form einer feinen Oberfläche gegeben. In der Praxis verwendet man hierfür meist eine feine Triangulierung  $\Omega_f$ . Liegt die Oberfläche jedoch in Form einer Punktmenge vor, so lässt sich daraus in effizienter Weise eine feine Triangulierung erzeugen, man vergleiche beispielsweise mit [14, 31].

Weiterhin liege eine grobe Triangulierung  $\Delta$  vor, welche mit  $\Omega_f$  an den Eckpunkten übereinstimmt. Zur Konstruktion einer solchen Triangulierung existieren geeignete Methoden, genannt *mesh simplification*, siehe [15, 40].

Im ersten Schritt des Interpolationsalgorithmus berechnen wir einen stetigen kubischen Spline  $\tilde{\mathbf{s}}$ , welcher  $\Omega_f$  an allen Bézier-Bernstein-Punkten von  $\Delta$  interpoliert. Sei also  $\tilde{\mathbf{s}} : \Omega \rightarrow \mathbb{R}^3$  ein Spline in drei Komponenten  $(\tilde{s}_1, \tilde{s}_2, \tilde{s}_3)$ , welcher die Interpolationsbedingungen

$$\tilde{\mathbf{s}}(\mathbf{z}) = \mathbf{f}(\mathbf{z}), \quad \mathbf{z} \in \mathcal{P}_3(\Delta), \quad (3.20)$$

erfüllt, wobei mit  $\mathbf{f}(\mathbf{z}) = [f_1(\mathbf{z}), f_2(\mathbf{z}), f_3(\mathbf{z})]$  diejenigen Daten von  $\Omega_f$  bezeichnet sind, die den jeweiligen Interpolationspunkten  $\mathbf{z}$  bezüglich der euklidischen Norm am nächsten liegen.

Da auf jedem Punkt  $\mathbf{z} \in \Omega$ , welcher mehreren Dreiecken  $T \in \Delta$  angehört, für jedes zugehörige Polynomstück  $\mathbf{s}|_T$  derselbe Interpolationswert gewählt wird, ist der resultierende Spline  $\tilde{\mathbf{s}}$  stetig auf  $\Omega$ .

In computertechnischen Implementierungen werden die Werte  $\mathbf{f}(\mathbf{z})$ ,  $\mathbf{z} \in \mathcal{P}_3(\Delta)$ , ermittelt, indem die Oberfläche  $\Omega_f$  durch eine endliche große Menge von Punkten  $\mathcal{P} \subset \Omega_f$  diskretisiert wird. Ist beispielsweise  $\Omega_f$  eine feine Dreiecksfläche  $\Delta_f$ , so wählt man  $\mathcal{P} := \mathcal{P}_\lambda(\Delta_f)$  als die Menge der Bézier-Bernstein-Punkte auf  $\Delta_f$  von einer gewünschten Punktdichte  $\lambda \in \mathbb{N}$ . Aus Gründen der Effizienz ist es dabei nötig, die Punktmenge  $\mathcal{P}$  in lokale Teilmengen zu zerlegen. Man spricht hierbei vom Vorgang des *Clusterns* der Menge. In dieser Weise müssen für jeden Bézier-Bernstein-Punkt  $\mathbf{z}$  nur noch Abstände zu wesentlich weniger Daten aus  $\mathcal{P}$  berechnet und verglichen werden. Eine effiziente Umsetzung dieses Problems implementierte beispielsweise Grüntjes, siehe [41], Seite 75ff.

Auf jedem Dreieck  $T \in \Delta$  und in jeder Komponente  $\tau = 1, 2, 3$  von  $\tilde{\mathbf{s}}$  führt nun der Zusammenhang

$$f_\tau(\mathbf{z}_{ijk}^{[T]}) = \sum_{i'+j'+k'=3} (\tilde{c}_\tau)_{i'j'k'}^{[T]} B_{i'j'k'}^3(\mathbf{z}_{ijk}^{[T]}), \quad i+j+k=3, \quad (3.21)$$

zwischen den Interpolationswerten  $[f_\tau(\mathbf{z}_{ijk}^{[T]})]_{i+j+k=3}$  und den Bézier-Bernstein-Koeffizienten  $[(\tilde{c}_\tau)_{ijk}^{[T]}]_{i+j+k=3}$  auf ein lineares Gleichungssystem  $Ax = b$  mit einer Determinanten  $A \in \mathbb{R}^{10 \times 10}$  unabhängig von  $T$ . Die Einträge in  $A$  berechnen sich aus (1.10) und (1.12) wie folgt:

$$A = \frac{1}{27} \begin{pmatrix} 27 & 1 & 8 & 1 & 8 & 1 & 0 & 0 & 0 & 0 \\ 0 & 12 & 6 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 12 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 12 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 6 & 12 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 12 & 6 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 6 & 12 & 0 & 0 \\ 0 & 0 & 0 & 3 & 3 & 8 & 1 & 8 & 27 & 0 \\ 0 & 8 & 1 & 1 & 0 & 0 & 8 & 1 & 0 & 27 \end{pmatrix}$$

Aufgrund der Diagonaldominanz von  $A$  existiert eine Inverse  $A^{-1} \in \mathbb{R}^{10 \times 10}$ , und die Bézier-Bernstein-Koeffizientenvektoren von  $\tilde{\mathbf{s}}$  sind in eindeutiger Weise bestimmbar. Im Einzelnen lassen sich für jedes Dreieck  $T \in \Delta$  die Komponenten  $\tau = 1, 2, 3$  der Koeffizientenvektoren  $\tilde{\mathbf{c}}_{ijk}^{[T]}$ ,  $i + j + k = 3$ , wie folgt berechnen:

$$\begin{aligned}
\tilde{\mathbf{c}}_{300}^{[T]} &= \mathbf{f}(\mathbf{z}_{300}^{[T]}) \\
\tilde{\mathbf{c}}_{030}^{[T]} &= \mathbf{f}(\mathbf{z}_{030}^{[T]}) \\
\tilde{\mathbf{c}}_{003}^{[T]} &= \mathbf{f}(\mathbf{z}_{003}^{[T]}) \\
\tilde{\mathbf{c}}_{210}^{[T]} &= \frac{1}{6} \left[ 18 \mathbf{f}(\mathbf{z}_{210}^{[T]}) + 2 \mathbf{f}(\mathbf{z}_{030}^{[T]}) - 5 \mathbf{f}(\mathbf{z}_{300}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{120}^{[T]}) \right] \\
\tilde{\mathbf{c}}_{120}^{[T]} &= \frac{1}{6} \left[ 18 \mathbf{f}(\mathbf{z}_{120}^{[T]}) + 2 \mathbf{f}(\mathbf{z}_{300}^{[T]}) - 5 \mathbf{f}(\mathbf{z}_{030}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{210}^{[T]}) \right] \\
\tilde{\mathbf{c}}_{201}^{[T]} &= \frac{1}{6} \left[ 18 \mathbf{f}(\mathbf{z}_{201}^{[T]}) + 2 \mathbf{f}(\mathbf{z}_{003}^{[T]}) - 5 \mathbf{f}(\mathbf{z}_{300}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{102}^{[T]}) \right] \\
\tilde{\mathbf{c}}_{021}^{[T]} &= \frac{1}{6} \left[ 18 \mathbf{f}(\mathbf{z}_{021}^{[T]}) + 2 \mathbf{f}(\mathbf{z}_{003}^{[T]}) - 5 \mathbf{f}(\mathbf{z}_{030}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{012}^{[T]}) \right] \\
\tilde{\mathbf{c}}_{102}^{[T]} &= \frac{1}{6} \left[ 18 \mathbf{f}(\mathbf{z}_{102}^{[T]}) + 2 \mathbf{f}(\mathbf{z}_{300}^{[T]}) - 5 \mathbf{f}(\mathbf{z}_{003}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{201}^{[T]}) \right] \\
\tilde{\mathbf{c}}_{012}^{[T]} &= \frac{1}{6} \left[ 18 \mathbf{f}(\mathbf{z}_{012}^{[T]}) + 2 \mathbf{f}(\mathbf{z}_{030}^{[T]}) - 5 \mathbf{f}(\mathbf{z}_{003}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{021}^{[T]}) \right] \\
\tilde{\mathbf{c}}_{111}^{[T]} &= \frac{1}{12} \left[ 4 \mathbf{f}(\mathbf{z}_{300}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{210}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{120}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{201}^{[T]}) \right. \\
&\quad \left. + 4 \mathbf{f}(\mathbf{z}_{030}^{[T]}) + 54 \mathbf{f}(\mathbf{z}_{111}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{021}^{[T]}) - 9 \mathbf{f}(\mathbf{z}_{102}^{[T]}) \right. \\
&\quad \left. - 9 \mathbf{f}(\mathbf{z}_{012}^{[T]}) + 4 \mathbf{f}(\mathbf{z}_{003}^{[T]}) \right]
\end{aligned} \tag{3.22}$$

Die Berechnung der 10 Koeffizientenvektoren aus den 10 Interpolationswerten durch Gleichung (3.22) erfolgt schnell und unabhängig von der Lage des entsprechenden Dreiecks.

Die in dieser Weise interpolierten stetigen Splines benötigen im Vergleich zu Dreiecksflächen nur etwa ein Viertel der Datenmenge, ohne dass visuelle Unterschiede festgestellt werden können, siehe [41,47]. Auch ohne eine Glättung bietet demnach bereits die Verwendung stetiger Splineoberflächen in dieser Hinsicht erhebliche Vorteile. Abbildung 3.3 zeigt einen menschlichen Kopf in Form einer Dreiecksfläche von 640.000 Dreiecken (Bild links) und einen daraus gewonnenen stetigen kubischen Interpolanten, definiert auf einer Triangulierung von lediglich 9.000 Dreiecken (Bild rechts). Die Darstellung entspricht einer Implementierung von Kohlmüller [47].

Eine Glättung des berechneten stetigen Interpolanten  $\tilde{\mathbf{s}}$ , wie wir sie im Folgenden vornehmen, bringt eine weitere Reduktion benötigter Daten bei gleichbleibender visueller Qualität hervor.



Abbildung 3.3: Eine feine Triangulierung mit 640.000 Dreiecken (links) und der dazu gehörige stetige Interpolant über 90.000 Dreiecken (rechts)

### 3.3 Konstruktion einer groben Quadrangulierung im Raum

Die in Abschnitt 3.4 beschriebene Modifikation von Bézier-Bernstein-Koeffizienten basiert auf einer (groben) Quadrangulierung im Raum. Da der stetige Interpolant jedoch auf einer 3D-Triangulierung definiert ist, entwickeln wir zunächst eine Methode, um benachbarte Dreiecke zu Vierecken zusammenzufassen.

Wir erhalten dadurch eine Menge  $\diamond$ , genannt *quadrangulierte 3D-Triangulierung*, welche die Paare der als Vierecke zusammengefassten Dreiecke und außerdem übrig gebliebene einzelne Dreiecke enthält. Sei  $N_1$  die Anzahl der zusammengefassten Paare,  $N_2$  die Anzahl der übrig gebliebenen Dreiecke und  $N = N_1 + N_2$ , so schreiben wir

$$\diamond = \{ \{T_1^\Delta, T_1^\nabla\}, \dots, \{T_{N_1}^\Delta, T_{N_1}^\nabla\} \} \cup \{T_1, \dots, T_{N_2}\}. \quad (3.23)$$

Paare von Dreiecken werden durch folgenden Algorithmus so zu Vierecken zusammengefasst, dass nur wenige Dreiecke im Inneren der Triangulierung einzeln bleiben:

#### Algorithmus 3.5

Man fasse benachbarte Dreiecke einer Triangulierung  $\Delta$  in der Ebene oder im Raum wie im Folgenden beschrieben zu Vierecken zusammen und bezeichne die Menge der zusammengefassten und übrig bleibenden Dreiecke mit  $\diamond$ . Man fasse dabei jeweils nur Paare zusammen, welche, bezüglich der gemeinsamen Kante in eine Ebene gedreht, ein strikt konvexes Viereck bilden.



1. Man wähle einen Eckpunkt  $v$  von  $\Delta$  und setze

$$\mathcal{S}_i := \begin{cases} \emptyset & \text{für } i = 0, \\ \text{star}_1^\Delta(v) & \text{für } i = 1, \\ \text{star}_i^\Delta(v) \setminus \text{star}_{i-1}^\Delta(v) & \text{für } i \geq 2. \end{cases}$$

Für  $i = 1, 2, \dots, m$  wiederhole man die Schritte 2 bis 4 solange  $\mathcal{S}_i \neq \emptyset$  ist.

2. Falls nicht durchlaufene Dreiecke aus  $\mathcal{S}_i$  mit zwei Nachbardreiecken in  $\mathcal{S}_{i-1}$  existieren, bezeichne man die Menge dieser Dreiecke mit  $\mathcal{T}_2$  und verfare mit jedem Dreieck  $T \in \mathcal{T}_2$  wie folgt:

- (a) Besitzt  $T$  nicht zusammengefasste Nachbardreiecke in  $\mathcal{S}_{i-1}$ , so fasse man  $T$  mit einem dieser Nachbarn zusammen.
- (b) Besitzt  $T$  andernfalls nicht zusammengefasste Nachbardreiecke in  $\mathcal{S}_i$ , so fasse man  $T$  mit einem dieser Nachbarn zusammen.
- (c) Andernfalls lasse man  $T$  einzeln.

3. Falls nicht durchlaufene Dreiecke  $T \in \mathcal{S}_i$  mit genau einem Nachbardreieck in  $\mathcal{S}_{i-1}$  existieren, bezeichne man die Menge dieser Dreiecke mit  $\mathcal{T}_1$  und verfare wie folgt:

3.1 Man beginne mit einem beliebigen Dreieck  $T \in \mathcal{T}_1$ .

- (a) Falls  $T$  einen nicht zusammengefassten Nachbarn in  $\mathcal{S}_{i-1}$  besitzt, so fasse man dieses benachbarte Paar zusammen.
- (b) Falls  $T$  andernfalls zwei nicht zusammengefasste Nachbarn  $T_1, T_2 \in \mathcal{S}_i$  besitzt, so verfare man wie folgt: Man bezeichne für  $i = 1, 2$  mit  $\mathcal{M}_i \subseteq \mathcal{S}_i$  die größtmögliche benachbarte Menge nicht zusammengefasster Dreiecke ohne Nachbarn in  $\mathcal{S}_{i-1}$ , die  $T_i$  enthält.
  - (i) Sind sowohl  $\mathcal{M}_1$  als auch  $\mathcal{M}_2$  Ketten mit einem zusammengefassten Nachbarn in  $\mathcal{S}_i$ , so überprüfe man die Längen  $l_1$  und  $l_2$  dieser Ketten. Ist genau eine Länge ungerade, so fasse man  $T$  mit dem entsprechenden Dreieck zusammen, ansonsten wähle man ein beliebiges der Dreiecke.
  - (ii) Ist hingegen genau eine Menge  $\mathcal{M}_i$ ,  $i \in \{1, 2\}$ , eine Kette wie in (i), so fasse man  $T$  mit  $T_i$  zusammen, falls die entsprechende Länge  $l_i$  ungerade ist, ansonsten mit dem anderen Dreieck.
  - (iii) Ist weder  $\mathcal{M}_1$  noch  $\mathcal{M}_2$  eine Kette wie in (i), so fasse man  $T$  mit einem beliebigen Dreieck  $T_1$  oder  $T_2$  zusammen.
- (c) Falls  $T$  andernfalls genau einen nicht zusammengefassten Nachbarn in  $\mathcal{S}_i$  besitzt, so fasse man  $T$  mit diesem zusammen.
- (d) Andernfalls lasse man das Dreieck  $T$  einzeln.

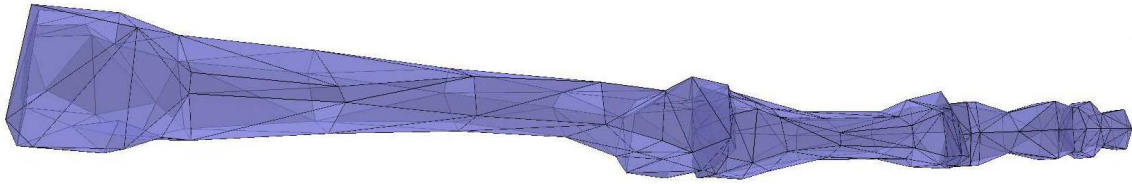


Abbildung 3.4: Beispiel einer 3D-Triangulierung ohne Randdreiecke

- 3.2 Falls ein Dreieck  $T \in \mathcal{T}_1$  verbleibt, welches das zuletzt durchlaufene, nicht einzeln gelassene Dreieck berührt, so verfähre man für  $T$  wie in 3.1.
- 3.3 Solange Dreiecke  $T \in \mathcal{T}_1$  verbleiben, wiederhole man Schritt 3.2.
4. Falls Dreiecke aus  $\mathcal{S}_i$  verbleiben, so bezeichne man die Menge dieser Dreiecke mit  $\mathcal{T}_0$  und verfähre wie folgt:
  - 4.1 Solange Dreiecke  $T \in \mathcal{T}_0$  mit zusammengefasstem Nachbarn existieren, so fasse man  $T$ , falls es einen nicht zusammengefassten Nachbarn in  $\mathcal{S}_i$  besitzt, mit diesem zusammen und lasse  $T$  ansonsten einzeln.
  - 4.2 Solange weitere Dreiecke in  $\mathcal{T}_0$  existieren, fasse man jedes solches Dreieck  $T$ , falls es einen nicht zusammengefassten Nachbarn  $T_1$  besitzt, mit diesem zusammen und lasse  $T$  ansonsten einzeln.

Durch diese Vorgehensweise wird erreicht, dass die konstruierten Zerlegungen  $\diamond$  nur sehr wenige einzelne Dreiecke im Inneren von  $\diamond$  enthalten. Dadurch, dass Dreiecke aus  $\mathcal{S}_i$  mit zwei Nachbardreiecken aus  $\mathcal{S}_{i-1}$  vorrangig zu Dreiecken mit genau einem Nachbardreiecke aus  $\mathcal{S}_{i-1}$  und diese wiederum vorrangig zu solchen ohne Nachbardreiecke aus  $\mathcal{S}_{i-1}$  behandelt werden, wird erreicht, dass nur in Sonderfällen Dreiecke aus  $\mathcal{S}_{i-1}$  einzeln bleiben.

Wir stellen im Folgenden die verschiedenen Sonderfälle dar, in denen einzelne Dreiecke im Inneren von  $\diamond$  verbleiben. Dabei sei  $T \in \mathcal{S}_i \subseteq \diamond$  für  $i \in \{1, \dots, m\}$  ein solches einzelnes inneres Dreieck. Dann gilt offensichtlich eine der folgenden Aussagen:

- (i)  $T$  ist ein inneres Dreieck von  $\diamond$  mit ausschließlich zusammengefassten Nachbardreiecken, und es besitzt kein Nachbardreieck in  $\mathcal{S}_{i+1}$ .
- (ii)  $T$  ist ein inneres Dreieck von  $\diamond$  mit ausschließlich zusammengefassten Nachbardreiecken, und es besitzt ein oder zwei Nachbardreiecke in  $\mathcal{S}_{i+1}$ , von denen jedes mindestens ein weiteres Nachbardreieck in  $\mathcal{S}_i$  besitzt, mit dem es zusammengefasst ist.

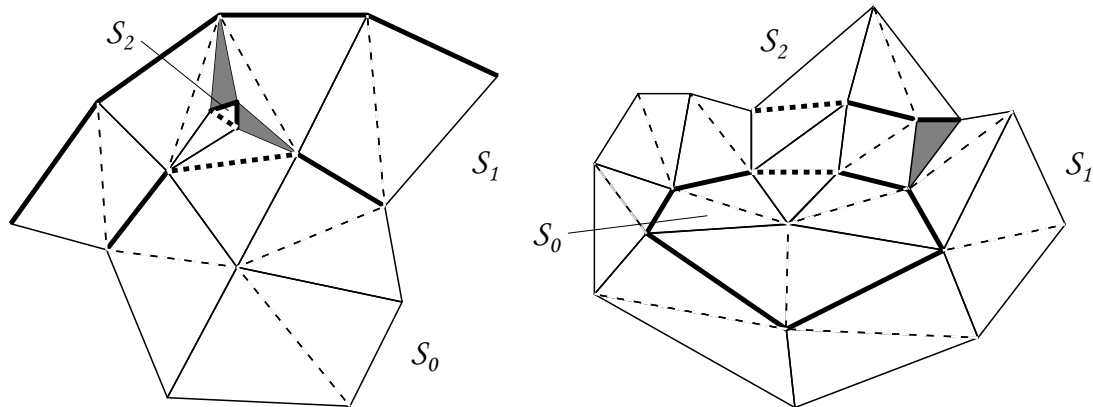


Abbildung 3.5: Sonderfälle einzelner innerer Dreiecke; Die gemeinsamen Kanten der zusammengefassten Dreiecke sind gestrichelt dargestellt. Die einzeln bleibenden Dreiecke sind grau schraffiert.

- (iii)  $T$  bildet, mit einem entsprechenden Nachbardreiecke  $T'$  zusammengefasst und bezüglich  $T \cap T'$  in eine gemeinsame Ebene gedreht, kein strikt konvexes Viereck.

Zu (i) bemerken wir Folgendes: Besäße  $T$  einen nicht zusammengefassten Nachbarn, so wäre  $T$  mit diesem Nachbarn zusammengefasst worden. Besäße  $T$  ein Nachbardreieck in  $\mathcal{S}_{i+1}$ , so wäre dieser Nachbar gemäß 3.1 (a) mit  $T$  zusammengefasst worden.

In den Situationen (i) und (ii) müssen weitere Bedingungen vorliegen, damit  $T$  tatsächlich einzeln bleibt. Für (i) gelten folgende Überlegungen: Da Dreiecke mit Nachbarn aus  $\mathcal{S}_{i-1}$  vorrangig zu Dreiecken ohne solche Nachbarn behandelt werden, sind nur die folgenden Situationen denkbar:

1.  $T$  besitzt drei Nachbarn aus  $\mathcal{S}_{i-1}$ .
2.  $T$  besitzt zwei Nachbarn aus  $\mathcal{S}_{i-1}$  und einen Nachbarn aus  $\mathcal{S}_i$ , der ebenfalls mindestens ein Nachbardreieck aus  $\mathcal{S}_{i-1}$  besitzt.
3.  $T$  besitzt einen Nachbarn aus  $\mathcal{S}_{i-1}$  und zwei Nachbarn aus  $\mathcal{S}_i$ , die beide ebenfalls jeweils mindestens ein Nachbardreieck aus  $\mathcal{S}_{i-1}$  besitzen.

Fall 1 kann nur bei Triangulierungen  $\Delta$  mit sehr verschieden großen Dreiecken oder bei solchen 3D-Triangulierungen auftreten, bei denen Dreiecke durch die Krümmung der Oberfläche wieder „zusammenlaufen“ (vgl. Abbildung 3.4). Außerdem müssen, damit  $T$  einzeln bleibt, alle drei Nachbardreiecke aus  $\mathcal{S}_{i-1}$  mit von  $T$  verschiedenen Dreiecken zusammengefasst sein. Dass Fall 1 auftritt und  $T$  einzeln bleibt, ist ein selten auftretender Sonderfall. Ein Beispiel zeigt Abbildung 3.5 (links).

Wir betrachten Fall 2: Besitzt  $T$  genau zwei Nachbarn in  $\mathcal{S}_{i-1}$ , so ist der dritte Nachbar  $\tilde{T}_1$  von  $T$  aus  $\mathcal{S}_i$  und besitzt auch zwei Nachbarn aus  $\mathcal{S}_{i-1}$ , andernfalls würde  $T$  gemäß Schritt 2 im Algorithmus vorrangig behandelt und mit  $\tilde{T}_1$  zusammengefasst. Dass zwei benachbarte Dreiecke aus  $\mathcal{S}_i$  jeweils zwei Nachbarn in  $\mathcal{S}_{i-1}$  besitzen und dass genau eines davon einen nicht zusammengefassten Nachbarn in  $\mathcal{S}_{i-1}$  besitzt und das andere keinen, ist ein seltener Sonderfall.

Es verbleibt Fall 3: Seien  $\tilde{T}_1$  und  $\tilde{T}_2$  die in  $\mathcal{S}_i$  liegenden Nachbardreiecke von  $T$  und  $\tilde{T}_3$  sowie  $\tilde{T}_4$  deren Nachbarn in  $\mathcal{S}_{i-1}$ . Damit  $T \in \Delta$  als einzelnes, inneres Dreieck verbleibt, muss nun  $\tilde{T}_3$  mit  $\tilde{T}_1$  zusammengefasst sein und  $\tilde{T}_4$  mit  $\tilde{T}_2$ . Diese Situation tritt nur in Sonderfällen auf.

In (ii) gehen wir zunächst davon aus, dass  $T$  genau einen Nachbarn  $\tilde{T}_1$  in  $\mathcal{S}_i$  besitzt. Dieser muss wie beschrieben einen Nachbarn  $\tilde{T}_2$  in  $\mathcal{S}_{i+1}$  besitzen, mit dem er zusammengefasst ist. Da  $\tilde{T}_1$  gemäß 3.1(a) jedoch vorrangig mit  $T$  zusammengefasst würde, müssen  $\tilde{T}_1$  und  $\tilde{T}_2$  beim Durchlauf von  $\tilde{T}_2$  zusammengefasst werden. Insbesondere darf das Dreieck  $\tilde{T}_2$  keinen nicht zusammengefassten Nachbarn in  $\mathcal{S}_i$  und keinen weiteren Nachbarn  $\tilde{T}_3$  in  $\mathcal{S}_i$  besitzen, mit dem es gemäß 3.1(b) vorrangig zusammengefasst wird. Abbildung 3.5 (rechts) zeigt einen solchen Sonderfall, in welcher all diese Bedingungen zusammentreffen. Offensichtlich ist der Fall, dass  $T$  einen zweiten Nachbarn in  $\mathcal{S}_i$  besitzt und einzeln bleibt, nochmals wesentlich seltener.

Wir betrachten nun Situation (iii): In groben Triangulierungen  $\Delta$ , wie sie durch typische Methoden des mesh simplification generiert werden, treten nur selten Dreiecke mit Innenwinkel  $> 90^\circ$  auf. Somit ist die Summe benachbarter Winkel in den weitaus meisten Fällen  $< 180^\circ$ , und Situationen wie in (iii) beschrieben treten nur in Sonderfällen auf.

Wegen dieser Argumente bleibt im Allgemeinen ein sehr geringer Anteil innerer Dreiecke einzeln.

### 3.4 Modifikation von Bézier-Bernstein-Koeffizienten

Wir beschreiben im Folgenden die Glättung des Bézier-Bernstein-Netzes der stetigen Splineoberfläche: Sukzessive glätten wir den Interpolanten über jedem Eckpunkt, welcher mindestens zwei Dreiecken aus  $\diamond$  angehört, und außerdem über jeder inneren Kante. Wir erhalten die zugehörige Triangulierung  $\diamond$  hierbei, indem wir, grob gesprochen, die Elemente  $Q \in \diamond$  in geeigneter Weise durchlaufen und dabei bestimmte Dreiecke  $T \in \Delta$  weiter unterteilen. Die Glättung erfolgt wie in Unterabschnitt 3.1 beschrieben durch eine Projektion gewisser Bézier-Bernstein-Koeffizienten in bestimmte Tangentialebenen in den gemeinsamen Eckpunkten bzw. in

den Mittelpunkten der gemeinsamen Kanten. Die Reihenfolge der geglätteten Polynomstücke wird durch eine Färbung der Vierecke wie in Abschnitt 2.2 oder durch Prioritätsprinzipien wie in Abschnitt 2.4 ermittelt.

Zunächst beschreiben wir in Unterabschnitt 3.4.1 die bei der Anwendung von Färbungsalgorithmen und Prioritätsprinzipien auf quadrangulierte 3D-Triangulierungen zu berücksichtigenden Besonderheiten und Ausnahmen. In Unterabschnitt 3.4.2 beschreiben wir die Glättung des Interpolanten.

In Abschnitt 3.5 zeigen wir, dass die Interpolationsmethode lokal ist und für die Interpolanten ein geringerer Datenaufwand benötigt wird als in [47]. Geht man von einer groben Triangulierung der Ebene aus, so besitzen die geglätteten Interpolanten darüber hinaus optimale Approximationsordnung. Für diese Aussage geben wir einen umfangreichen und komplexen Beweis, wie er im Wesentlichen von Nürnberger, Kohlmüller und Zeilfelder [48] entwickelt wurde.

### 3.4.1 Färbungs- und Prioritätsprinzipien

Wir stellen nun die Besonderheiten dar, die bei der Anwendung von Färbungsverfahren und Prioritätsprinzipien der Abschnitte 2.2 bis 2.4 auf quadrangulierte 3D-Triangulierungen im Raum zu berücksichtigen sind.

Wir betrachten zunächst den Färbungsalgorithmus 2.9 von Nürnberger, Schumaker und Zeilfelder [68]. Offensichtlich ist dieses Verfahren analog für quadrangulierte 3D-Triangulierungen anwendbar:

#### Algorithmus 3.6

Sei  $\diamond$  eine quadrangulierte 3D-Triangulierung. Man beginne mit einer beliebigen Schwarzweißfärbung von  $\diamond$ . Solange ein Element  $Q \in \diamond$  mit drei oder vier gleichfarbigen Nachbarn existiert, wechsele man die Farbe von  $Q$ .

Wir modifizieren nun die Färbungsmethoden aus Abschnitt 2.2 und beginnen mit Konstruktionsalgorithmus 2.1. Dabei ist zu berücksichtigen, dass quadrangulierte 3D-Triangulierungen nicht notwendigerweise einfach zusammenhängende Gebiete  $\Omega$  besitzen:

#### Algorithmus 3.7

Man konstruiere eine quadrangulierte 3D-Triangulierung  $\diamond$  entsprechend Algorithmus 2.1, wobei anstelle der Vierecke  $Q \in \diamond$  einzelne Dreiecke und Paare von Dreiecken betrachtet werden.

Bezüglich der Reihenfolge der angefügten Kegel beachte man folgende Regel: Angenommen, ein Teilgebiet  $\Omega_{i-1}$  ist einfach zusammenhängend, und durch Anfügen

eines Kegels  $K_i$  entsteht eine Teilzerlegung  $\diamond_i$  mit nicht einfach zusammenhängendem Gebiet  $\Omega_i$ , wobei eine zu  $\diamond_i$  disjunkte Teilzerlegung  $\bar{\diamond}$  von  $\diamond$  des Gebiets  $\bar{\Omega}$  existiert, sodass  $\Omega_i \cup \bar{\Omega}$  einfach zusammenhängend ist. Dann füge man geeignete Kegel  $\tilde{K}_i, \dots, \tilde{K}_{i+n}$ ,  $n \in \mathbb{N}$ , an, sodass  $\tilde{\diamond}_{i+n} \supseteq (\diamond_i \cup \bar{\diamond})$  folgt und  $\tilde{\Omega}_j$  einfach zusammenhängend ist für alle  $i \leq j \leq i+n$ . Offensichtlich existieren stets solche Kegel  $\tilde{K}_j$ ,  $i \leq j \leq i+n$ , vgl. Beweis zu Theorem 2.2.

Grob gesprochen werden Kegel in einer Reihenfolge angefügt, sodass  $\Omega_i$  „so lange wie möglich“ einfach zusammenhängend ist. Offensichtlich lassen sich auf diese Weise, analog zu Theorem 2.2, beliebige quadrangulierte 3D-Triangulierungen durchlaufen. Die Begriffe aus Abschnitt 2.1, wie *Zellen*, *Kegel*, *Ketten*, *T-Systeme* und *Anfügen*, gelten hier für  $\diamond$  analog, indem anstelle der Vierecke  $Q \in \diamond$  die zu Vierecken zusammengefassten Dreieckpaare und die einzelnen Vierecke betrachtet werden.

Ist für ein  $i \in \{1, \dots, n\}$  das Gebiet  $\Omega_i$ , nicht aber  $\Omega_{i-1}$  einfach zusammenhängend, so wird  $K_i$  angefügt, sodass  $K_i$  und  $\Omega_{i-1}$  alle Randkanten von  $K_i$  gemeinsam haben. Wir geben im folgenden eine Lemma 2.3 entsprechende Aussage, in welcher dies berücksichtigt wird:

### Lemma 3.8

*Es werde ein Kegel  $K$  von mindestens zwei Elementen an eine quadrangulierte 3D-Triangulierung  $\diamond$  angefügt, wobei  $K$  und  $\diamond$  mindestens drei gemeinsame Kanten besitzen. Dann lässt sich  $K$  in ein Tupel  $(K_1, \dots, K_n)$  von Kegeln mit folgenden Eigenschaften zerlegen, sodass  $K_1, \dots, K_n$  an  $\diamond$  angefügt werden können:*

$$(i) \quad |K_i| = 1, \quad i = 1, \dots, n-1$$

$$(ii) \quad |K_n| = 1, \text{ oder } K_n \text{ wird mit genau zwei gemeinsamen Kanten angefügt.}$$

Der Beweis verläuft im Wesentlichen analog zu jenem von Lemma 2.3. Da jedoch die dortigen Aussagen I und (k) nicht erfüllt sind, besitzt für den Fall  $\mathcal{B} \cup \mathcal{C} = \emptyset$  der zuletzt angefügte, einelementige Kegel  $K_n$  genau drei Nachbarn in  $\diamond_{i-1}$ , falls es sich um ein einzelnes Dreieck handelt und genau vier Nachbarn in  $\diamond_{i-1}$ , falls es sich um ein Paar von Dreiecken handelt. Dieser Fall muss im Färbungsalgorithmus berücksichtigt werden.

Damit lässt sich Färbungsalgorithmus 2.4 wie folgt auf quadrangulierte 3D-Triangulierungen anwenden:

### Algorithmus 3.9

Man färbe eine gemäß Algorithmus 3.7 konstruierte quadrangulierte 3D-Triangulierung entsprechend der Vorgehensweise wie in Algorithmus 2.4 aus Kapitel 2 beschrieben.

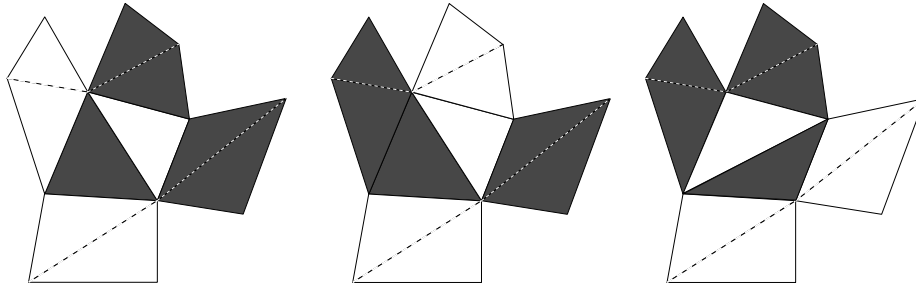


Abbildung 3.6: Sonderfall eines zuletzt im Inneren verbleibenden ungefärbten Vierecks bei jeweils zwei schwarzen und zwei weißen Nachbarelementen

Hinsichtlich Fall 1.3, *Färbung eines angefügten einelementigen Kegels*  $K_i = \{Q\}$ , ist eine weitere Situation zu berücksichtigen:  $Q \in \diamond$  ist ein Element mit vier Nachbarn in  $\diamond_{i-1}$ . Offensichtlich handelt es sich dabei um ein Paar  $\{T^\Delta, T^\nabla\}$  von Dreiecken. Besitzen drei oder vier Nachbarelemente von  $Q$  die gleiche Farbe, so färbe man  $Q$  in der anderen Farbe.

Besitzen dagegen jeweils genau zwei Nachbarn die gleiche Farbe, so gehe man wie folgt vor: Dabei seien die vier Eckpunkte von  $Q$  mit  $v_1$  bis  $v_4$  bezeichnet, wobei  $[v_1, v_3]$  die durchgezogene Diagonale sei und  $T^\Delta$  den Eckpunkt  $v_2$  sowie  $T^\nabla$  den Eckpunkt  $v_4$  enthalte. Besitzen die beiden Nachbarelemente von  $Q$  mit Eckpunkt  $v_2$  dieselbe Farbe, so entferne man  $Q$  aus  $\diamond$ , füge die beiden einzelnen Dreiecke  $T^\Delta$  sowie  $T^\nabla$  zu  $\diamond$  hinzu und färbe  $T^\nabla$  in dieser Farbe sowie  $T^\Delta$  in der anderen Farbe, siehe Abbildung 3.6 (links). Besitzen dagegen für alle  $i = 1, 2, 3, 4$  die beiden den Eckpunkt  $v_i$  enthaltenden Nachbarelemente unterschiedliche Farben, so modifiziere man  $\diamond$  auf dieselbe Weise und wähle für  $T^\nabla$  eine beliebige sowie für  $T^\Delta$  die andere Farbe, siehe Abbildung 3.6 (Mitte). Besitzen schließlich die beiden Nachbarelemente von  $Q$  mit Eckpunkt  $v_1$  dieselbe Farbe, so entferne man  $Q$  aus  $\diamond$ , füge die beiden Dreiecke  $\tilde{T}^\Delta = [v_1, v_2, v_4]$  sowie  $\tilde{T}^\nabla = [v_2, v_3, v_4]$  zu  $\diamond$  hinzu und färbe  $\tilde{T}^\nabla$  in dieser sowie  $\tilde{T}^\Delta$  in der anderen Farbe, siehe Abbildung 3.6 (rechts).

Bei der Anwendung des Färbungsverfahrens ist von Bedeutung, dass Lemma 2.5 und Bemerkung 2.6 analog gelten. Offensichtlich sind alle in den zugehörigen Beweisen verwendeten Argumente hier entsprechend anwendbar. Insbesondere folgen die lineare Komplexität des Algorithmus sowie die Tatsache, dass benachbarte Mengen gleichfarbiger Elemente stets nicht geschlossene Ketten von meist kurzer Länge bilden.

Zuletzt übertragen wir das in Abschnitt 2.4 verwendete Prioritätsprinzip auf quadrangulierte 3D-Triangulierungen  $\diamond$  im Raum:

**Algorithmus 3.10**

Für  $l = 0, 1, 2, 3, 4$  gehe man wie folgt vor: Solange ein noch nicht gewähltes Element  $Q \in \diamond$  existiert, welches genau  $l$  bereits gewählte Nachbarelemente aus  $\diamond$  besitzt, wähle man  $Q$ .

Für  $l = 0, \dots, 4$  bezeichnen wir mit  $\mathcal{K}_l$  die Menge aller in der Iteration  $l$  gewählten Elemente.

Es ist offensichtlich, dass ebenso wie bei Quadrangulierungen Klassen  $\mathcal{K}_0, \dots, \mathcal{K}_4$  entstehen, deren Elemente paarweise nicht benachbart sind. Die Argumente aus dem Beweis zu Theorem 2.28 gelten analog.

### 3.4.2 Glättung des Interpolanten

Ausgehend von diesen Färbungs- und Prioritätsprinzipien triangulieren wir  $\diamond$  und glätten den stetigen Spline-Interpolanten. Dabei wird zu bestimmten Dreieckspaaren  $\{[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3], [\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4]\} \in \diamond$  die zweite Diagonale wie folgt hinzugefügt, siehe Abbildung 3.7:

Zunächst werden die beiden Makrodreiecke bezüglich ihrer gemeinsamen Kante  $[\mathbf{v}_1, \mathbf{v}_3]$  in eine Ebene gedreht und im dadurch entstandenen Viereck  $[\mathbf{v}_1, \tilde{\mathbf{v}}_2, \mathbf{v}_3, \tilde{\mathbf{v}}_4]$  der Ebene wird die zweite Diagonale  $[\tilde{\mathbf{v}}_2, \tilde{\mathbf{v}}_4]$  hinzugefügt, wobei  $\mathbf{v}_5$  den Diagonalschnittpunkt bezeichne. Schließlich werden die beiden Makrodreiecke durch die vier Mikrodreiecke

$$[\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_5], \quad i = 1, 2, 3, 4, \quad j = i + 1 \text{ für } i \leq 3, \quad j = 1 \text{ sonst.}$$

unterteilt.

Die Bézier-Bernstein-Koeffizienten auf den beiden Makrodreiecken von  $\tilde{\mathbf{s}}$  lassen sich gemäß Satz 1.9 in einfacher und effizienter Weise für jede Komponente  $\tau = 1, 2, 3$  auf diese vier Mikrodreiecke umrechnen. Des Weiteren werden gewisse einzelne Dreiecke  $T \in \diamond$  durch Clough-Tocher-Splits unterteilt. Bei einem Clough-Tocher-Split wird jeder Eckpunkt  $\mathbf{v}_i$ ,  $i = 1, 2, 3$ , von  $T$  mit dem Baryzentrum  $\mathbf{v}_0 = \frac{1}{3} \sum_{i=1}^3 \mathbf{v}_i$  verbunden, und man erhält drei Mikrodreiecke  $T_1, T_2, T_3$ . Auch hier erfolgt eine Umrechnung der entsprechenden Bézier-Bernstein-Koeffizienten für jede Komponente des Interpolanten.

Wir beschreiben die Unterteilung von  $\diamond$  und Glättung von  $\tilde{\mathbf{s}}$  zunächst für den gefärbten Fall: Analog wie in Abschnitt 2.3 beschrieben, wird jedes schwarze Element  $Q_i$  mit einem ungeraden Index  $i$  in in einer Kette  $(Q_1, \dots, Q_n)$  der Länge  $n \geq 3$  und jedes weiße Element weiter unterteilt. Für jedes zu unterteilende Dreieckpaar  $Q$  fügen



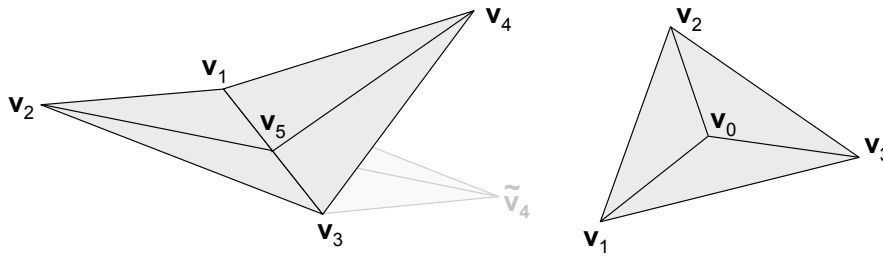


Abbildung 3.7: Unterteilung von Dreieckspaaren und einzelnen Dreiecken

wir, wie beschrieben, die zweite Diagonale hinzu, für jedes zu unterteilende einzelne Dreieck verwenden wir einen Clough-Tocher-Split.

Durch folgenden Algorithmus wird der Interpolant über den Eckpunkten und Kanten der unterteilten und gefärbten Triangulierung  $\diamond$  geglättet:

### Algorithmus 3.11

Man glätte den stetigen Interpolanten  $\tilde{s}$  über den Eckpunkten und Kanten von  $\diamond$  unter Verwendung der Algorithmen 3.1 und 3.3 wie folgt:

1. Auf jedem Eckpunkt  $\mathbf{v}$  von  $\diamond$ , der mindestens zwei Dreiecken aus  $\diamond$  angehört, glätte man  $\tilde{s}$  durch Modifikation der Koeffizienten des  $C^1$ -Rings um  $\mathbf{v}$ .
2. In jedem einzelnen nicht unterteilten Element  $Q = \{T^\Delta, T^\nabla\} \in \diamond$  glätte man  $\tilde{s}$  auf der gemeinsamen Kante  $T^\Delta \cap T^\nabla$  durch Modifikation der zu den Baryzentren von  $T^\Delta$  und  $T^\nabla$  gehörenden Bézier-Bernstein-Koeffizienten. Die durchlaufenen einzelnen und nicht unterteilten Dreiecke aus  $\diamond$  werden nicht modifiziert.
3. Für jedes Paar  $\{Q_1, Q_2\}$  nicht unterteilter Elemente aus  $\diamond$  gehe man wie folgt vor: Die enthaltenen Dreiecke bilden eine nicht geschlossene Kette  $(T_1, \dots, T_n)$  der Länge  $l = 2$ , falls es sich um zwei einzelne Dreiecke handelt, der Länge  $l = 3$ , falls es sich um ein einzelnes Dreieck und ein Paar zusammengefasster Dreiecke handelt und der Länge  $l = 4$ , falls es sich um zwei Paare zusammengefasster Dreiecke handelt.
  - l=2: Man glätte  $\tilde{s}$  auf der gemeinsamen Kante  $T_1 \cap T_2$ .
  - l=3: Man glätte  $\tilde{s}$  auf den beiden Dreiecken  $T_1$  und  $T_3$  bezüglich des Dreiecks  $T_2$  durch Modifikation der im Inneren von  $T_1$  und  $T_3$  liegenden Bézier-Bernstein-Koeffizienten.
  - l=4: Man glätte  $\tilde{s}$  zunächst auf der Kante  $T_2 \cap T_3$  und im Anschluss auf den beiden Dreiecken  $T_1$  sowie  $T_4$  bezüglich  $T_2$  und  $T_3$ .
4. Für jedes unterteilte schwarze Element  $Q \in \diamond$  gehe man wie folgt vor: Ist  $Q$  ein Dreieck, dann glätte man  $\tilde{s}$  zunächst auf jedem Mikrodreieck  $T_i$ ,  $i = 1, 2, 3$ ,

mit bereits durchlaufenem Nachbarerelement in  $\diamond$  bezüglich dieses Nachbarn. Im Anschluss lassen sich für jede Komponente  $\tau = 1, 2, 3$  zunächst die Bézier-Bernstein-Koeffizienten des  $C^1$ -Rings um das Baryzentrum  $\mathbf{v}_0$  von  $Q$  und anschließend der zu  $\mathbf{v}_0$  gehörende Koeffizient wie im bivariaten Fall gemäß (1.17) und (1.16) modifizieren, sodass  $\tilde{s}_\tau|_Q$  und somit  $\tilde{\mathbf{s}}|_Q$  differenzierbar ist.

Ist  $Q$  hingegen ein unterteiltes Paar  $\{T^\Delta, T^\nabla\}$  zusammengefasster Dreiecke, so glätte man  $\tilde{\mathbf{s}}$  wiederum auf jedem Mikrodreieck  $T_i$ ,  $i = 1, 2, 3, 4$ , mit bereits durchlaufenem Nachbarerelement aus  $\diamond$  bezüglich dieses Nachbarn und verfähre im Anschluss wie folgt: Zunächst drehe man die Mikrodreiecke bezüglich der Diagonalen  $T^\Delta \cap T^\nabla$  in eine gemeinsame Ebene, das entsprechende Makrovier-eck heiße  $\tilde{Q}$ . Dann modifiziere man für jede Komponente  $\tau = 1, 2, 3$  alle Bézier-Bernstein-Koeffizienten mit Abstand  $\leq 1$  zum Diagonalschnittpunkt  $\mathbf{v}_0$  wie im bivariaten Fall.

5. Für jedes weiße Element aus  $\diamond$  mit geradem Index gehe man wie in Schritt 4 vor.
6. Schließlich verfähre man auf dieselbe Weise mit allen verbleibenden Elementen.

Weiterhin betrachten wir die Anwendung des Prioritätsprinzips aus Algorithmus 3.10 für die Glättung des Interpolanten. Wir unterteilen dabei jedes Element  $Q$  der Klassen  $\mathcal{K}_2$ ,  $\mathcal{K}_3$  und  $\mathcal{K}_4$  wie folgt: Ist  $Q$  ein Paar von Dreiecken, so fügen wir wie oben beschrieben die zweite Diagonale hinzu, andernfalls unterteilen wir  $Q$  durch einen Clough-Tochter-Split. Für jede Komponente  $\tau = 1, 2, 3$  werden die Bézier-Bernstein-Koeffizienten von  $\tilde{\mathbf{s}}$  gemäß Satz 1.9 auf die Mikrodreiecke umgerechnet. Die resultierende Triangulierung sei wiederum mit  $\diamond$  bezeichnet.

Wir beschreiben nun die Glättung des Interpolanten bei Anwendung des Prioritätsprinzips:

### Algorithmus 3.12

Man glätte den stetigen Interpolanten  $\tilde{\mathbf{s}}$  über den Eckpunkten und Kanten von  $\diamond$  unter Verwendung der Algorithmen 3.1 und 3.3 wie folgt:

1. Auf jedem Eckpunkt  $\mathbf{v}$  von  $\diamond$ , der mindestens zwei Dreiecken aus  $\diamond$  angehört, glätte man  $\tilde{\mathbf{s}}$  durch Modifikation der Koeffizienten des  $C^1$ -Rings um  $\mathbf{v}$ .
2. Für jedes Element  $Q$  der Klasse  $\mathcal{K}_0$  verfähre man wie in Algorithmus 3.11, Schritt 2.
3. Für jedes Dreieckspaar  $Q$  aus  $\mathcal{K}_1$  glätte man  $\tilde{\mathbf{s}}$  zunächst über dem Dreieck  $T^\Delta$  von  $Q$  mit Nachbarerelement aus  $\mathcal{K}_0$  bezüglich dieses Nachbarn. Im Anschluss glätte man das andere Dreieck von  $Q$  bezüglich  $T^\Delta$ . Jedes einzelne Dreieck aus  $\mathcal{K}_1$  glätte man bezüglich seines Nachbarn aus  $\mathcal{K}_0$ .
4. Für  $l = 2, 3, 4$  gehe man wie folgt vor: Für jedes Element der Klasse  $Q \in \mathcal{K}_l$  verfähre man wie in Algorithmus 3.11, Schritt 4.

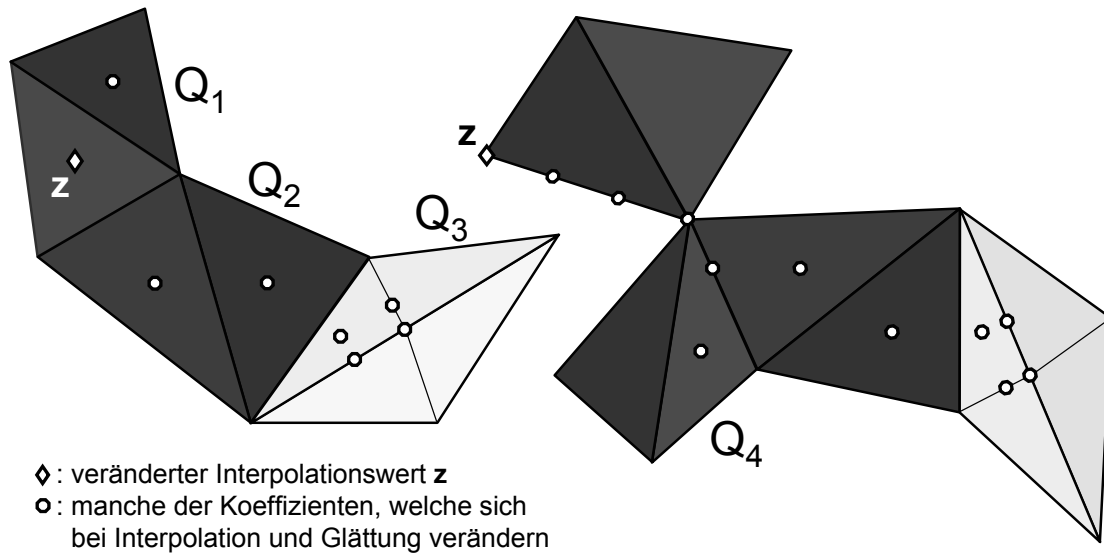


Abbildung 3.8: Maximale Veränderung von  $\mathbf{s}$  bei der Veränderung eines Interpolationswertes im Inneren (links) oder auf dem Rand (rechts) eines Vierecks

## 3.5 Eigenschaften der Methode

In diesem Abschnitt zeigen wir die Lokalität der Methode. Weiterhin zeigen wir ähnlich wie in [48], dass die konstruierten Interpolanten optimale Approximationsordnung besitzen, falls Triangulierungen der Ebene betrachtet werden. Schließlich begründen wir, dass für die Darstellung der geglätteten Splines im Schnitt ein geringerer Datenaufwand benötigt wird als in [47].

### 3.5.1 Lokalität

#### Theorem 3.13

Gegeben sei eine feine Triangulierung  $\Omega_f$  und der zugehörige geglättete Interpolant  $\mathbf{s} : \Omega_{\diamond} \rightarrow \mathbb{R}^3$ . Bei Veränderung eines Interpolationswertes  $\mathbf{f}_{\mathbf{z}}, \mathbf{z} \in \mathcal{P}_q(\Delta)$ , verändert sich  $\mathbf{s}$  maximal in einem  $\text{star}^3(Q), Q \in \diamond$ . Insbesondere ist die Interpolationsmethode lokal.

**Beweis:** Wir betrachten zunächst die Veränderung des stetigen Interpolanten  $\tilde{\mathbf{s}}$ . Für die Bestimmung jeder polynomialen 3D-Oberfläche  $\tilde{\mathbf{s}}|_T, T \in \Delta$ , werden lediglich die zu  $T$  gehörenden Interpolationswerte benötigt, und  $\tilde{\mathbf{s}}$  verändert sich nur auf den Dreiecken, welche  $\mathbf{z}$  enthalten. Bei der Unterteilung von Dreiecken wirkt sich die

Veränderung nur auf die Mikrodreiecke der veränderten Makrodreiecke aus. Wir untersuchen nun den geglätteten Interpolanten:

Fall 1:

Liege  $\mathbf{z}$  zunächst im Inneren eines Elements von  $\diamond$ . Da auf jeder Kante von  $\diamond$  bei der stetigen Interpolation vier Interpolationswerte vorliegen, bleibt der stetige und somit auch der geglättete Spline-Interpolant auf den Kanten von  $\diamond$  unverändert.

Wir betrachten zunächst die Methode auf gefärbten Vierecken und Dreiecken: Im hinsichtlich Trägergröße ungünstigsten Fall liegt  $\mathbf{z}$  im Inneren eines Dreiecks  $T_1$  eines Paares  $Q_1 = \{T_1, T_2\}$  schwarzer Dreiecke, welches zu einem weiteren Paar  $Q_2 = \{T_3, T_4\}$  schwarzer Dreiecke benachbart ist, wobei weder  $Q_1$  noch  $Q_2$  weiter unterteilt sind (vgl. Abschnitt 2.3.3). Dann bewirkt die Veränderung von  $\mathbf{z}$  im ungünstigsten Fall, dass sich  $\mathbf{c}_{111}^{[T_i]}$  für alle  $i = 1, 2, 3, 4$  ändert. Im ungünstigsten Fall besitzt  $Q_2$  eine gemeinsame Kante  $E$  mit einem Paar  $Q_3$  weißer Dreiecke, welches in vier Mikrodreiecke  $\{T_5, T_6, T_7, T_8\}$  unterteilt ist. Durch die weitere Glättung verändert sich dann auch der Koeffizient  $\mathbf{c}_{111}^{[T_i]}$  des Dreiecks  $T_i$ ,  $i \in \{5, 6, 7, 8\}$ , mit Nachbarn  $Q_2$  und schließlich die Koeffizienten der  $C^1$ -Scheibe um den Mittelpunkt  $\bigcap_{i=5}^8 T_i$ . Insgesamt verändert sich  $\mathbf{s}$  im ungünstigsten Fall daher auf einem  $\text{star}^2(Q)$ ,  $Q \in Q$ , siehe Abbildung 3.8 (links).

Es bleibt Fall 1 für das Prioritätsprinzip zu untersuchen. Im ungünstigsten Fall liegt  $\mathbf{z}$  im Inneren eines Dreieckspaares  $Q_1$  vom Typ  $l = 0$  mit benachbartem Dreieckspaar  $Q_2$  vom Typ  $l = 1$ , sodass sich  $\mathbf{c}_{111}^{[T]}$  auf allen vier Mikrodreiecken  $T$  von  $Q_1$  und  $Q_2$  ändert. Im ungünstigsten Fall besitzt  $Q_2$  weiterhin ein viergeteiltes benachbartes Element  $Q_3$ , und die Argumentation ist wie im gefärbten Fall. Deshalb zeigt auch hier Abbildung 3.8 (links) eine entsprechende Situation.

Fall 2:

Liege  $\mathbf{z}$  nun auf einer Kante von  $\diamond$ . Im ungünstigsten Fall liegt  $\mathbf{z}$  auf einem Eckpunkt  $\mathbf{v}$  von  $\diamond$ . Durch den veränderten Interpolationswert verändern sich Bézier-Bernstein-Koeffizienten auf allen Kanten aus  $\diamond$  mit Eckpunkt  $\mathbf{v}$ . Durch die Mittelung der Tangentialebenen in allen zu  $\mathbf{v}$  benachbarten Eckpunkten verändern sich im ungünstigsten Fall die Koeffizienten in den  $C^1$ -Scheiben um diese Eckpunkte, also insbesondere auch auf Kanten, die  $\mathbf{v}$  nicht enthalten. Sei  $\mathbf{c}_{201}^{[T_9]}$  ein solcher Koeffizient auf der Diagonalen eines Paares  $Q_4 = \{T_9, T_{10}\}$  schwarzer Dreiecke. Durch die Glättung über die Diagonale dieses Paares verändert sich ein Koeffizient  $\mathbf{c}_{111}^{[T_i]}$ ,  $i \in \{9, 10\}$ . Diese Argumentation gilt bei Anwendung von Färbungsverfahren und Prioritätsprinzip gleichermaßen.

Bei Berücksichtigung beider Fälle liegt  $\mathbf{z}$  im ungünstigsten Fall folglich auf einem Eckpunkt  $\mathbf{v}$  von  $\diamond$ , und der geglättete Spline-Interpolant  $\mathbf{s}$  verändert sich in einem  $\text{star}^3(Q)$ ,  $Q \in \diamond$ , siehe Abbildung 4.15 (rechts).  $\square$

### 3.5.2 Optimale Approximationsordnung

#### Begriffliche Grundlagen

Sei  $\mathbf{g} : \Omega \rightarrow \mathbb{R}^3$   $\nu$ -fach stetig differenzierbar in  $\Omega \subseteq \mathbb{R}^2$ , dann bezeichne für alle  $0 \leq \omega \leq \nu$

$$D^\omega \mathbf{g} = \left( \left( (g_\tau)_{x^\omega y^\mu} \right)_{\mu=0}^\omega \right)_{\tau=1}^3 \quad (3.24)$$

die  $\omega$ -te Ableitung von  $\mathbf{g}$ . Hierbei ist mit  $(\cdot)_{x^\omega y^\mu}$  die  $(\omega - \mu)$ -te Ableitung in  $x$ -Richtung und  $\mu$ -te Ableitung in  $y$ -Richtung bezeichnet. Die  $\omega$ -te Ableitung von  $\mathbf{g}$  in einem Punkt  $z \in \Omega$  enthält demnach alle gemischt partiellen Ableitungen vom Grad  $\omega$  für die drei Komponenten  $g_\tau$ ,  $\tau = 1, 2, 3$ , als Tupel.

Wie gewohnt bezeichne  $|\cdot| : \mathbb{R} \rightarrow \mathbb{R}$  den absoluten Betrag eines reellen Wertes und  $\|\cdot\|_2$  die euklidische Norm im  $\mathbb{R}^n$ ,  $n \in \mathbb{N}$ . Des Weiteren bezeichne  $\|\mathbf{x}\|$  die Maximumsnorm  $\max_{i=1, \dots, n} x_i$  eines Vektors  $\mathbf{x} = (x_1, \dots, x_n)$ ,  $n \in \mathbb{N}$ .

Die Norm der Ableitung  $D^\omega \mathbf{g}$  ist als Maximum aller gemischt partiellen Ableitungen aller Komponenten definiert:

$$\| |D^\omega \mathbf{g}| \| := \max_{\tau=1,2,3, \mu=0, \dots, \omega} \left\{ \| (g_\tau)_{x^\omega y^\mu} \| \right\}$$

Mit diesen Definitionen lässt sich der Begriff der optimalen Approximationsordnung völlig analog zum bivariaten Fall definieren, falls die zum Interpolanten  $\mathbf{s}$  gehörende Triangulierung  $\Delta$  in der Ebene liegt: Gegeben eine zu interpolierende Funktion  $\mathbf{f} = (f_1, f_2, f_3) : \Omega \rightarrow \mathbb{R}^3$  mit  $\Omega \subset \mathbb{R}^2$  und  $f_\tau \in C^4(\Omega)$ ,  $\tau = 1, 2, 3$ , dann existiert eine außer von  $\mathbf{f}$  lediglich vom kleinsten Winkel  $\alpha$  der zugrunde liegenden Triangulierung  $\Delta$  abhängende Konstante  $L > 0$ , sodass  $\mathbf{s} = I\mathbf{f}$  für alle  $\omega \in \{0, \dots, 3\}$  folgende Fehlerabschätzung erfüllt:

$$\| |D^\omega \mathbf{f} - D^\omega \mathbf{s}| \| \leq L h^{4-\omega} \quad (3.25)$$

Hierbei bezeichnet wieder  $h$  den größten Umkreis der Dreiecke von  $\Delta$ .

Die Abbildung  $\mathbf{f}$  sei *regulär*, d. h. jeder (Einheits-)Normalenvektor  $\mathbf{n}(\mathbf{f})(z)$ ,  $z \in \Omega$ , sei wohldefiniert:

$$\mathbf{n}(\mathbf{f})(z) = \frac{\mathbf{f}_x(z) \times \mathbf{f}_y(z)}{\|\mathbf{f}_x(z) \times \mathbf{f}_y(z)\|_2}$$

#### Beweis der optimalen Approximationsordnung

Wie in den Grundlagen beschrieben, bringt die Interpolation mit Polynomen und somit auch die Interpolation mit stetigen, bivariaten Splineoberflächen optimale Approximationsordnung hervor. Offensichtlich folgt mit dem Differenzationsbegriff für

3D-Splines direkt die optimale Approximationsordnung des stetigen Interpolanten  $\tilde{\mathbf{s}}$ . Wir zeigen nun ähnlich wie in [48], dass die optimale Approximationsordnung durch die Modifikation der Bézier-Bernstein-Koeffizienten von  $\tilde{\mathbf{s}}$  erhalten bleibt.

Für den Beweis der optimalen Approximationsordnung des Verfahrens benötigen wir eine Abschätzung für die maximale Abweichung der normierten Normalenvektoren von stetigem Interpolanten  $\tilde{\mathbf{s}}$  und zu interpolierender Funktion  $\mathbf{f}$ . Diese bestimmen wir durch das folgende

**Lemma 3.14**

Gegeben seien die Vektoren  $\mathbf{a}, \tilde{\mathbf{a}}, \mathbf{b}, \tilde{\mathbf{b}} \in \mathbb{R}^3$ , sodass  $\mathbf{a} \times \mathbf{b}$  und  $\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}$  ungleich Null sind. Dann folgt

$$\left\| \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a} \times \mathbf{b}\|_2} - \frac{\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}}{\|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2} \right\| \leq 6 \frac{\|\mathbf{a}\| \cdot \|\mathbf{b} - \tilde{\mathbf{b}}\| + \|\tilde{\mathbf{b}}\| \cdot \|\mathbf{a} - \tilde{\mathbf{a}}\|}{\|\mathbf{a} - \mathbf{b}\|_2}.$$

**Beweis:** Es ist

$$\begin{aligned} & \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a} \times \mathbf{b}\|_2} - \frac{\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}}{\|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2} \\ &= \frac{\|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2 \cdot (\mathbf{a} \times \mathbf{b}) - \|\mathbf{a} \times \mathbf{b}\|_2 \cdot (\tilde{\mathbf{a}} \times \tilde{\mathbf{b}})}{\|\mathbf{a} \times \mathbf{b}\|_2 \cdot \|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2} \\ &= \frac{\|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2 \cdot ((\mathbf{a} \times \mathbf{b}) - (\tilde{\mathbf{a}} \times \tilde{\mathbf{b}})) + (\|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2 - \|\mathbf{a} \times \mathbf{b}\|_2)(\tilde{\mathbf{a}} \times \tilde{\mathbf{b}})}{\|\mathbf{a} \times \mathbf{b}\|_2 \cdot \|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2}. \end{aligned}$$

Somit folgt mit  $\|\mathbf{c}\| \leq \|\mathbf{c}\|_2$ ,  $\mathbf{c} \in \mathbb{R}^3$ , sowie der Dreiecksungleichung

$$\begin{aligned} & \left\| \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a} \times \mathbf{b}\|_2} - \frac{\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}}{\|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2} \right\| \\ & \leq \frac{\|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2 \cdot \|(\mathbf{a} \times \mathbf{b}) - (\tilde{\mathbf{a}} \times \tilde{\mathbf{b}})\| + (\|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2 - \|\mathbf{a} \times \mathbf{b}\|_2) \|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|}{\|\mathbf{a} \times \mathbf{b}\|_2 \|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\|_2} \quad (3.26) \\ & \leq 2 \frac{\|(\mathbf{a} \times \mathbf{b}) - (\tilde{\mathbf{a}} \times \tilde{\mathbf{b}})\|_2}{\|\mathbf{a} \times \mathbf{b}\|_2}. \end{aligned}$$

Wegen  $\|\mathbf{c}\|_2 \leq \sqrt{3} \|\mathbf{c}\|$ ,  $\mathbf{c} \in \mathbb{R}^3$ , erhält man weiterhin

$$\begin{aligned} \|(\mathbf{a} \times \mathbf{b}) - (\tilde{\mathbf{a}} \times \tilde{\mathbf{b}})\|_2 & \leq \|\mathbf{a} \times (\mathbf{b} - \tilde{\mathbf{b}})\|_2 + \|(\mathbf{a} - \tilde{\mathbf{a}}) \times \tilde{\mathbf{b}}\|_2 \\ & \leq \|\mathbf{a}\|_2 \cdot \|\mathbf{b} - \tilde{\mathbf{b}}\|_2 + \|\mathbf{a} - \tilde{\mathbf{a}}\|_2 \cdot \|\tilde{\mathbf{b}}\|_2 \quad (3.27) \\ & \leq 3 (\|\mathbf{a}\| \cdot \|\mathbf{b} - \tilde{\mathbf{b}}\| + \|\mathbf{a} - \tilde{\mathbf{a}}\| \cdot \|\tilde{\mathbf{b}}\|). \end{aligned}$$

Lemma 3.14 folgt unmittelbar aus den Ungleichungen (3.26) und (3.27).  $\square$

Die optimale Approximationsordnung des Verfahrens lässt sich zeigen, indem man die Abweichung der Bézier-Bernstein-Koeffizienten des stetigen Interpolanten  $\tilde{\mathbf{s}}$  von jenen des geglätteten Splines  $\mathbf{s}$  abschätzt:

**Lemma 3.15**

Es existiert eine lediglich von  $f$  und dem kleinsten Winkel  $\alpha$  in  $\Delta$  abhängende Konstante  $C > 0$ , sodass für die Bézier-Bernstein-Koeffizienten  $\mathbf{c}_{ijk}^{[T]} - \tilde{\mathbf{c}}_{ijk}^{[T]}$ ,  $i + j + k = 3$ ,  $T \in \diamond$ , von  $\mathbf{s} - \tilde{\mathbf{s}}$  die folgende Abschätzung gilt:

$$\left\| \mathbf{c}_{ijk}^{[T]} - \tilde{\mathbf{c}}_{ijk}^{[T]} \right\| \leq C h^4, \quad i + j + k = 3, \quad T \in \diamond \quad (3.28)$$

Dabei bezeichnet  $h$  den maximalen Umkreis der Dreiecke in  $\Delta$ .

**Beweis:** Wir schätzen zunächst die Koeffizienten  $\mathbf{c}_{ijk}^{[T]} - \tilde{\mathbf{c}}_{ijk}^{[T]}$  mit maximalem Abstand eins zu den Eckpunkten  $v$  von  $\diamond$  ab.

Dabei bleiben die zu den Eckpunkten aus  $\diamond$  gehörenden Koeffizienten unverändert. Weiterhin bleiben alle Koeffizienten aus  $\diamond$  mit Abstand eins zu Eckpunkten aus  $\diamond$  unverändert, welche nur einem einzigen Dreieck angehören.

Laut Algorithmus 3.1 werden die verbleibenden Koeffizienten mit Abstand eins zu Eckpunkten von  $\diamond$  wie folgt modifiziert: Zunächst werden die Normalenvektoren  $\mathbf{n}_l$  der Splinestücke  $\tilde{\mathbf{s}}|_{T_l}$  auf den Dreiecken  $T_l$ ,  $l = 1, \dots, N$ , mit Eckpunkt  $v$  in diesem Punkt gemittelt, d. h. man berechnet

$$\mathbf{n}_l = \frac{(\tilde{\mathbf{c}}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]}) \times (\tilde{\mathbf{c}}_{201}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]})}{\left\| (\tilde{\mathbf{c}}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]}) \times (\tilde{\mathbf{c}}_{201}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]}) \right\|}, \quad l = 1, \dots, N.$$

Im Anschluss mittelt man die Tangentialebenen:

$$\mathbf{n}(v) = \frac{1}{N} \sum_{l=1}^N \mathbf{n}_l$$

Schließlich werden alle Koeffizienten mit Abstand eins zu  $v$  in die gemittelte Ebene projiziert, d. h. man erhält

$$\mathbf{c}_{210}^{[T_l]} = \tilde{\mathbf{c}}_{210}^{[T_l]} - \left\langle \tilde{\mathbf{c}}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]}, \mathbf{n}(v) \right\rangle \mathbf{n}(v), \quad l = 1, \dots, N.$$

Mit  $\|\mathbf{n}(v)\|_2 \leq 1$  und der Dreiecksungleichung ergibt sich für jeden Koeffizienten  $\mathbf{c}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{210}^{[T_l]}$ ,  $l = 1, \dots, N$ , die folgende Abschätzung:

$$\begin{aligned} \left\| \mathbf{c}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{210}^{[T_l]} \right\|_2 &\leq \left| \left\langle \tilde{\mathbf{c}}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]}, \mathbf{n}(v) \right\rangle \right| \|\mathbf{n}(v)\|_2 \\ &\leq \left| \left\langle \tilde{\mathbf{c}}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]}, \mathbf{n}(v) - \mathbf{n}(\tilde{\mathbf{s}}|_{T_l})(v) \right\rangle \right| \\ &\quad + \left| \left\langle \tilde{\mathbf{c}}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]}, \mathbf{n}(\tilde{\mathbf{s}}|_{T_l})(v) \right\rangle \right| \end{aligned} \quad (3.29)$$

Mit Hilfe von Satz 1.8 (Anwendung des Algorithmus von de Casteljau) lässt sich  $\tilde{\mathbf{c}}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]}$  wie folgt umformen:

$$\tilde{\mathbf{c}}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]} = \frac{1}{3} \left\| \mathcal{P}_{030}^{[T_l]} - v \right\|_2 (r_x(\tilde{\mathbf{s}}|_{T_l})_x(v) + r_y(\tilde{\mathbf{s}}|_{T_l})_y(v)) \quad (3.30)$$

Hierbei ist  $\mathbf{r} = (r_x, r_y) = (\mathcal{P}_{030}^{[T_l]} - v) / (\|\mathcal{P}_{030}^{[T_l]} - v\|_2)$  der Einheitsvektor in Richtung der Kante  $[\mathcal{P}_{030}^{[T_l]}, v]$ , auf welcher sich der abzuschätzende Koeffizient  $\mathbf{c}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{210}^{[T_l]}$  befindet. Daher folgt mit (3.29) und der Cauchy-Schwarz'schen Ungleichung

$$\left\| \mathbf{c}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{210}^{[T_l]} \right\|_2 \leq \left\| \tilde{\mathbf{c}}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]} \right\|_2 \cdot \left\| \mathbf{n}(v) - \mathbf{n}(\tilde{\mathbf{s}}|_{T_l})(v) \right\|_2. \quad (3.31)$$

Wir verwenden an dieser Stelle Abschätzung (3.44) aus dem Beweis zu Theorem 3.16. Mit Darstellung (3.30) folgt:

$$\begin{aligned} \left\| \mathbf{n}(v) - \mathbf{n}(\tilde{\mathbf{s}}|_{T_l})(v) \right\|_2 &\leq \frac{1}{3} (\|r_x(\tilde{\mathbf{s}}|_{T_l})_x(v)\|_2 + \|r_y(\tilde{\mathbf{s}}|_{T_l})_y(v)\|_2) h \\ &\leq \frac{1}{\sqrt{3}} (\|\mathbf{f}_x\| + \|\mathbf{f}_y\| + 2L h^3) h \end{aligned} \quad (3.32)$$

Gleichung (3.30) lässt sich auf die Darstellung von  $\tilde{\mathbf{c}}_{201}^{[T_l]} - \tilde{\mathbf{c}}_{300}^{[T_l]}$  übertragen, indem Satz 1.8 analog angewandt wird. Wir verwenden diese Zusammenhänge, weiterhin Gleichung (3.47) aus dem Beweis zu Theorem 3.16 sowie die Dreiecksungleichung für folgende Abschätzung:

$$\begin{aligned} &\left\| \mathbf{n}(v) - \mathbf{n}(\tilde{\mathbf{s}}|_{T_l})(v) \right\|_2 \\ &= \left\| \frac{1}{m} \sum_{\mu=1}^N (\mathbf{n}(\tilde{\mathbf{s}}|_{T_\mu})(v) - \mathbf{n}(\tilde{\mathbf{s}}|_{T_l})(v)) \right\|_2 \\ &\leq \frac{1}{m} \sum_{\mu=1}^N (\|\mathbf{n}(\tilde{\mathbf{s}}|_{T_\mu})(v) - \mathbf{n}(\mathbf{f})(v)\|_2 + \|\mathbf{n}(\mathbf{f})(v) - \mathbf{n}(\tilde{\mathbf{s}}|_{T_l})(v)\|_2) \\ &\leq 2\sqrt{3} C_0 h^3. \end{aligned} \quad (3.33)$$



Nun folgt für alle  $l = 1, \dots, N$  mit (3.31) und (3.32)

$$\left\| \mathbf{c}_{210}^{[T_l]} - \tilde{\mathbf{c}}_{210}^{[T_l]} \right\| \leq 2 C_0 (\|\mathbf{f}_x\| + \|\mathbf{f}_y\| + 2Lh^3) h^4. \quad (3.34)$$

Offensichtlich gilt eine analoge Abschätzung für die Koeffizienten  $\mathbf{c}_{201}^{[T_l]} - \tilde{\mathbf{c}}_{201}^{[T_l]}$ ,  $l = 1, \dots, N$ .

Sei  $\hat{\mathbf{s}} : \Omega \rightarrow \mathbb{R}^3$  derjenige Spline, welcher sich nach Beendigung von Schritt 1 der Algorithmen 3.11 bzw. 3.12 ergibt. Aufgrund obiger Aussagen erfüllt  $\hat{\mathbf{s}}$  für alle  $\omega \in \{0, \dots, 3\}$  eine Abschätzung wie in (3.25), wobei die entsprechende Konstante  $L_0$  lediglich von  $\mathbf{f}$  sowie vom kleinsten Winkel  $\alpha$  von  $\Delta$  abhängt. Mit ähnlichen Argumenten folgt die Regularität von  $\hat{\mathbf{s}}$ , d. h. für alle  $z \in \Omega$  ist  $\hat{\mathbf{s}}_x(z) \times \hat{\mathbf{s}}_y(z) \neq 0$ .

Wir fahren mit der Abschätzung derjenigen Bézier-Bernstein-Koeffizienten  $\mathbf{c}_{111}^{[T_l]} - \tilde{\mathbf{c}}_{111}^{[T_l]}$  des Splines  $\mathbf{s} - \tilde{\mathbf{s}}$  fort, welche bei der Glättung modifiziert werden. Wir behandeln dabei den Fall, dass  $\hat{\mathbf{s}}$  über einer Kante  $[u, v] \in \mathcal{V}^{[\Delta]}$ , welche zwei Dreiecken  $T_l = [u, v, w_l] \in \mathcal{D}$ ,  $l = 1, 2$ , angehört, geglättet wird, sodass beide Koeffizienten  $\mathbf{c}_{111}^{[T_l]}$ ,  $l = 1, 2$ , modifiziert werden. Für die Situation, dass der Interpolant auf einem Dreieck bezüglich des benachbarten Dreiecks geglättet wird, gelten ähnliche Argumente.

Wir verwenden die Notationen aus Algorithmus 3.3, setzen dabei jedoch  $\hat{\mathbf{s}}$  bzw.  $\hat{\mathbf{p}}_l$ ,  $l = 1, 2$ , für den ursprünglichen sowie  $\mathbf{s}$  bzw.  $\mathbf{p}_l$ ,  $l = 1, 2$ , für den über  $[u, v]$  geglätteten Spline. Die Cauchy-Schwarz'sche Ungleichung und die Dreiecksungleichung implizieren

$$\bar{\alpha}_l \leq \|w_l - u\|_2 + |\bar{\beta}_l| \|v - u\|_2 \leq 2 \|w_l - u\|_2. \quad (3.35)$$

Mit der Übereinstimmung  $\hat{\mathbf{p}}_1|_{[u,v]} \equiv \hat{\mathbf{p}}_2|_{[u,v]} \equiv \mathbf{p}_1|_{[u,v]} \equiv \mathbf{p}_2|_{[u,v]}$  ist insbesondere

$$\hat{\mathbf{p}}_1(m) = \hat{\mathbf{p}}_2(m) = \mathbf{p}_1(m) = \mathbf{p}_2(m). \quad (3.36)$$

Ferner erhalten wir für  $l = 1, 2$

$$\begin{aligned} & (\mathbf{p}_l)_{\mathbf{r}_l^+}(m) - (\hat{\mathbf{p}}_l)_{\mathbf{r}_l^+}(m) \\ &= (\mathbf{p}_l)_{\bar{\mathbf{r}}_l}(m) - (\hat{\mathbf{p}}_l)_{\bar{\mathbf{r}}_l}(m) + \frac{\langle \mathbf{s}_r(m), (\hat{\mathbf{p}}_l)_{\mathbf{r}_l^+}(m) \rangle}{\|\mathbf{s}_r(m)\|_2^2} \cdot (\mathbf{s}_r(m) - \hat{\mathbf{s}}_r(m)) \\ &= (\mathbf{p}_l)_{\bar{\mathbf{r}}_l}(m) - (\hat{\mathbf{p}}_l)_{\bar{\mathbf{r}}_l}(m). \end{aligned} \quad (3.37)$$

Aus der Definition von  $\bar{\mathbf{r}}_l$ ,  $l = 1, 2$ , und (3.36), folgt weiterhin

$$\begin{aligned} (\hat{\mathbf{p}}_2)_{\bar{\mathbf{r}}_1}(m) - (\hat{\mathbf{p}}_2)_{\bar{\mathbf{r}}_2}(m) &= -\frac{\langle \mathbf{s}_r(m), (\hat{\mathbf{p}}_2)_{\mathbf{r}_2^+}(m) \rangle}{\|\mathbf{s}_r(m)\|_2^2} \hat{\mathbf{s}}_r(m) + \frac{\langle \mathbf{s}_r(m), (\hat{\mathbf{p}}_1)_{\mathbf{r}_1^+}(m) \rangle}{\|\mathbf{s}_r(m)\|_2^2} \hat{\mathbf{s}}_r(m) \\ &= \frac{\langle \mathbf{s}_r(m), (\hat{\mathbf{p}}_1)_{\mathbf{r}_1^+}(m) - (\hat{\mathbf{p}}_2)_{\mathbf{r}_2^+}(m) \rangle}{\|\mathbf{s}_r(m)\|_2^2} \mathbf{s}_r(m). \end{aligned}$$

Daher liefert die Cauchy-Schwarz'sche Ungleichung

$$\|(\hat{\mathbf{p}}_2)_{\bar{\mathbf{r}}_1}(m) - (\hat{\mathbf{p}}_2)_{\bar{\mathbf{r}}_2}(m)\|_2 \leq \left\| (\hat{\mathbf{p}}_2)_{\mathbf{r}_2^\perp}(m) - (\hat{\mathbf{p}}_1)_{\mathbf{r}_1^\perp}(m) \right\|_2. \quad (3.38)$$

Da für  $l = 1, 2$  die Mengen aus (3.17) Orthogonalsysteme sind, erhalten wir die Gleichheit

$$\|\mathbf{n}_l(m) \times \mathbf{s}_r(m)\|_2 = \|\mathbf{n}_l(m)\|_2 \cdot \|\mathbf{s}_r(m)\|_2 = \|(\mathbf{p}_l)_{\bar{\mathbf{r}}_l}(m)\|_2, \quad l = 1, 2.$$

Mit (3.15) folgt weiter

$$(\mathbf{p}_l)_{\bar{\mathbf{r}}_l}(m) = \mathbf{n}_l(m) \times \mathbf{s}_r(m), \quad l = 1, 2. \quad (3.39)$$

Mit der Dreiecksungleichung, (3.38) und den approximativen Eigenschaften von  $\hat{\mathbf{s}}$  erhalten wir

$$\begin{aligned} \|\mathbf{n}_1(m) - \mathbf{n}_2(m)\| &= \left\| \frac{((\mathbf{p}_1)_{\bar{\mathbf{r}}_1}(m) - (\mathbf{p}_2)_{\bar{\mathbf{r}}_2}(m)) \times \mathbf{s}_r(m)}{\|\mathbf{s}_r(m)\|_2^2} \right\| \\ &\leq \frac{\|((\mathbf{p}_1)_{\bar{\mathbf{r}}_1}(m) - (\mathbf{p}_2)_{\bar{\mathbf{r}}_2}(m))\|_2 \cdot \|\mathbf{s}_r(m)\|_2}{\|\mathbf{s}_r(m)\|_2^2} \\ &\leq \frac{\|((\mathbf{p}_1)_{\bar{\mathbf{r}}_1}(m) - (\mathbf{p}_2)_{\bar{\mathbf{r}}_1}(m))\|_2 + \|(\mathbf{p}_2)_{\bar{\mathbf{r}}_1}(m) - (\mathbf{p}_2)_{\bar{\mathbf{r}}_2}(m)\|_2}{\|\mathbf{s}_r(m)\|_2} \\ &\leq \sqrt{3} \frac{\|((\mathbf{p}_1)_{\bar{\mathbf{r}}_1}(m) - (\mathbf{p}_2)_{\bar{\mathbf{r}}_1}(m))\| + \|(\hat{\mathbf{p}}_2)_{\mathbf{r}_2^\perp}(m) - (\hat{\mathbf{p}}_1)_{\mathbf{r}_1^\perp}(m)\|}{\|\mathbf{s}_r(m)\|_2} \\ &\leq \frac{2\sqrt{3} L_0}{\|\mathbf{s}_r(m)\|_2} h^3. \end{aligned} \quad (3.40)$$

Mit den Gleichungen (3.37) sowie (3.39) und den Definitionen von  $\mathbf{s}_{\bar{r}}(m)$  sowie  $\mathbf{n}(m)$  (vgl. (3.18)) folgt die Abschätzung

$$\begin{aligned} \mathbf{c}_{111}^{[T_l]} - \tilde{\mathbf{c}}_{111}^{[T_l]} &= \frac{2}{3} \bar{\alpha}_l ((\mathbf{p}_l)_{\mathbf{r}_l^\perp}(m) - (\hat{\mathbf{p}}_l)_{\mathbf{r}_l^\perp}(m)) \\ &= \frac{2}{3} \bar{\alpha}_l ((\mathbf{p}_l)_{\bar{\mathbf{r}}_l}(m) - (\hat{\mathbf{p}}_l)_{\bar{\mathbf{r}}_l}(m)) \\ &= \frac{2}{3} \bar{\alpha}_l (\mathbf{n}(m) \times \mathbf{s}_r(m) - \mathbf{n}_l(m) \times \mathbf{s}_r(m)) \\ &= \frac{1}{3} \bar{\alpha}_l ((\mathbf{n}_2(m) - \mathbf{n}_1(m)) \times \mathbf{s}_r(m)), \quad l = 1, 2. \end{aligned} \quad (3.41)$$

Gemeinsam mit den Abschätzungen (3.35) und (3.40) folgt

$$\begin{aligned} \left\| \mathbf{c}_{111}^{[T_l]} - \tilde{\mathbf{c}}_{111}^{[T_l]} \right\| &\leq \frac{1}{\sqrt{3}} \bar{\alpha}_l \|\mathbf{n}_2(m) - \mathbf{n}_1(m)\| \cdot \|\mathbf{s}_r(m)\| \\ &\leq 4 L_0 h^4, \quad l = 1, 2. \end{aligned} \quad (3.42)$$

Wegen Theorem 3.13 folgt induktiv die Existenz einer lediglich von  $f$  und  $\alpha$  abhängenden Konstanten, welche Abschätzung (3.28) für alle Koeffizienten  $\mathbf{c}_{111}^{[T]} - \tilde{\mathbf{c}}_{111}^{[T]}$ ,  $T \in \diamond$ , erfüllt.

Es bleibt die Abschätzung von gewissen Koeffizienten  $\mathbf{c}_{ijk} - \tilde{\mathbf{c}}_{ijk}$  in Dreiecken  $T$ , welche durch Clough-Tochter-Splits unterteilt wurden und in Paaren  $\{T^\Delta, T^\nabla\}$  von Dreiecken, welche in vier Dreiecke unterteilt wurden. Für diese Koeffizienten gilt, aufgrund von Argumenten wie in [72] und [68], Abschätzung (3.28) mit einer geeigneten Konstanten  $C > 0$ .

Somit ist (3.28) erfüllt für sämtliche Koeffizienten  $\mathbf{c}_{ijk}^{[T]} - \tilde{\mathbf{c}}_{ijk}^{[T]}$ ,  $T \in \diamond$ ,  $i + j + k = 3$ . Dies beweist Lemma 3.15.  $\square$

Wir fahren fort mit dem Beweis der optimalen Approximationsordnung:

### Theorem 3.16

*Ausgehend von einer ebenen Triangulierung  $\Delta$  existiert für jeden wie in Abschnitt 3.4 berechneten Approximanten  $\mathbf{s} : \Omega \rightarrow \mathbb{R}^3$  einer Funktion  $\mathbf{f} = (f_1, f_2, f_3) : \Omega \rightarrow \mathbb{R}^3$  mit  $f_\tau \in C^4(\Omega)$ ,  $\tau = 1, 2, 3$ , eine geeignete, lediglich von  $\mathbf{f}$  sowie dem kleinsten Winkel  $\alpha$  von  $\Delta$  abhängenden Konstante  $K > 0$ , sodass für alle  $\omega \in \{0, \dots, 3\}$  die folgende Abschätzung erfüllt ist:*

$$\| \|D^\omega(\mathbf{f} - \mathbf{s})\| \| \leq K h^{4-\omega}.$$

Hierbei bezeichnet  $h$  den maximalen Umfang der Dreiecke aus  $\Delta$ .

**Beweis:** Im ersten algorithmischen Schritt wird ein stetiger Spline  $\tilde{\mathbf{s}} : \Omega \rightarrow \mathbb{R}^3$  in drei Komponenten  $\tilde{s}_\tau$ ,  $\tau = 1, 2, 3$ , bestimmt, welcher die folgenden Interpolationsbedingungen erfüllt:

$$\tilde{s}_\tau(z) = f_\tau(z), \quad \tau = 1, 2, 3, \quad z \in \mathcal{P}_3^2(\Delta)$$

Aus der multivariaten Approximationstheorie ist bekannt, dass Gleichung (3.25),  $\omega \in \{0, \dots, 3\}$ , mit einer geeigneten Konstanten  $L > 0$  für jedes Teilpolynom  $\tilde{s}_\tau|_T$ ,  $\tau = 1, 2, 3$ ,  $T \in \Delta$ , erfüllt ist, vgl. [13]. Für die Richtungsableitungen erster Ordnung folgt insbesondere

$$\| \mathbf{f}_o - \tilde{\mathbf{s}}_o \| \leq L h^3, \quad o \in \{x, y\}. \quad (3.43)$$

Angewandt auf die einzelnen Punkte aus  $\Omega$  ergibt sich

$$\| \tilde{\mathbf{s}}_o(z) \| \leq \| \tilde{\mathbf{s}}_o \| \leq L h^3 + \| \mathbf{f}_o \|, \quad z \in \Omega, \quad o \in \{x, y\}. \quad (3.44)$$

Mit den Voraussetzungen für  $\mathbf{f}$  erhält man die Existenz eines Minimums

$$c := \min_{z \in \Omega} \| \mathbf{f}_x(z) \times \mathbf{f}_y(z) \|_2 > 0. \quad (3.45)$$

Für die im Beweis folgenden Argumente ist es nötig, zunächst die Wohldefiniertheit der Normalenvektoren  $\tilde{\mathbf{s}}_x(z) \times \tilde{\mathbf{s}}_y(z)$ ,  $z \in T$ ,  $T \in \Delta$ , aller Polynomstücke  $\tilde{\mathbf{s}}|_T$ ,  $T \in \Delta$ , zu zeigen. Seien also  $T \in \Delta$  sowie  $z \in T$ , und wir zeigen  $\tilde{\mathbf{s}}_x(z) \times \tilde{\mathbf{s}}_y(z) \neq (0, 0, 0)$ . Mit  $\mathbf{a} = \mathbf{f}_x(z)$ ,  $\tilde{\mathbf{a}} = \tilde{\mathbf{s}}_x(z)$ ,  $\mathbf{b} = \mathbf{f}_y(z)$ ,  $\tilde{\mathbf{b}} = \tilde{\mathbf{s}}_y(z)$  liefert Abschätzung (3.27)

$$\begin{aligned} & \|(\mathbf{f}_x(z) \times \mathbf{f}_y(z)) - (\tilde{\mathbf{s}}_x(z) \times \tilde{\mathbf{s}}_y(z))\|_2 \\ & \leq 3 (\|\mathbf{f}_x(z)\| \cdot \|\mathbf{f}_y(z) - \tilde{\mathbf{s}}_y(z)\| + \|\tilde{\mathbf{s}}_y(z)\| \cdot \|\mathbf{f}_x(z) \times \tilde{\mathbf{s}}_x(z)\|). \end{aligned}$$

Somit folgt aus der Dreiecksungleichung, den Abschätzungen (3.43) und (3.44) sowie Definition (3.45) für ausreichend kleine  $h$

$$\begin{aligned} \|\tilde{\mathbf{s}}_x(z) \times \tilde{\mathbf{s}}_y(z)\|_2 & \geq \|\mathbf{f}_x(z) \times \mathbf{f}_y(z)\|_2 - \|(\mathbf{f}_x(z) \times \mathbf{f}_y(z)) - (\tilde{\mathbf{s}}_x(z) \times \tilde{\mathbf{s}}_y(z))\|_2 \\ & \geq c - 3L (\|\mathbf{f}_x\| + \|\mathbf{f}_y\| + L h^3) h^3 \\ & \geq \frac{c}{2}. \end{aligned} \tag{3.46}$$

Insbesondere folgt  $\tilde{\mathbf{s}}_x(z) \times \tilde{\mathbf{s}}_y(z) \neq (0, 0, 0)$ . Ebenso folgt mit Ungleichung (3.27) und der Definition der Normalenvektoren  $\mathbf{n}(o)(z)$ ,  $o \in \{\mathbf{f}, \tilde{\mathbf{s}}\}$ , die folgende Abschätzung:

$$\|\mathbf{n}(\mathbf{f})(z) - \mathbf{n}(\tilde{\mathbf{s}})(z)\| \leq \frac{6L}{c} (\|\mathbf{f}_x\| + \|\mathbf{f}_y\| + L h^3) h^3$$

Man erhält

$$\|\mathbf{n}(\mathbf{f})(z) - \mathbf{n}(\tilde{\mathbf{s}})(z)\| \leq C_0 h^3 \tag{3.47}$$

mit einer lediglich von  $\mathbf{f}$  und dem kleinsten Winkel  $\alpha$  aus  $\Delta$  abhängenden Konstanten  $C_0 > 0$ .

Schließlich lässt sich die optimale Approximationsordnung des Verfahrens zeigen, indem die in Lemma 3.15 angegebene Abschätzung für die Bézier-Bernstein-Koeffizienten der Funktion  $\mathbf{s} - \tilde{\mathbf{s}}$  verwendet wird. Wie in Kapitel 1 dargestellt, nehmen bivariate Bernsteinpolynome  $B_{ijk}^3$ ,  $i + j + k = 3$ , über einem Dreieck  $T$  ausschließlich Werte  $\geq 0$  an und summieren sich zur Abbildung  $1_T : T \rightarrow \{1\}$  auf. Daher erhalten wir für jedes Dreieck  $T \in \diamond$  mit der Dreiecksungleichung und Abschätzung (3.28)

$$\begin{aligned} \|\mathbf{s}|_T - \tilde{\mathbf{s}}|_T\| & = \left\| \sum_{i+j+k=3} B_{ijk}^3|_T (\mathbf{c}_{ijk}^{[T]} - \tilde{\mathbf{c}}_{ijk}^{[T]}) \right\| \\ & = \sum_{i+j+k=3} B_{ijk}^3|_T \left\| \mathbf{c}_{ijk}^{[T]} - \tilde{\mathbf{c}}_{ijk}^{[T]} \right\| \\ & = C h^4. \end{aligned}$$

Da wie gesehen eine lediglich von  $\alpha$  und  $\mathbf{f}$  abhängende Konstante  $L > 0$  existiert, welche (3.25) für den stetigen Interpolanten  $\tilde{\mathbf{s}}$  erfüllt, folgt schließlich:

$$\|\mathbf{f} - \mathbf{s}\| \leq \|\mathbf{f} - \tilde{\mathbf{s}}\| + \|\mathbf{s} - \tilde{\mathbf{s}}\| \leq (L + C) h^4$$

Wir verwenden diese Aussage nun für eine *weak-interpolation*-Argumentation wie in [72], und zwar für jede einzelne Komponente  $f_\tau - s_\tau$ ,  $\tau = 1, 2, 3$ , von  $\mathbf{f} - \mathbf{s}$ . Hiermit folgt Theorem 3.16.  $\square$

### 3.5.3 Datenaufwand der geglätteten Interpolanten

Bei der Anwendung der Interpolationsmethode entspricht der Datenaufwand für einen geglätteten Interpolanten der Menge sämtlicher Bézier-Bernstein-Koeffizienten  $\mathbf{c}_{ijk}^{[T]}$ ,  $i + j + k = 3$ ,  $T \in \diamond$ .

Mit den Ergebnissen aus Abschnitt 2.5 ergibt sich für die Anzahl der Mikrodreiecke im Schnitt

$$|\diamond| \approx 1\frac{1}{2} |\Delta|$$

für die in Kapitel 2 entwickelten Färbungs- und Prioritätsprinzipien und  $|\diamond| \approx 1\frac{2}{3} |\Delta|$  für Färbungen wie in [68]. Dabei gehen wir davon aus, dass die in  $\diamond$  einzeln verbleibenden Dreiecke und die in Algorithmus 3.9 zu färbenden Dreieckspaare mit vier gefärbten Nachbarn wegen ihres sehr geringen Anteils nicht von Bedeutung sind.

Damit ist der Datenaufwand für die geglätteten Interpolanten bei allen Färbungs- und Prioritätsprinzipien wesentlich geringer als bei der Methode in [47], bei welcher sich  $|\Delta| \approx 2 |\Delta|$  für die Triangulierung  $\Delta$  des geglätteten Interpolanten ergibt.

## Kapitel 4

# 3D-Splineoberflächen auf projizierten Triangulierungen

Es ist ein bislang in der Literatur ungelöstes Problem, mit differenzierbaren 3D-Splineoberflächen zu interpolieren. Grob gesprochen können benachbarte Polynomstücke über einer gemeinsamen Kante nur dann differenzierbar zusammengesetzt werden, wenn die zugehörigen Dreiecke in einer Ebene liegen.

In diesem Kapitel entwickeln wir erstmals ein Verfahren, bei welchem die interpolierenden Splineoberflächen auf allen Eckpunkten und fast allen Kanten differenzierbar sind. Auf den wenigen verbleibenden Kanten erhält man Differenzierbarkeit für drei Punkte. Dabei werden, grob gesprochen, möglichst große Teilbereiche der entsprechenden 3D-Triangulierung in geeignete Ebenen projiziert. Die interpolierenden Splines sind lokal bestimmbar. Für ebene Triangulierungen ist es uns möglich, Ergebnisse für die Approximationsordnung zu geben: Ähnlich wie in [48] ergibt sich die optimale Approximationsordnung der Methode.

Wir geben eine grobe Darstellung des Verfahrens: Zunächst wird ein stetiger kubischer Interpolant wie in Abschnitt 3.2 bestimmt. Anschließend unterteilen wir, basierend auf einer Schwarzweißfärbung der entsprechenden Triangulierung, etwa die Hälfte der Dreiecke durch Clough-Tochter-Splits.

In Abschnitt 4.1 zerlegen wir die zugrunde liegende 3D-Triangulierung in Teilbereiche, welche sich in bestimmte Ebenen projizieren lassen, ohne dass ein vorgegebener Winkel zwischen Dreiecken und Projektionsebenen überschritten wird. Wir entwickeln hierfür einen effizienten Algorithmus, welcher bereits für kleine Winkel von beispielsweise  $25^\circ$  sehr große projizierbare Bereiche liefert. Auf diese Weise werden die Dreiecke bei der Projektion nicht verzerrt. Dabei werden, ausgehend von Startdreiecken, Mengen benachbarter Dreiecke mit gering abweichenden Normalenvektoren in die Ebenen dieser Startdreiecke projiziert. Die Wahl geeigneter Startdreiecke spielt dabei eine entscheidende Rolle für die Größe der projizierbaren Bereiche. Wir

entwickeln einen effizienten Algorithmus, welcher in jedem Iterationsschritt dasjenige Startdreieck ermittelt, welches den größtmöglichen projizierbaren Bereich ergibt. Die ermittelte Zerlegung wird im Anschluss modifiziert, um, grob gesprochen, Bereiche mit Löchern und „nicht glatten“ Rändern zu vermeiden, siehe Abbildung 4.2 auf Seite 121 und Abbildung 4.10 auf Seite 130.

Des Weiteren stellen wir die umfangreiche, im Rahmen dieser Arbeit erstellte computertechnische Implementierung der Methode dar. Schließlich testen wir die entwickelten Algorithmen an verschiedenen 3D-Triangulierungen. Es zeigt sich, dass auch für komplexe 3D-Triangulierungen in effizienter Weise sehr große projizierbare Bereiche erzeugt werden. Damit sind die konstruierten Interpolanten auf fast allen Kanten differenzierbar. Der Anteil dieser Kanten lag bei den Tests bei durchschnittlich 90%.

Ausgehend von dieser Zerlegung glätten wir den Spline-Interpolanten zunächst über den Eckpunkten der Triangulierung. Auf den im Inneren der projizierten Bereiche liegenden Eckpunkten glätten wir komponentenweise ähnlich wie im bivariaten Fall. Mit den verbleibenden Eckpunkten  $\mathbf{v}$ , welche mehreren Bereichen angehören, verfahren wir wie folgt: Wir projizieren diejenigen Koeffizienten des  $C^1$ -Rings um  $\mathbf{v}$ , welche zwei Bereichen angehören, wie in Kapitel 3 in die gemittelte Tangentialebene. Für die restlichen Koeffizienten dieses  $C^1$ -Rings verfahren wir komponentenweise.

Schließlich glätten wir den Spline über den Kanten der Triangulierung. Die Reihenfolge der betrachteten Dreiecke richtet sich nach deren Färbung. Die Glättung über einer Kante benachbarter Dreiecke erfolgt dabei stets komponentenweise, falls beide Dreiecke derselben Teiltriangulierung angehören und wie in Kapitel 3, falls die Dreiecke unterschiedlichen Teiltriangulierungen angehören. Auf diese Weise erhält man lokal bestimmbare Interpolanten, welche auf den weitaus meisten Kanten und auf allen Eckpunkten der unterteilten Triangulierung differenzierbar sind. Auf den wenigen verbleibenden Kanten ergibt sich Differenzierbarkeit an den Eckpunkten und am Kantenmittelpunkt. Für Ausgangstriangulierungen der Ebene besitzt das Verfahren optimale Approximationsordnung.

## 4.1 Bestimmung projizierbarer Teiltriangulierungen

### 4.1.1 Zerlegung von 3D-Triangulierungen

Wir entwickeln im Folgenden einen effizienten Algorithmus für die Zerlegung von 3D-Triangulierungen in projizierbare Teilbereiche. Ausgehend von Startdreiecken werden benachbarte Dreiecke zusammengefasst, sodass ein festgelegter Winkel  $\alpha$  bei der Projektion in die Ebene des Startdreiecks nicht überschritten wird.

**Algorithmus 4.1 (Zerlegung von  $\Delta$ )**

Gegeben sei eine Triangulierung  $\Delta$  im Raum. Für  $l = 1, \dots, N$  konstruiere man wie folgt paarweise disjunkte Teilmengen  $\Delta_l$  von  $\Delta$ , solange  $\bigcup_{l=1}^N \Delta_l \neq \Delta$  ist. Dabei bezeichne man jedes hinzugefügte Dreieck als *gewählt*.

Zunächst füge man zu  $\Delta_l$  ein beliebiges noch nicht gewähltes Dreieck  $T_0^{[l]}$  hinzu und bezeichne es als *Startdreieck der Teiltriangulierung  $\Delta_l$*  bzw. als  $l$ -tes Startdreieck.

Im Anschluss füge man rekursiv jedes nicht gewählte Dreieck  $T$  mit Nachbarn in  $\Delta_l$  zu  $\Delta_l$  hinzu, solange die Normalenvektoren von  $T$  und  $T_0^{[l]}$  nur um einen Winkel  $\leq \alpha$  voneinander abweichen.

Es ist offensichtlich, dass diese Vorgehensweise in bezüglich  $|\Delta|$  linearer Komplexität eine Zerlegung  $\Delta_l$ ,  $l = 1, \dots, N$ , von  $\Delta$  liefert und dass sich jede Teiltriangulierung  $\Delta_l$  in die durch  $T_0^{[l]}$  aufgespannte Ebene  $E^{[l]}$  projizieren lässt, wobei bei der Projektion der Winkel  $\alpha$  nicht überschritten wird.

Die in Abschnitt 4.2 konstruierten geglätteten Spline-Interpolanten  $\mathbf{s}$  sind, grob gesprochen, auf jeder Kante von  $\Delta$  differenzierbar, welche nur in einer Teiltriangulierung der Zerlegung enthalten ist. Ziel ist daher ein möglichst hoher Anteil solcher Kanten.

Abbildung 4.1 zeigt eine einfache 3D-Triangulierung mit etwa 33.500 Dreiecken und die durch Algorithmus 4.1 ermittelten Teilbereiche. Das rechte Bild zeigt dabei das Ergebnis einer erweiterten Vorgehensweise, wie sie in Unterabschnitt 4.1.3 beschrieben wird. Als maximaler Winkel wurde 40 Grad gewählt.

In dem Beispiel wurden die Startdreiecke durch einen Zufallsgenerator ermittelt. Die im Rahmen dieser Arbeit implementierten Tests zeigen, dass bereits bei einer zufälligen Wahl von Startdreiecken große Bereiche konstruiert werden und sich ein hoher Anteil von Kanten ergibt, auf denen der geglättete Interpolant  $\mathbf{s}$  differenzierbar ist. Jedoch lässt sich dieser Anteil erhöhen, indem „geeignete“ Startdreiecke gewählt werden. In Abschnitt 4.1.3 entwickeln wir effiziente Algorithmen zur Ermittlung von Startdreiecken, sodass die resultierenden projizierbaren Bereiche möglichst groß werden.

**4.1.2 Geometrische Phänomene**

Theoretische Überlegungen und die im Rahmen dieser Arbeit erstellten computertechnischen Umsetzungen zeigen, dass Algorithmus 4.1 hervorragend geeignet ist, 3D-Triangulierungen verschiedenster Geometrie effizient in große projizierbare Bereiche zu zerlegen.

Bei der Anwendung des Verfahrens treten einige geometrische Besonderheiten auf, die wir in diesem Abschnitt beschreiben.



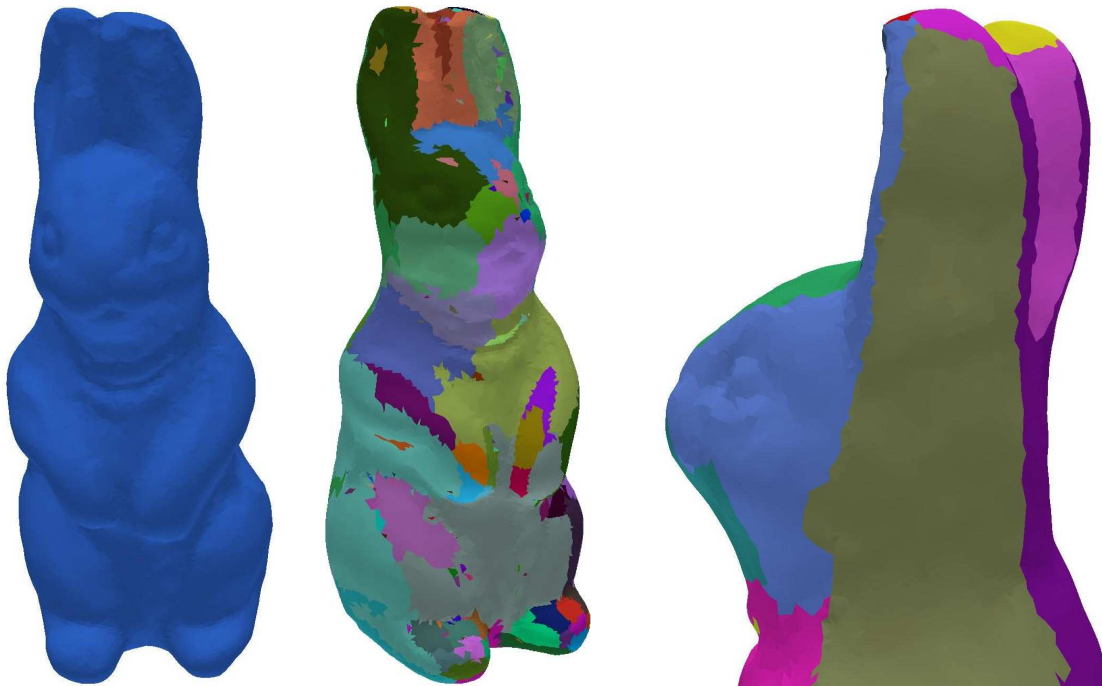


Abbildung 4.1: Anwendung von Algorithmus 4.1 ohne (Mitte) und mit (rechts) erweiterten Algorithmen

### Löcher und Zwischenbereiche

Bei den maximal projizierbaren Teilbereichen handelt es sich oftmals nicht um einfach zusammenhängende Gebiete  $\Omega_{\Delta_l}$ , sondern um Teiltriangulierungen mit *Löchern*. Als Loch einer Teiltriangulierung  $\Delta_l$ ,  $l = 1, \dots, N$ , bezeichnen wir eine 3D-Triangulierung  $M \subset \Delta \setminus \Delta_l$  mit einfach zusammenhängendem Gebiet, sodass jede Randkante von  $M$  zugleich Randkante von  $\Delta_l$  ist, für ein Beispiel siehe Abbildung 4.2.

Im Folgenden charakterisieren wir verschiedene Typen von Löchern und untersuchen deren Bedeutung hinsichtlich des Anteils der Kanten, auf denen  $s$  differenzierbar ist. Anschließend entwickeln wir ein effizientes algorithmisches Verfahren zur Auflösung von Löchern und erzielen damit eine Erhöhung dieses Anteils.

Wir unterscheiden folgende drei verschiedene Typen von Löchern, siehe Abbildung 4.3:

- Typ A:

Normalenpaare benachbarter Dreiecke  $T_1, T_2$  bilden in einem Bereich  $M$  von wenigen benachbarten Dreiecken bereits große Winkel, beispielsweise Win-



Abbildung 4.2: Beispiele für Löcher in einem projizierbaren Teilbereich

kel  $> \alpha$ , in der Nähe von  $M$  treten dagegen kleine Abweichungen auf. Im Allgemeinen verbleibt  $M$  als Loch in der jeweiligen Teiltriangulierung  $\Delta_l$ .

- Typ B:

Normalenpaare benachbarter Dreiecke bilden stets kleine Winkel. Es entsteht ein Loch  $M$  in  $\Delta_l$  dadurch, dass jedes Randdreieck in  $M$  eine Abweichung  $> \alpha$  zu  $T_0^{[l]}$  besitzt und im Inneren von  $M$  zudem Dreiecke mit erheblich größerer Normalenabweichung existieren.

- Typ C:

Normalenpaare benachbarter Dreiecke bilden stets kleine Winkel. Es entsteht ein Loch  $M$  in  $\Delta_l$  dadurch, dass Dreiecke aus  $M$  Abweichungen  $\approx \alpha$ ,  $> \alpha$  zu  $T_0^{[l]}$  besitzen.

Durch den großen Abweichungswinkel zwischen Normalen benachbarter Dreiecke lassen sich bei Löchern vom Typ A unabhängig von der algorithmischen Vorgehensweise keine großen Projektionsbereiche finden, denn stets wird die maximal vorgegebene Abweichung überschritten. Die Entstehung solcher Löcher ist bedingt

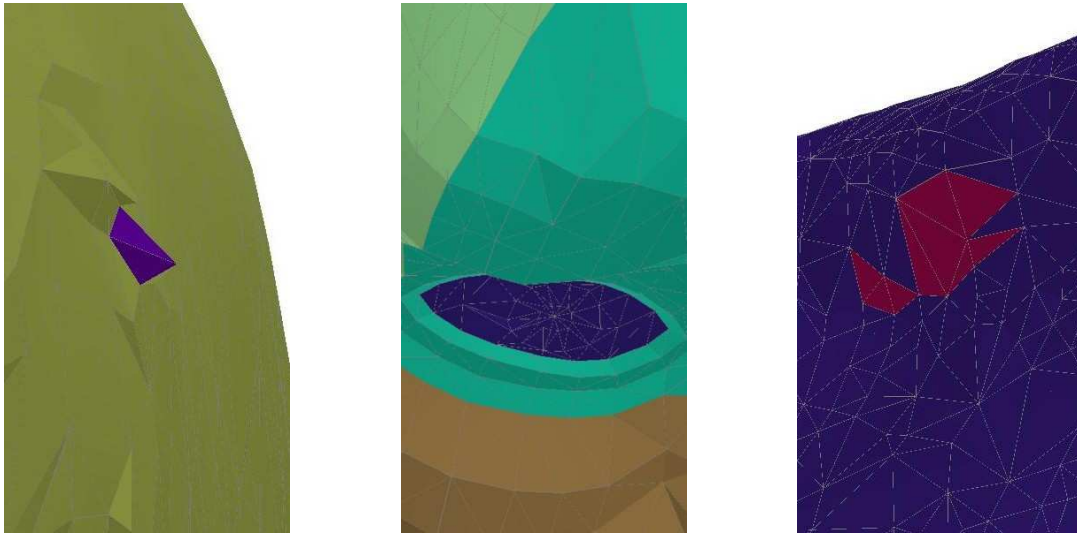


Abbildung 4.3: Loch vom Typ A (links): In  $\Delta$  existieren Paare benachbarter Dreiecke mit großer Abweichung zwischen den Normalenvektoren. Loch vom Typ B (Mitte): Benachbarte Dreiecke besitzen stets ähnliche Normalenvektoren. Die Entstehung des Lochs liegt dennoch in der Geometrie von  $\Delta$  begründet. Loch vom Typ C (rechts): Benachbarte Dreiecke besitzen stets ähnliche Normalenvektoren, und die Entstehung des Lochs liegt in der Vorgehensweise des Zerlegungsalgorithmus begründet.

durch die Geometrie der zugrunde liegenden 3D-Triangulierung, und eine Vermeidung bzw. Auflösung ist in effizienter Weise nicht möglich und auch nicht sinnvoll hinsichtlich des Anteils an Kanten, über denen  $s$  differenzierbar ist.

Ähnliches gilt für Löcher des Typs B: Zwischen Dreiecken aus  $M$  und der umgebenden Teiltriangulierungen treten große Winkel bei den zugehörigen Normalen auf, sodass eine Zerlegung unabhängig von der algorithmischen Umsetzung zu mehreren Bereichen führt. Somit liegt die Entstehung von  $M$  auch hier in der Geometrie von  $\Delta$  begründet, für Details siehe Abschnitt 4.1.3.

Anders ist die Situation für Löcher vom Typ C: Da zwischen  $M$  und den benachbarten Dreiecken keine großen Abweichungen zwischen den Winkeln der zugehörigen Normalen auftreten, ist die Entstehung solcher Löcher eine inhärente Eigenschaft von Algorithmus 4.1 und insbesondere in der Vorgehensweise begründet, projizierbare Bereiche von maximaler Größe zu konstruieren. Eine Verkleinerung der Bereiche zur Vermeidung von Löchern des Typs C ist jedoch nicht sinnvoll, da durch eine solche Vorgehensweise der Anteil der Kanten, auf denen  $s$  differenzierbar ist, erheblich kleiner wird. Jedoch ist es in effizienter Weise möglich, die meisten Löcher aufzulösen. Hierfür werden die Normalenabweichungen der enthaltenen Dreiecke zum Startdreieck der umgebenden Triangulierung  $\Delta_i$  überprüft. Falls dabei eine vorgegebene maximale Abweichung  $\beta > \alpha$  nicht überschritten wird, werden die Dreiecke im

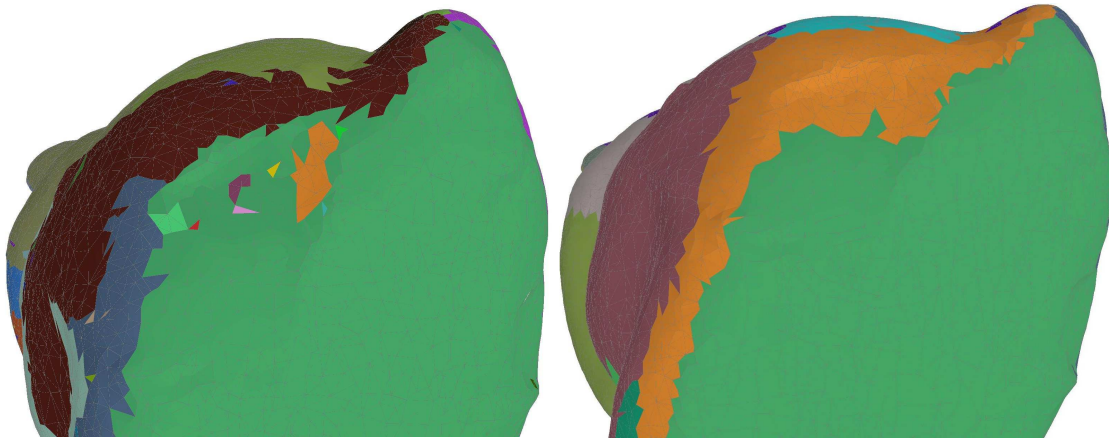


Abbildung 4.4: Ergebnis ohne und mit Verwendung erweiterter Teilalgorithmen,  $\alpha = 40^\circ$ ,  $\beta = 45^\circ$

Anschluss zu  $\Delta_l$  hinzugefügt. In Abschnitt 4.1.3 entwickeln wir effiziente Algorithmen, die auf diesem Ansatz basieren. Die Tests der Verfahren aus Abschnitt 4.1.4 zeigen, dass die Methode zu einer erheblichen Erhöhung des Anteils der Kanten, auf denen  $s$  differenzierbar ist, führt, wobei sich die maximale Abweichung von Normalenvektoren nur geringfügig erhöht. Abbildung 4.4 zeigt die Auflösung von Löchern an einem Beispiel.

In weniger häufigen Fällen führt die Struktur der 3D-Triangulierung dazu, dass zwischen zwei Teiltriangulierungen  $\Delta_{l_1}$  und  $\Delta_{l_2}$  kleine Bereiche übrig bleiben, siehe Abbildung 4.8 auf Seite 128. Solche Zwischenbereiche entstehen in ähnlichen Situationen, wie dies bei Löchern der Fall ist. Grob gesprochen liegt entweder ein kleiner Bereich mit bereits großen Normalenabweichungen vor, oder die Wahl der Startdreiecke  $T_0^{[l_1]}$  und  $T_0^{[l_2]}$  führt dazu, dass einige wenige, zwischen den Teiltriangulierungen liegende Dreiecke keiner dieser Triangulierungen zugeordnet werden kann. Im zweiten Fall ist eine Auflösung hinsichtlich des Anteils der Kanten, auf denen  $s$  differenzierbar ist, sinnvoll. Einen entsprechenden Algorithmus entwickeln wir in Abschnitt 4.1.3. Auch hier wird, ausgehend von einem zweiten Winkel  $\beta > \alpha$ , geprüft, welche Dreiecke sich bezüglich  $\beta$  den Bereichen  $\Delta_{l_1}$  und  $\Delta_{l_2}$  zuordnen lassen.

### Ränder von Teilbereichen

Um einen hohen Anteil von Kanten zu erhalten, auf denen  $s$  differenzierbar ist, ist von Vorteil, dass jede konstruierte Teiltriangulierung  $\Delta_l$  möglichst wenige Randkanten und möglichst viele innere Kanten besitzt. Insbesondere sollen die Ränder der projizierbaren Teilbereiche möglichst „glatt“ verlaufen, siehe Abbildung 4.5.

Je nach Geometrie der zu zerlegenden Triangulierung treten jedoch oftmals Teilbereiche auf, die im Verhältnis zu ihrer Größe viele Randkanten besitzen, siehe Ab-

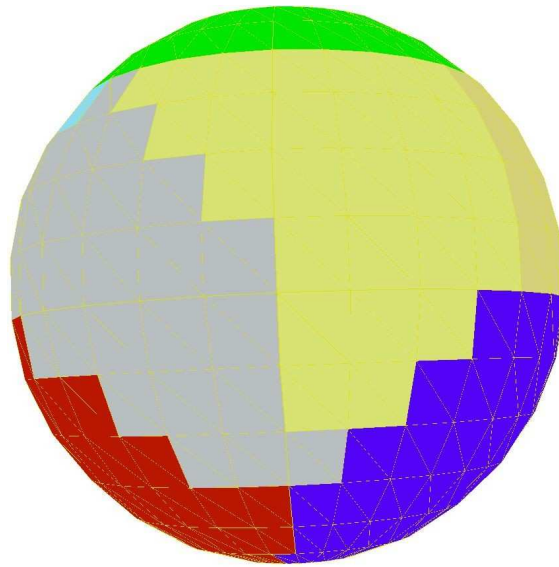


Abbildung 4.5: Teilbereiche mit vielen inneren und wenig Randkanten

bildung 4.6. Wir charakterisieren nun verschiedene Fälle, bei denen solche Bereiche auftreten und untersuchen deren Bedeutung hinsichtlich des Anteils der Kanten, auf denen  $s$  differenzierbar ist. Weiter unten entwickeln wir zwei spezielle Algorithmen zur Vermeidung von Randkanten und somit zur Erhöhung des Anteils der Kanten, auf denen  $s$  differenzierbar ist.

Wir unterscheiden folgende Typen von Bereichen mit vielen Randkanten:

- Typ A:

Die Geometrie der 3D-Triangulierung  $\Delta$  ist so, dass größtmögliche Bereiche, innerhalb derer ausschließlich Dreiecke mit paarweise geringer Normalenabweichung liegen, einen hohen Anteil von Randkanten besitzen, z. B. „langgezogen“ sind. Somit ergeben sich durch die Konstruktion von Bereichen maximaler Größe projizierbare Bereiche mit vielen Randkanten, ein Beispiel zeigt Abbildung 4.6 (links).

- Typ B:

Normalenpaare benachbarter Dreiecke bilden stets kleine Winkel. Es entsteht ein Bereich  $\Delta_l$  mit vielen Randkanten dadurch, dass möglichst große projizierbare Bereiche gewählt werden und alle Dreiecke mit Abweichung  $< \alpha, \approx \alpha$  noch hinzugefügt werden können, alle Dreiecke mit Abweichung  $> \alpha, \approx \alpha$  jedoch nicht mehr. Dies führt oftmals dazu, dass der zu  $\Delta_l$  gehörende Rand nicht „gerade“ verläuft sondern „zickzack“. Ein Beispiel zeigt Abbildung 4.6 (rechts).



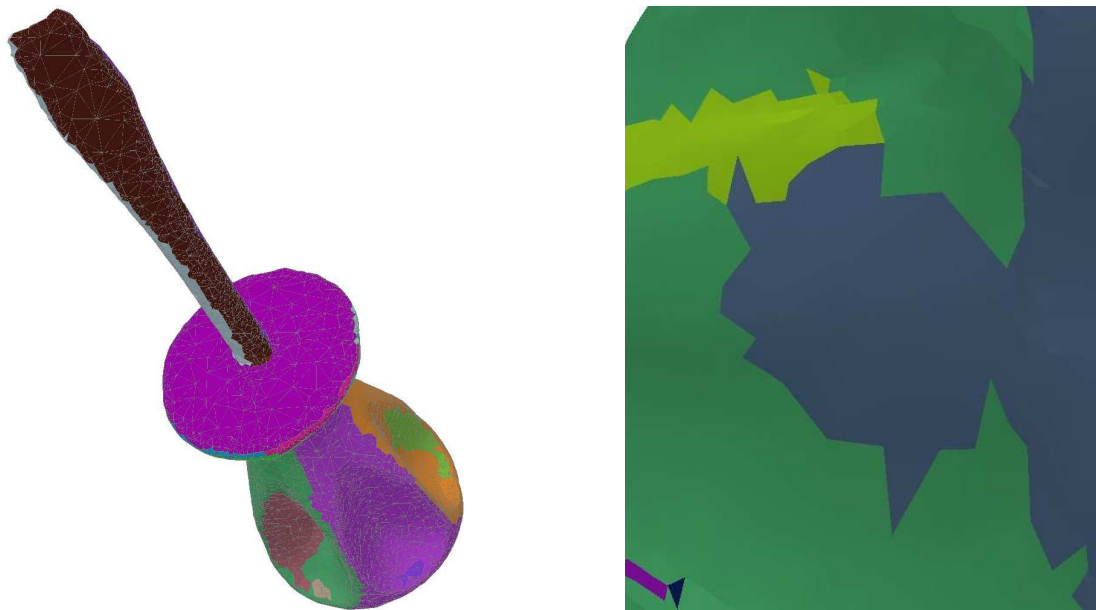


Abbildung 4.6: Bereiche mit vielen Randkanten, Typ A (links): Bereiche haben deshalb einen hohen Anteil an Randkanten, weil sie sich der Geometrie von  $\Delta$  anpassen. Typ B (rechts): Die Entstehung von vielen Randkanten liegt in der Vorgehensweise des Zerlegungsalgorithmus begründet.

Bei Bereichen vom Typ A ist die Entstehung vieler Randkanten bedingt durch die Oberflächenkrümmung von  $\Delta$ . Im Allgemeinen ist die Wahl von großen Bereichen mit vielen Randkanten hier hinsichtlich des Anteils der Kanten, auf denen  $\mathbf{s}$  differenzierbar ist, vorteilhaft gegenüber der Wahl von kleineren Bereichen mit einem geringeren Anteil von Randkanten. Betrachtet man beispielsweise Abbildung 4.6 (links), so wird deutlich, dass sich die Zerlegung gut der Struktur der Triangulierung anpasst und somit große projizierbare Bereiche liefert.

Anders ist die Situation für Bereiche vom Typ B: Da die Entstehung vieler Randkanten nicht notwendig ist für die Konstruktion großer projizierbarer Teiltriangulierungen, sondern lediglich eine inhärente Eigenschaft von Algorithmus 4.1 darstellt, ist es sinnvoll, eine Vorgehensweise zu entwickeln, welche eine Entstehung von Bereichen des Typs B verhindert oder die Bereiche im Nachhinein glättet. Solche Methoden entwickeln wir in Abschnitt 4.1.3, siehe auch Abbildung 4.10 auf Seite 130.

### 4.1.3 Erweiterte Teilalgorithmen

#### Behandlung von Löchern und Zwischenbereichen

Wie die Klassifizierung von Löchern aus Abschnitt 4.1.2 zeigt, ist es sinnvoll, möglichst viele Löcher vom Typ C aufzulösen und Löcher vom Typ A und B zu belassen.

Im Folgenden geben wir einen Algorithmus zur Auflösung von Löchern des Typs C:

##### Algorithmus 4.2

Sei  $\{\Delta_1, \dots, \Delta_N\}$  eine Zerlegung einer 3D-Triangulierung  $\Delta$ , die sich durch Anwendung von Algorithmus 4.1 für einen Winkel  $\alpha$  ergibt. Man wähle einen Winkel  $\beta > \alpha$  und gehe für jedes Loch  $M$  innerhalb jeder Teiltriangulierung  $\Delta_l$ ,  $l \in \{1, \dots, N\}$ , wie folgt vor: Man überprüfe die Normalenabweichung des Startdreiecks  $T_0^{[l]}$  zu jedem Dreieck aus  $M$ . Falls jede Abweichung  $\leq \beta$  ist, füge man alle Dreieck aus  $M$  zu  $\Delta_l$  hinzu.

Bei dieser Vorgehensweise gibt man einen zweiten Winkel  $\beta > \alpha$  vor, sodass für in Löchern  $M \subseteq \Delta$  liegende Dreiecke größere Winkel  $\leq \beta$  zum entsprechenden Startdreieck  $T_1^{[l]}$  des umgebenden Bereichs  $\Delta_l$  auftreten können. Man wählt dabei einen geringfügig größeren Winkel als  $\alpha$ , beispielsweise  $\alpha + 5^\circ$ . Es ist offensichtlich, dass dieser Algorithmus in effizienter Weise sämtliche Löcher vom Typ C auflöst, denn laut Voraussetzung ist für jedes Dreieck  $T \in M$  der Winkel zum entsprechenden Startdreieck  $\approx \alpha$  und damit  $< \beta$ .

Im Allgemeinen treten jedoch viele Löcher  $M$  auf, bei denen die Abweichung für einen Großteil  $M_1$  der Dreiecke  $\leq \beta$  ist und im Inneren Dreiecke  $M_2$  mit Abweichungswinkel  $> \beta$  verbleiben. Meist ist es hierbei sinnvoll, möglichst viele Dreiecke aus  $M_1$  zur umgebenden Teiltriangulierung  $\Delta_l$  hinzuzufügen, denn auf diese Weise erhöht sich der Anteil von Kanten, auf denen  $s$  differenzierbar ist.

Wir entwickeln daher folgenden Algorithmus zur Verkleinerung von Löchern:

##### Algorithmus 4.3

Sei  $\{\Delta_1, \dots, \Delta_N\}$  eine Zerlegung einer 3D-Triangulierung  $\Delta$ , die sich durch Anwendung von Algorithmus 4.1 für einen Winkel  $\alpha$  ergibt. Man wähle einen Winkel  $\beta > \alpha$  und gehe für jedes Loch  $M$  innerhalb jeder Teiltriangulierung  $\Delta_l$ ,  $l \in \{1, \dots, N\}$ , wie folgt vor: Solange Randdreiecke aus  $M$  mit einer Normalenabweichung  $\leq \beta$  zum Startdreieck  $T_0^{[l]}$  existieren, so füge man diese Randdreiecke zu  $\Delta_l$  hinzu.

Somit werden nicht nur alle Löcher vom Typ A geschlossen, sondern es wird weiterhin eine Verkleinerung vieler Löcher, insbesondere aller Löcher vom Typ B, erzielt.

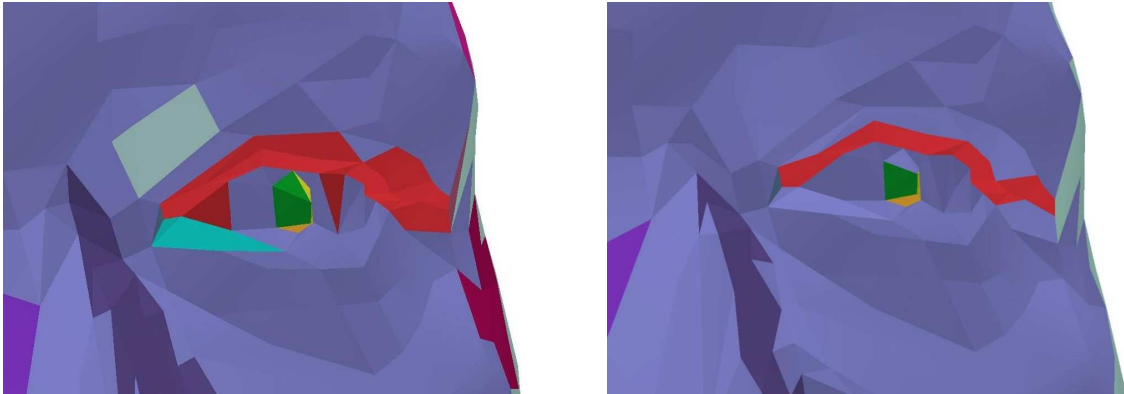


Abbildung 4.7: Verkleinerung eines Lochs durch Algorithmus 4.3

In den weitaus meisten Fällen wird durch jede Verkleinerung erreicht, dass der Anteil der Kanten, auf denen  $\mathbf{s}$  differenzierbar ist, steigt. Ein Beispiel für ein auf diese Weise verkleinertes Loch zeigt Abbildung 4.7 (rechts). Das linke Bild zeigt den entsprechenden Bereich von  $\Delta$  vor der Anwendung von Algorithmus 4.3.

Durch eine weitere Modifikation dieses Verfahrens lassen sich nicht nur Löcher schließen oder verkleinern, sondern auch viele Zwischenbereiche:

#### Algorithmus 4.4

Sei  $\{\Delta_1, \dots, \Delta_N\}$  eine durch Algorithmus 4.1 konstruierte Zerlegung einer 3D-Triangulierung, und man wähle einen Winkel  $\beta > \alpha$ . Für jede Teiltriangulierung  $\Delta_l$ ,  $l = 1, \dots, N$ , mit maximal  $n \in \mathbb{N}$  Dreiecken verfare man wie folgt: Solange ein Dreieck  $T \in \Delta_l$  existiert, welches ein Nachbardreieck einer anderen Teiltriangulierung  $\Delta_{\bar{l}}$  besitzt, sodass die Normalenabweichung zwischen  $T$  und  $T_0^{[\bar{l}]}$  maximal  $\beta$  beträgt, so füge man  $T$  zu  $\Delta_{\bar{l}}$  hinzu.

Auf diese Weise lässt sich ein großer Teil aller Zwischenbereiche auflösen, und nahezu alle Zwischenbereiche werden verkleinert. Ein Beispiel für die Anwendung dieses Algorithmus zeigt Abbildung 4.8. Es stellt sich die Frage der Wahl des Parameters  $n$ : Für kleine Werte, z. B. 30, können auch nur kleine Zwischenbereiche und Löcher aufgelöst bzw. verkleinert werden. Wählt man dagegen einen sehr großen Wert, z. B. 500, so tritt in seltenen Fällen das Phänomen auf, dass durch die Verkleinerung eines großen Zwischenbereichs ein oder mehrere neue, kleine Zwischenbereiche und somit Kanten entstehen, auf denen  $\mathbf{s}$  nicht differenzierbar ist. Eine alternative Vorgehensweise ist daher, Algorithmus 4.4 nur auf solche Bereiche anzuwenden, die sich vollständig auflösen lassen. Dies wiederum hat den erheblichen Nachteil, dass eine teilweise Auflösung von Löchern und Zwischenbereichen in den weitaus meisten Fällen von Vorteil ist hinsichtlich des Anteils der Kanten, auf denen  $\mathbf{s}$  differenzierbar ist.



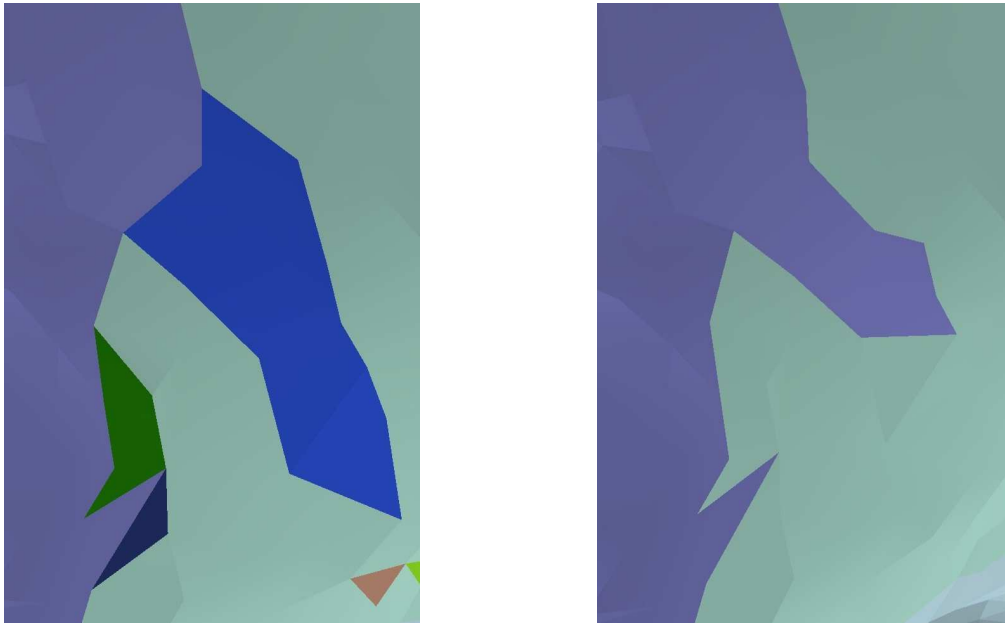


Abbildung 4.8: Auflösung von Bereichen zwischen Teiltriangulierungen: Sukzessive werden die Dreiecke des Zwischenbereichs zu den umgebenden Teilbereichen hinzugefügt.

Aufgrund dieser Beobachtungen entwickeln wir eine weitere effiziente Methode, indem wir Algorithmus 4.4 mehrmals mit unterschiedlichen Parametern anwenden:

**Algorithmus 4.5**

Sei  $\{\Delta_1, \dots, \Delta_N\}$  eine durch Algorithmus 4.1 konstruierte Zerlegung einer 3D-Triangulierung, und man wähle einen Winkel  $\beta > \alpha$ , eine maximale Bereichsgröße  $n$  in Dreiecken sowie eine Anzahl  $\lambda \in \mathbb{N}$  von Iterationen. Für  $i = 1, \dots, \lambda$  wende man Algorithmus 4.4 mit den Parametern

$$\beta_i = \alpha + (\beta - \alpha) \frac{i}{\lambda} \quad \text{und} \quad n_i = n - [(i - 1)n] \text{ div } \lambda$$

an, wobei  $x \text{ div } y$  den ganzzahligen, abgerundeten Quotienten zweier Zahlen  $x, y \in \mathbb{N}$  liefert.

Durch diese Methode werden zunächst große Bereiche bearbeitet, wobei maximal erlaubte Winkel  $\beta_i$  für die Normalenabweichung nur wenig größer als  $\alpha$  ist, bis schließlich nur noch kleine Bereiche betrachtet werden und sich der erlaubte Winkel  $\beta$  nähert. Auf diese Weise kann außer in Sonderfällen stets vermieden werden, dass die Verkleinerung eines betrachteten Bereichs dazu führt, dass neue kleine Teilbereiche entstehen, da diese im Allgemeinen in den folgenden Iterationen aufgelöst werden.

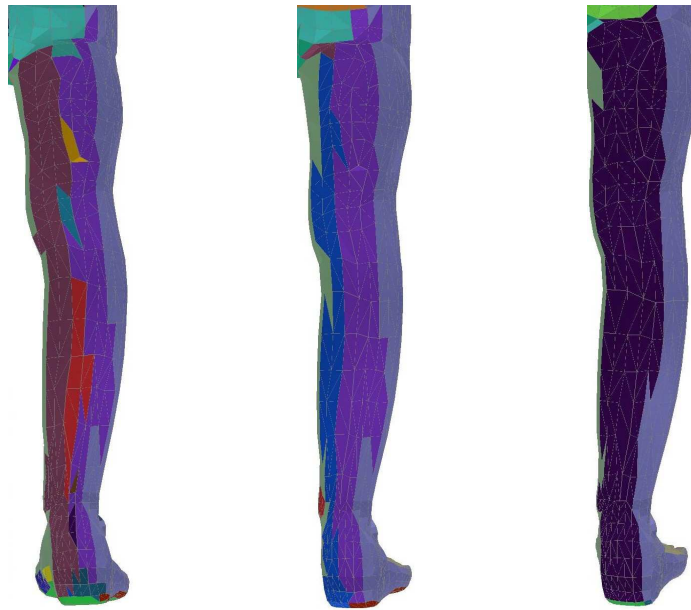


Abbildung 4.9: Anpassung kleiner Bereiche: Mit Algorithmus 4.4 (Mitte) werden kleine Bereiche angepasst und aufgelöst. Bei Algorithmus 4.5 (rechts) wird diese Methode mehrmals mit unterschiedlichen Parametern angewandt, sodass sich die Bereiche noch besser der Geometrie von  $\Delta$  anpassen.

Die Implementierung der verschiedenen Varianten zeigt, dass eine solche iterative Anpassung von Bereichen den weitaus höchsten Anteil von Kanten liefert, auf denen  $s$  differenzierbar ist. Die Ergebnisse sind auch dann noch sehr gut, wenn Winkel  $\beta$  gewählt werden, die nur geringfügig, z. B. um  $5^\circ$ , größer sind als  $\alpha$ . Für die Anzahl der Iterationen sind optimalerweise Werte  $\lambda \geq 10$  zu wählen. Ein Beispiel für die Anwendung der Algorithmen 4.4 und 4.5 zeigt Abbildung 4.9.

### Glättung von Rändern projizierbarer Teilbereiche

Wir entwickeln nun einen einfachen und effizienten Algorithmus zur Glättung der Ränder zwischen den einzelnen Teiltriangulierungen. Wie weiter oben gesehen entstehen Ränder, welche nicht „glatt“ verlaufen dadurch, dass in Algorithmus 4.1 Bereiche von optimaler Größe gewählt werden. Die einfachste und effizienteste Vorgehensweise ist es, für jedes einzelne Randdreieck jeder Teiltriangulierung zu überprüfen, ob es sich unter Berücksichtigung eines maximalen Winkels  $\gamma$  zu einer Nachbartriangulierung hinzufügen lässt und ob ein solcher Bereichswechsel den Anteil der Kanten erhöht, auf denen  $s$  differenzierbar ist. Diese Überprüfung von Dreiecken führt man solange fort, bis der Algorithmus terminiert. Als Winkel  $\gamma$  wird im Allgemeinen der in Algorithmus 4.1 bzw. den erweiterten Algorithmen zugrunde

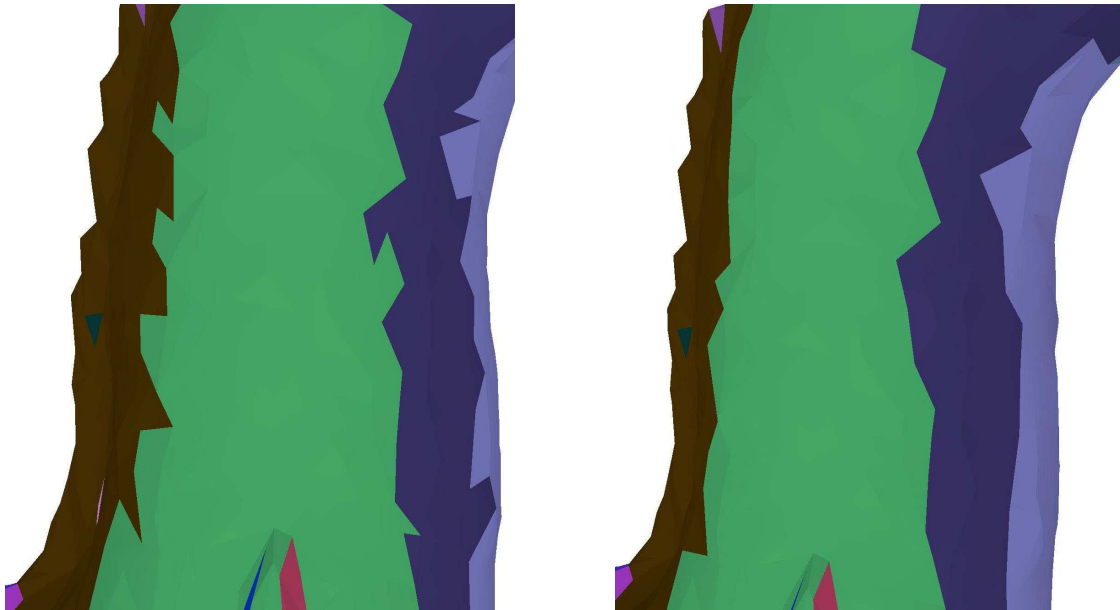


Abbildung 4.10: Glättung von Rändern durch Algorithmus 4.6

gelegte maximale Winkel  $\alpha$  bzw.  $\beta$  gewählt.

Dieser Algorithmus für die Glättung der Ränder lautet wie folgt:

**Algorithmus 4.6**

Solange Dreiecke  $T \in \Delta_l$  in  $\Delta$  mit mehr Nachbardreiecken in einer anderen Teiltriangulierung  $\Delta_{\bar{l}} \neq \Delta_l$  als in  $\Delta_l$  existieren, sodass die Normalenabweichung von  $T$  zu jedem Nachbardreieck aus  $\Delta_{\bar{l}}$  einen vorgegebenen Winkel  $\gamma$  nicht überschreitet, so entferne man  $T$  aus  $\Delta_l$  und füge es zu  $\Delta_{\bar{l}}$  hinzu.

Es ist offensichtlich, dass dieser Algorithmus in linearer Komplexität terminiert und dass sich mit jedem bearbeiteten Dreieck der Anteil der Kanten, auf denen  $s$  differenzierbar ist, vergrößert. Die computertechnische Umsetzung des Verfahrens zeigt, dass sich durch die Anwendung des Algorithmus der Anteil dieser Kanten wesentlich erhöht und dass die zwischen den projizierbaren Bereichen verlaufenden Kantenzüge glatter verlaufen. Ein Beispiel zeigt Abbildung 4.10.

In seltenen Situationen ist Algorithmus 4.6 nicht in der Lage, Kanten zu glätten. Dies ist immer dann der Fall, wenn ein Wechsel eines einzelnen Randdreiecks noch nicht zu einer Erhöhung des Anteils der Kanten führt, auf denen  $s$  differenzierbar ist, sondern gleichzeitig mehrere Dreiecke in eine andere Teiltriangulierung aufgenommen werden müssten.

Folgender Algorithmus ist geeignet, die Ränder der Teilbereiche auch in diesen seltenen Fällen zu glätten:

**Algorithmus 4.7**

Sei  $M \subseteq \Delta_l$  eine Menge von maximal  $\lambda$  Dreiecken mit der folgenden Eigenschaft: Es lassen sich alle Dreiecke  $T \in M$  aus  $\Delta_l$  entfernen und zu einer benachbarten Teiltriangulierung  $\Delta_{\bar{l}}$  hinzufügen, sodass

- (i) sowohl  $\Delta_l$  als auch  $\Delta_{\bar{l}}$  anschließend noch Triangulierungen sind.
- (ii) die Normale jedes Dreiecks aus  $M$  maximal um einen fest vorgegebenen Winkel  $\gamma$  zur Normalen des Startdreiecks  $T_0^{[\bar{l}]}$  abweicht.
- (iii) der Anteil der Kanten, auf denen  $\mathbf{s}$  differenzierbar ist, für die neue Zerlegung größer ist als für die ursprüngliche.

Solange eine solche Menge  $M$  für eine Teiltriangulierung  $\Delta_l$  existiert, entferne man jedes Dreieck von  $M$  aus  $\Delta_l$  und füge es zu  $\Delta_{\bar{l}}$  hinzu.

Zwar führt dieser Algorithmus für  $\lambda \geq 2$  in manchen Fällen zu geringfügig besseren Ergebnissen als Algorithmus 4.6, allerdings nimmt die Effizienz des Verfahrens ab, je größer  $\lambda$  gewählt wird. Des Weiteren ist eine computertechnische Umsetzung der Methode aufwändig, sodass wir angesichts der geringen Vorteile im Rahmen dieser Arbeit lediglich das einfachere Verfahren implementiert haben.

**Suche optimaler Bereiche**

Die Implementierung der Methode zeigt, dass bereits bei einer zufälligen Wahl von Startdreiecken Zerlegungen mit großen projizierbaren Teiltriangulierungen und einem hohen Anteil an Kanten, auf denen  $\mathbf{s}$  differenzierbar ist, erzeugt werden. Allerdings lässt sich dieser Anteil durch eine Wahl geeigneter Startdreiecke wesentlich erhöhen. Wir entwickeln im Folgenden einen Algorithmus zur Ermittlung solcher Dreiecke.

Zunächst charakterisieren wir das Problem für die Wahl geeigneter Startdreiecke:

**Problem 4.8**

Seien eine 3D-Triangulierung  $\Delta$  und ein Winkel  $\alpha$  gegeben. Man wende Algorithmus 4.1 auf  $\Delta$  an und wähle dabei Startdreiecke  $T_0^{[1]}, \dots, T_0^{[N]}$ , sodass der Anteil  $f$  der Kanten, auf denen  $\mathbf{s}$  differenzierbar ist, so groß wie möglich ist, dass also jede andere Wahl  $\tilde{T}_0^{[1]}, \dots, \tilde{T}_0^{[\tilde{N}]}$  von Startdreiecken einen Anteil  $\tilde{f} \leq f$  ergibt.

Es ist offensichtlich, dass diese optimale Wahl  $T_0^{[1]}, \dots, T_0^{[N]}$  nicht in effizienter Weise ermittelt werden kann. Insbesondere sind die einzelnen Iterationsschritte nicht unabhängig, denn jede optimale Wahl eines Startdreiecks ist abhängig davon, welche Dreiecke bereits Teiltriangulierungen zugeordnet sind.

Eine zweckmäßige Vorgehensweise ist es, in jedem Iterationsschritt ein Startdreieck zu ermitteln, welches zu einem Bereich mit möglichst vielen Dreiecken führt. Dies ist wesentlich effizienter implementierbar als für jede Wahl von Startdreiecken die inneren und äußeren Kanten der entstehenden Teiltriangulierung zu ermitteln. Auf diese Weise entstehen große Bereiche und damit eine Zerlegung mit einem hohen Anteil von Kanten, auf denen  $\mathbf{s}$  differenzierbar ist.

Es gibt verschiedene Ansätze für die Ermittlung von Startdreiecken, die zu Bereichen mit vielen Dreiecken führen. Sehr effizient sind lokale Verfahren. Grob gesprochen wählt man ein beliebiges in noch keiner Teiltriangulierung enthaltenes Dreieck  $T$  und überprüft Dreiecke eines lokalen Bereichs um  $T$ .

Ein einfaches und effizientes lokales Verfahren lautet wie folgt:

**Algorithmus 4.9**

Man wende Algorithmus 4.1 auf eine Triangulierung  $\Delta$  im Raum an. Für die Wahl jedes Startdreiecks  $T_0^{[l]}$ ,  $l = 1, \dots, N$ , gehe man wie folgt vor:

1. Man wähle ein beliebiges in noch keiner Teiltriangulierung enthaltenes Dreieck.
2. Falls das aktuell gewählte Dreieck  $T$  kein Nachbardreieck besitzt, welches als Startdreieck zu einem Teilbereich mit mehr Dreiecken führt als  $T$ , so wähle man  $T$  als Startdreieck  $T_0^{[l]}$ .
3. Andernfalls wähle man dasjenige Nachbardreieck, welches zu einem Teilbereich mit maximal vielen Dreiecken führt und verfare mit diesem Nachbardreieck wie in 2.

Diese Vorgehensweise hat den Nachteil, dass meist nur kleine Bereiche für die Wahl eines Startdreiecks überprüft werden. Ein weitaus besseres Ergebnis erhält man, indem in jedem Iterationsschritt jedes verbleibende Dreieck als Startdreieck überprüft wird. Dies ist jedoch sehr zeitintensiv, da für jede Iteration alle Dreiecke stets von Neuem zu prüfen sind.

Wir entwickeln im Folgenden einen effizienten Algorithmus, durch welchen für jeden Iterationsschritt ein größtmöglicher Bereich ermittelt wird, ohne dass alle Startdreiecke von Neuem überprüft werden müssen:

**Algorithmus 4.10**

Man wende Algorithmus 4.1 auf eine Triangulierung  $\Delta$  im Raum an. Hinsichtlich der Wahl von Startdreiecken gehe man wie folgt vor:

1. Man erstelle eine Liste, welche für jedes Dreieck  $T$  die Größe  $s(T)$  des entsprechenden projizierbaren Bereichs in Dreiecken enthält, wenn man  $T$  als Startdreieck wählt.
2. Solange eine nichtleere Menge  $M \subseteq \Delta$  von Dreiecken existiert, die noch in keiner Teiltriangulierung enthalten sind, gehe man wie folgt vor:
  - 2.1 Man wähle dasjenige Dreieck  $T \in M$  mit größtem Wert  $s(T) = \max_{\tilde{T} \in \Delta} s(\tilde{T})$ .
  - 2.2 Man aktualisiere den Wert  $s(T)$ .
  - 2.3 Falls auch der aktualisierte Wert maximal ist, wähle man  $T$  als Startdreieck, andernfalls wiederhole man die Schritte 2.1 bis 2.3.

Die Implementierung zeigt, dass diese Methode auch für komplexe Triangulierungen mit sehr vielen Dreiecken effizient ist, vgl. Übersicht 4.13 auf Seite 141.

#### 4.1.4 Implementierung der Methode

Alle Algorithmen wurden in *Java* [37, 49, 60, 84, 85] implementiert und mit dem Java Development Kit der Version 5.0.05 kompiliert. Für die grafischen Umsetzungen kam das *Java 3D-API* [6, 29, 39, 61, 81] in der Version 1.3.1 zum Einsatz, welches auf frei verfügbaren Low-Level-Treibern der *Open Graphics Library (OpenGL)* basiert. Die Module wurden auf einem Notebook mit 1,73 GHz Intel Pentium M<sup>TM</sup>-Prozessor und 512 MB RAM entwickelt und getestet. Das verwendete Betriebssystem war Microsoft Windows XP<sup>TM</sup> in der Version 5.1.2600.

#### Übersicht über die einzelnen Module

Alle implementierten Module wurden im package `spline3d` zusammengefasst. Folgende Zusammenstellung gibt einen Überblick über die enthaltenen Klassen:

class Output3D	(extends Applet, uses OpenGL) dient der Darstellung interaktiver 3D-Umgebungen.
class Tools	enthält zahlreiche Methoden zur Umsetzung mathematischer, grafischer und sonstiger verwendeter Algorithmen.
class Triangulation	dient der Verwaltung von (3D-)Triangulierungen.
class View3D	ermöglicht die Ansicht von (3D-)Triangulierungen und deren Zerlegungen in projizierbare Bereiche mit Hilfe des Output3D-Moduls.
class Projection /Decompose	liest eine 3D-Triangulierung, zerlegt diese mit Hilfe des Moduls "Projection/Decomposer" in projizierbare Bereiche und schreibt das Ergebnis in eine Zielfeile.
class Projection /Decomposer	zerlegt ein übergebenes "Triangulation"-Objekt in projizierbare Bereiche und implementiert dabei wesentliche in der Arbeit entwickelte Teilalgorithmen.

Unter einem 3D-Modell versteht man die Zusammenfassung aller Daten zur Darstellung und Verarbeitung der entsprechenden 3D-Triangulierung, also einer Liste der enthaltenen Dreiecke sowie einer Liste der zugehörigen Punktkoordinaten. Es gibt eine große Anzahl von groben und feinen 3D-Triangulierungen im Internet, insbesondere im umfangreichen Archiv der Universität von Stanford [73].

Mittlerweile existieren mehrere hundert verschiedene Dateiformate zur Speicherung von 3D-Modellen. Unsere Implementierungen setzen voraus, dass die einzulesenden Modelle im von Michael Garland [38] eingeführten smf-Format vorliegen. Dabei handelt es sich um ein einfaches, text- und zeilenbasiertes Format, bei dem zunächst eine Liste der Eckpunktkoordinaten und im Anschluss eine Liste der Dreiecke abgespeichert wird. Für die Speicherung von Zerlegungen haben wir dieses Modell erweitert, indem wir im Anschluss für jedes klassifizierte Dreieck  $T \in \Delta$  die zugehörige Klasse  $\Delta_l$ ,  $l = 1, \dots, N$ , angeben. Es folgt ein einfaches Modellbeispiel einer Zerlegung, bestehend aus zwei Dreiecken:

```
# Modelldatei für eine Zerlegung mit 2 Dreiecken
# Format: Michael Garlands smf, erweitert für Zerlegungen
# von 3D-Triangulierungen.

# 1. Liste der Knoten (vertices):
# ("v" xcoord ycoord zcoord)
v 0.0 0.0 0.0
v 1.0 0.0 0.0
v 0.0 1.0 0.0
v 0.0 0.0 1.0

# 2. Liste der Dreiecke (faces):
# ("f" vertex1 vertex2 vertex3)
f 0 1 2
f 0 1 3

# 3. Liste der Teiltriangulierungen (subtriangulations):
# ("s" face subtriangulation)
s 0 0
s 1 1

# 4. Liste der Projektionsebenen (planes):
# ("p" subtriangulation xnormal ynormal znormal)
p 0 0.0 0.0 1.0
p 1 0.0 1.0 0.0
```

Um das dreidimensionale Objekt im Rechner verwalten zu können, ist eine Anwendung verschiedener Datenstrukturen denkbar. Baumgart [7] entwickelte die so genannte *winged edge*-Struktur, bei welcher den Dreiecken die zugehörigen nummerierten Kanten und diesen wiederum die entsprechenden Eckpunkte zugeordnet sind. Wir haben uns für die Implementierung einer einfacheren Struktur entschieden, welche für die Bestimmung projizierbarer Teilbereiche hervorragend geeignet ist. Grob gesprochen werden die einzelnen Dreiecke nachbarweise verlinkt, außerdem werden jedem Knoten die zugehörigen Dreiecke zugewiesen. Auf diese Weise lassen sich sehr schnell die Nachbardreiecke und angrenzenden Dreiecke jedes Dreiecks  $T \in \Delta$  bestimmen und dadurch alle benötigten Methoden in einfacher und effizienter Weise umsetzen. Die folgende Übersicht zeigt die wichtigsten Attribute der Klasse `Triangulation`.



```

public class Triangulation

    //Vertex array:
    public Point3f[] vertex;

    //Triangle array containing each triangle's vertices:
    public int[][] triangle;

    //List containing each triangle's neighbours:
    public int[][] neighbour;

    //List linking each triangle to its vertices:
    public int[][] vertexMap;

    //List containing each triangle's normal vector:
    public Vector3f[] normal;

    //List containing each triangle's subtriangulation
    //(value is nrTriangles if a triangle is in no subtriangulation):
    public int[] subtriangle;

    //List linking each vertex v to its neighbouring vertices v1,...,vn
    //such that [v,vi], i=1,...,n are edges of the triangulation.
    public int[][] edgeList;

    //List linking each subtriangulation to the normal vector of the plane
    //in which it is to be projected.
    public Vector3f[] classNormal;

    [...]

```

Für die Darstellung der Objekte mit dem Java 3D-API wird ein so genanntes *simple universe* benötigt, eine dreidimensionale Umgebung, in welche Objekte aufgenommen und spezifiziert werden können und die eine zweidimensionale Projektion dieser Umgebung in einem Fenster, eine „Kameraperspektive“ quasi, wiedergibt. Für das `SimpleUniverse`-Objekt ist eine ganze Reihe von Eigenschaften festzulegen, angefangen von Helligkeits- und Lichtverhältnissen bis hin zu Steuerungselementen für Maus oder Tastatur. Alle relevanten Spezifikationen und Implementierungen fassen wir in der Applet-Subklasse `Output3D` zusammen. Eine Instanz dieser Klasse ist ein Fenster mit einem zugehörigen simple universe in Kameraperspektive. Dabei lässt sich mit der Maus navigieren, indem entweder die linke Maustaste („Rotation“) oder die rechte Maustaste („Seitwärtsbewegung“) gedrückt gehalten wird. Mit gedrückt gehaltener linker-Maustaste und der [Alt]-Taste kann „gezoomt“ werden. Ein sehr wichtiges Attribut der Klasse `Output3D` ist die `public BranchGroup root`. Dieses Objekt dient der Aufnahme sämtlicher darzustellender 3D-Objekte und -Objektspezifikationen, insbesondere der Dreiecke des Modells.

Für die Betrachtung einer 3D-Triangulierung und der farblichen Darstellung einer Zerlegung haben wir die Klasse `View3D` implementiert. Durch den Kommandozeilenaufruf

```
java spline3d/View3D Dateiname [Ansichtsmodus] [Reflexion] [Wireframe]
```

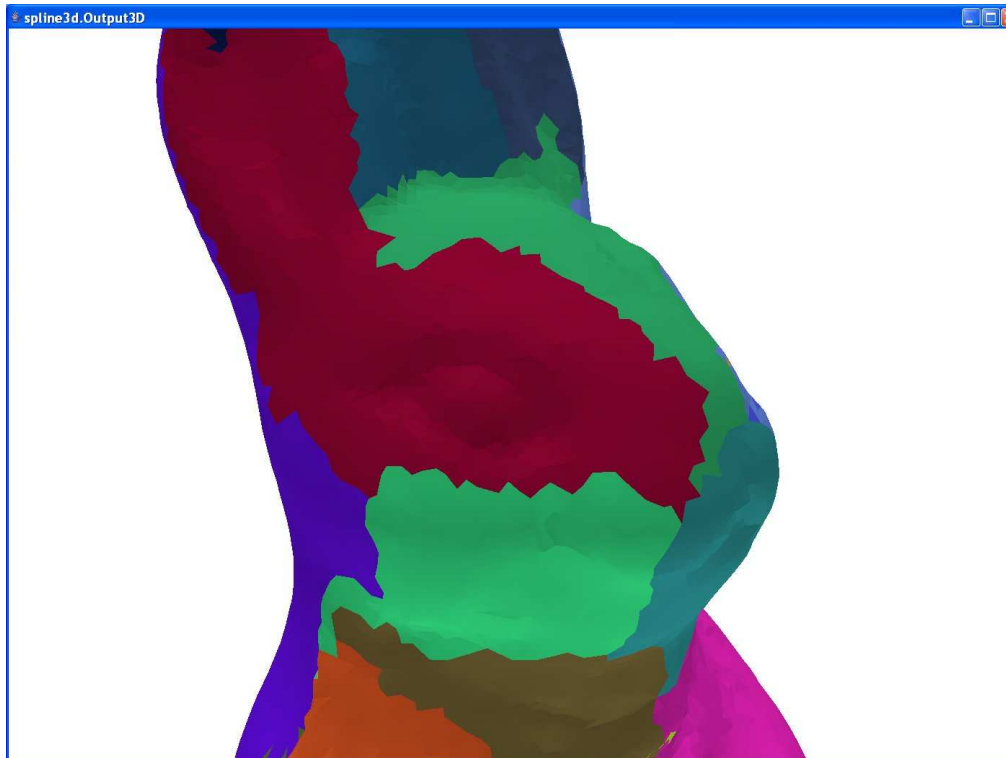


Abbildung 4.11: Ansicht eines Modells durch die `Output3D`-Umgebung

wird das in einer Datei abgelegte Modell in einer `Output3D`-Umgebung grafisch dargestellt. Die Parameter `Ansichtsmodus`, `Reflexion` und `Wireframe` spezifizieren die Art und Weise der Darstellung des Objekts, die Programmdokumentation auf der beigelegten CD beschreibt die diesbezüglichen Details. Als Standardwerte wird eine reflektierende und glänzende Materialoberfläche gewählt, das zum Objekt gehörende *wireframe* wird nicht dargestellt. Der `View3D`-Betrachter berechnet außerdem den Anteil der Kanten, auf denen  $s$  differenzierbar ist. Abbildung 4.11 zeigt die Ansicht eines in projizierbare Bereiche zerlegten 3D-Modells in einer `Output3D`-Umgebung.

### Zerlegung in projizierbare Bereiche

Das subpackage `spline3d/Projection` ermöglicht die Zerlegung eingelesener 3D-Triangulierung in projizierbare Bereiche gemäß den in dieser Arbeit entwickelten Algorithmen. Mit der Klasse `Projection/Decomposer` können diese Algorithmen auf eine gegebene Triangulierung bzw. Zerlegung angewandt werden. Ein Beispiel für die Verwendung dieser Klasse gibt die ausführbare Klasse `Projection/Decompose`: Durch den Kommandozeilenauf

```
java spline3d/Projection/Decompose Dateiname [-bwz]
```

wird die spezifizierte 3D-Triangulierung gelesen und gemäß festgelegter Optionen in projizierbare Bereiche zerlegt. Das Ergebnis wird in der Datei „`output.smf`“ im Verzeichnis „`Grobmodelle`“ abgespeichert. Durch die Option `-bwz` können `.smf`-Dateien eingelesen werden, bei denen die Nummerierung der Dreiecke und Eckpunkte mit 0 statt mit 1 beginnt.

Die folgende Zusammenfassung zeigt eine grobe Darstellung der wichtigsten Attribute der Klasse `Projection/Decomposer`:

```
public class Decomposer
{
    // Spezifiziert, welcher Algorithmus zur Behandlung von Löchern und
    // Zwischenbereichen verwendet werden soll (vgl. Algorithmen 4.5 und 4.6):
    // 0: kein Algorithmus, 1: einmaliges Anpassen, 2: iteratives Anpassen
    public static shrinkMode = 2;

    // Enthält die Anzahl der Iterationen, falls shrinkMode=2 ist:
    public static int nrShrinks = 10;

    // Legt fest, ob zwischen Rändern liegende Bereiche geglättet werden:
    public static boolean smoothEdges = true;

    // Maximale Winkel bezüglich des Hauptalgorithmus, des Algorithmus
    // zur Behandlung von Löchern sowie Zwischenbereichen und des Algorithmus
    // zur Ränderglättung:
    public static float maxAngle = (float)(30f/180f*Math.PI);
    public static float maxAngle2 = (float)(35f/180f*Math.PI);
    public static float maxAngle3 = (float)(35f/180f*Math.PI);

    // Methode zur Zerlegung von Triangulierungen:
    // Der Parameter tri legt die zu bearbeitende Triangulierung fest.
    // Durch die Option methodNr wird der für die Suche optimaler Bereiche
    // verwendete Algorithmus spezifiziert:
    // 0: Wahl zufälliger Startdreiecke
    // 1: Wahl optimaler Startdreiecke (hohe Komplexität des Algorithmus,
    // für den Performancevergleich implementiert)
    // 2: Wahl optimaler Startdreiecke (wie in Algorithmus 4.11 beschrieben,
    // hohe Effizienz)
    public static float decompose(Triangulation tri, int methodNr)
    {
        [...]
        // Liefert zerlegte Triangulierung tri2 zurück.
        return tri2;
    }
}
```

Die Zerlegung von  $\Delta$  durch die Methode `decompose` wurde rekursiv implementiert. Dadurch bleibt der Quellcode kurz und übersichtlich, und der Algorithmus ist sehr performant. Der Hauptspeicher heutiger Rechner ist bei der Verwendung praxisrelevanter grober Triangulierungen bei Weitem ausreichend. Aufgrund seiner zentralen Bedeutung geben wir im Folgenden den rekursiven Teil der Methode wieder. Für jeden Aufruf dieser Unter Methode `check` wird für ein bestimmtes Dreieck geprüft, ob es zum aktuell zu bearbeitenden Bereich noch hinzugefügt werden kann, ohne dass der maximale Winkel `maxAngle` überschritten wird:

```
01 public static void check(Triangulation tri, int toCheck)
02 {
03     check(tri, toCheck, tri.normal[startWithTriangle]);
04 }
05
06 public static void check(Triangulation tri, int toCheck, Vector3f fixedNormal)
07 {
08     if((tri.subtriangle[toCheck]<=classNr)
09         ||(tri.normal[toCheck].angle(fixedNormal)>maxAngle))
10     { return; }
11     else
12     {
13         classSize++;
14         tri.subtriangle[toCheck]=classNr;
15         // Recursive branch:
16         for (int i=0; i<tri.neighbour[toCheck].length; i++)
17             { check(tri, tri.neighbour[toCheck][i]); }
18     }
19 }
```

Die Parameter der Methode sind die zu bearbeitende Triangulierung, die Nummer des zu überprüfenden Dreiecks sowie gegebenenfalls der Normalenvektor der Ebene, in welche projiziert wird. Bei den in dieser Arbeit entwickelten Verfahren wird dabei stets der zum jeweiligen Startdreieck gehörende Normalenvektor verwendet. Damit die Implementierung gegebenenfalls auch für andere Ansätze geeignet ist (beispielsweise die Verwendung gemittelter Normalenvektoren bestimmter Bereiche), ist auch die Wahl eines anderen Projektionsvektors möglich. Der Index des aktuellen Bereichs ist dabei durch die als `private` deklarierte Klassenvariable `classNr` festgelegt, die Nummer des jeweiligen Startdreiecks wird vor dem ersten rekursiven Aufruf in der Variable `startWithTriangle` abgelegt. Somit wird mit dem Aufruf in Zeile 03 für den Normalenvektor als Standardwert der Normalenvektor des Startdreiecks übergeben.

Innerhalb des rekursiven Aufrufs wird zunächst geprüft, ob das zu überprüfende Dreieck bereits einem Bereich angehört. Ist dies der Fall, so wird die Rekursion beendet. Andernfalls wird die Abweichung der Normalenvektoren überprüft. Für eine Abweichung  $\leq \text{maxAngle}$  wird die Größe der aktuellen Klasse in der privaten Variable `classSize` angepasst und das Dreieck der Klasse hinzugefügt (Zeilen 13 und 14). Im Anschluss werden rekursiv alle Nachbarn des Dreiecks geprüft.

Bei der Wahl zufälliger Startdreiecke wird die Methode `check` so oft aufgerufen, bis keine unklassifizierten Dreiecke verbleiben. Bei jedem Aufruf wird dabei ein zufälliges nicht klassifiziertes Startdreieck gewählt und der Index der Klasse um eins erhöht. Bei der Suche geeigneter Startdreiecke wie in Abschnitt 4.1.3, *Suche optimaler Bereiche* beschrieben, ist die Vorgehensweise etwas komplexer. Für die Ermittlung einer Bereichsgröße, die sich aus einem zu überprüfenden Startdreieck  $T \in \Delta$  ergibt, wird die Methode `check` aufgerufen, wobei der resultierende Bereich nicht mit der Indexnummer `nrClasses`, sondern mit der Nummer `tri.nrTriangles+1` klas-

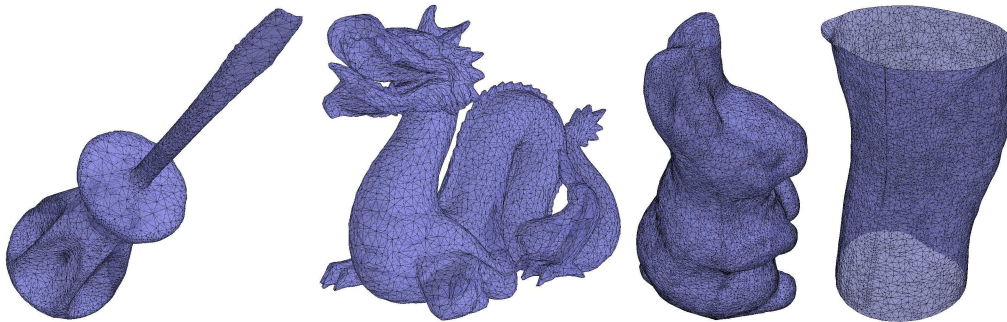


Abbildung 4.12: Die für die numerischen Tests verwendeten 3D-Modelle

sifiziert wird. Nicht klassifizierte Dreiecke werden dabei jeweils der Klasse mit der Nummer `tri.nrTriangles` zugeordnet. Nach der Klassifizierung enthält die private Klassenvariable `classSize` die ermittelte Bereichsgröße, und den Dreiecken wird wieder der Klassenwert `tri.nrTriangles` zugewiesen. Auf diese Weise können für beliebige Startdreiecke resultierende Bereichsgrößen ermittelt und verglichen werden, sodass sich optimale Startdreiecke wählen lassen. Auf diese Weise wurde der effiziente Algorithmus 4.10 für die Wahl optimaler Startdreiecke implementiert, indem die jeweils aktuellen Bereichsgrößen, die sich bei der Wahl bestimmter Startdreiecke ergeben, in einem Array abgespeichert werden.

### 4.1.5 Algorithmische Tests

Wir haben die verschiedenen Verfahren und Teilalgorithmen ausführlich an zahlreichen 3D-Modellen unterschiedlicher Größe und Komplexität auf Effizienz getestet und geben in diesem Abschnitt eine Zusammenfassung der wichtigsten Ergebnisse.

Im Rahmen der Tests haben wir die folgenden Werte verglichen:

1. Anteil der Kanten, auf denen  $s$  differenzierbar ist
2. Durchschnittliche Abweichung der Normalenvektoren von Dreiecken und Projektionsebenen
3. Zeitlicher Berechnungsaufwand

Es sei bemerkt, dass alle drei Größen stark von der Geometrie der zugrunde gelegten 3D-Triangulierung abhängen. Die durchschnittliche Abweichung der Normalenvektoren ist von Bedeutung, da kleine Winkel der projizierten Dreiecke die numerische Stabilität der Methode verringern können, siehe z. B. Theorem 2.19 oder [72]. Somit führt, grob gesprochen, eine geringere durchschnittliche Abweichung der Vektoren zu besseren approximativen Eigenschaften der Interpolanten.

<i>Modell</i>	Schrauben- zieher	Drachen	Hase	Knie
<i>Anzahl der Dreiecke</i>	13.574	20.000	33.518	18.861
<i>Zufällige Wahl der Startdreiecke, <math>\alpha = 30^\circ</math>, keine Zusatzalgorithmen</i>	80,402% 16,950° 0,031 sec.	70,568% 15,093° 0,125 sec.	87,019% 17,031° 0,156 sec.	92,647% 16,243° 0,047 sec.
<i>Optimale Wahl der Startdreiecke, <math>\alpha = 30^\circ</math></i>				
<i>keine Zusatzalgorithmen</i>	84,153% 16,832°	74,257% 15,973°	89,150% 17,491°	94,938% 16,082°
<i>Optimierung wie in Algorithmus 4.10: nicht optimiert:</i>	14,828 sec. 94,140 sec.	10,938 sec. 466,407 sec.	86,140 sec. 412,875 sec.	104,704 sec. 231,609 sec.
<i>Erweiterter Algorithmus 4.4 für Löcher und Zwischenbereiche, einfach, <math>\beta = 40^\circ</math></i>	87,037% 19,620°	78,753% 20,405°	93,179% 20,180°	96,562% 16,768°
<i>Erweiterter Algorithmus 4.5 für Löcher und Zwischenbereiche, 10 Iterationen, <math>\beta = 40^\circ</math></i>	87,564% 20,365°	79,473% 20,701°	93,492% 20,521°	96,562% 16,775°
<i>Erweiterter Algorithmus 4.6 für die Glättung von Rändern, <math>\gamma = 40^\circ</math></i>	85,641% 17,100°	77,050% 16,539°	91,464% 17,919°	96,100% 16,284°
<i>Kombination der Algorithmen 4.5 (10 Iterationen) und 4.6, <math>\beta = \gamma = 40^\circ</math></i>	88,400% 20,190°	81,056% 20,453°	94,517% 20,463°	97,236% 16,792°
<i>Zum Vergleich: Ohne Zusatzalgorithmen: Optimale Wahl der Startdreiecke, <math>\alpha = 40^\circ</math></i>	87,343% 21,667° 24,218 sec.	79,675% 21,896° 17,453 sec.	91,914% 22,607° 186,125 sec.	95,083% 21,357° 137,032 sec.

Angegeben sind der Anteil der Kanten, auf denen  $s$  differenzierbar ist, der durchschnittliche Winkel zwischen den Normalen der Dreiecke und Projektionsebenen sowie die Berechnungszeit.

Abbildung 4.13: Ergebnisse der Tests des Verfahrens

Wir haben diese Werte für vier verschiedene 3D-Triangulierungen ermittelt und in Übersicht 4.13 dargestellt. Diese Übersicht zeigt neben den Ergebnissen der Hauptalgorithmen die Verbesserungen durch die einzelnen Zusatzalgorithmen. Abbildung 4.12 zeigt die verwendeten Modelle.

Die Bestimmung optimaler Startdreiecke mit Hilfe einer aktualisierten Liste für die Bereichsgrößen führt bei diesen Beispielen zu einem bis zu 43 mal geringeren Zeitaufwand im Vergleich zum Algorithmus ohne Liste. Der Anteil der Kanten, auf denen  $\mathbf{s}$  differenzierbar ist, erhöht sich durch die optimale Wahl um durchschnittlich 3%. Die Kombination der Zusatzalgorithmen 4.5 (10 Iterationen) und 4.6 bringt für  $\beta = \gamma = 40^\circ$  eine weitere Erhöhung um durchschnittlich 3,4% mit sich.

Insgesamt ergibt sich, dass für alle betrachteten Modelle große projizierbare Bereiche mit einem hohen Anteil an Kanten erzeugt werden, auf denen  $\mathbf{s}$  differenzierbar ist. Dieser Anteil liegt für die Tests bei durchschnittlich 90%. Dabei erhält man durch die Anwendung der in Abschnitt 4.1.3 entwickelten erweiterten Teilalgorithmen einen wesentlich höheren Anteil von Kanten, auf denen  $\mathbf{s}$  differenzierbar ist, als bei einer Anwendung des Grundalgorithmus mit dem entsprechend größeren Wert für  $\alpha$ . Darüber hinaus ergeben sich dadurch durchschnittlich erheblich geringere Winkelabweichungen bei der Projektion, was sich positiv auf die approximativen Eigenschaften der Interpolanten auswirkt.

## 4.2 Bestimmung des Interpolanten

### 4.2.1 Berechnung des stetigen Interpolanten und Unterteilung der Dreiecke

Gegeben eine feine Oberfläche  $\Omega_f$ , berechnen wir zunächst einen stetigen, kubischen 3D-Splineinterpolanten  $\tilde{\mathbf{s}} : \Omega_\Delta \rightarrow \mathbb{R}^3$  wie in Kapitel 3. Dann färben wir die Dreiecke der zugehörigen (groben) 3D-Triangulierung  $\Delta$  wie in der bivariaten Methode: Ausgehend von einer beliebigen Färbung wechsle man die Farbe von Dreiecken mit mindestens zwei Nachbarn gleicher Farbe, solange solche Dreiecke existieren. Da die Anzahl der zwei gleichfarbigen Dreiecken angehörenden Kanten in jedem Iterationsschritt um mindestens zwei abnimmt, terminiert dieses Färbungsverfahren in linearer Komplexität.

Anschließend wird jedes weiße Dreieck  $[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$  durch einen Clough-Tochter-Split in drei Mikrodreiecke unterteilt, d. h. jeder Eckpunkt  $\mathbf{v}_i$ ,  $i = 1, 2, 3$ , wird mit dem Baryzentrum  $\mathbf{v}_0 = \frac{1}{3} \sum_{i=1}^3 \mathbf{v}_i$  des Dreiecks verbunden, siehe Abbildung 3.7 (rechts) auf Seite 104. Der Interpolant  $\tilde{\mathbf{s}}$  wird dabei wie in Satz 1.9 von den weißen Makrodreiecken auf die entsprechenden Mikrodreiecke umgerechnet.

Die unterteilte Triangulierung sei mit  $\triangle$  bezeichnet.

## 4.2.2 Glättung des Interpolanten

Im Anschluss zerlegen wir  $\Delta$  in projizierbare Teiltriangulierungen  $\Delta_l$ ,  $l = 1, \dots, N$ , und glätten den stetigen Spline  $\tilde{\mathbf{s}}$ , sodass der geglättete Interpolant  $\mathbf{s}$  auf fast allen Kanten differenzierbar ist. Auf den wenigen verbleibenden Kanten erhält man Differenzierbarkeit für drei Punkte. Die projizierbaren Bereiche ermitteln wir durch die in Abschnitt 4.1 entwickelten Algorithmen.

Wir bezeichnen im Folgenden für  $l = 1, \dots, N$  mit  $\Delta_l$  die projizierbaren Teiltriangulierungen und mit  $\bar{\Delta}_l$  die zugehörigen Projektionen in die Ebenen der Startdreiecke. Entsprechend seien für  $l = 1, \dots, N$  weiterhin mit  $\underline{\Delta}_l$  die Unterteilungen der projizierbaren Bereiche bezeichnet und mit  $\bar{\underline{\Delta}}_l$  die zugehörigen Projektionen.

Wir glätten  $\tilde{\mathbf{s}}$  zunächst auf den Knoten, anschließend auf den Kanten der 3D-Triangulierung.

### Glättung auf den Eckpunkten

Bei der Glättung von  $\tilde{\mathbf{s}}$  über den Eckpunkten von  $\Delta$  betrachten wir folgenden Sonderfall:

Enthält eine projizierte Teiltriangulierung  $\bar{\underline{\Delta}}_l$  zwei kollineare bzw. fast kollineare Randkanten mit gemeinsamem Eckpunkt  $v$ , so nennen wir jede weitere Kante  $[v, w]$  von  $\bar{\underline{\Delta}}_l$  und auch die entsprechende Kante in  $\underline{\Delta}_l$  *randdegeneriert bzw. fast randdegeneriert in  $v$* , andernfalls nennen wir sie *nicht randdegeneriert*.

Wir nennen eine in einem projizierten Eckpunkt  $v$  randdegenerierte bzw. fast randdegenerierte Kante auch *randdegeneriert bzw. fast randdegeneriert in  $\mathbf{v}$* , falls  $\mathbf{v}$  der zu  $v$  gehörende Eckpunkt aus  $\underline{\Delta}$  ist.

Dabei heißen zwei Kanten *kollinear*, falls sie dieselbe Steigung besitzen. Andernfalls heißen sie *fast kollinear*, falls sie maximal um einen vorgegebenen, kleinen Winkel  $\epsilon \in \mathbb{R}^+$  voneinander abweichen. Sind zwei Kanten weder kollinear noch fast kollinear, so heißen sie *nicht kollinear*.

Die gesonderte Behandlung dieser Fälle in Algorithmus 4.11 (Glättung von  $\tilde{\mathbf{s}}$  über den Eckpunkten von  $\Delta$ ) ist für die Stabilität der Methode von Bedeutung, siehe Abbildung 4.14.

Wir glätten den Interpolanten nun über den Eckpunkten von  $\Delta$  wie folgt:

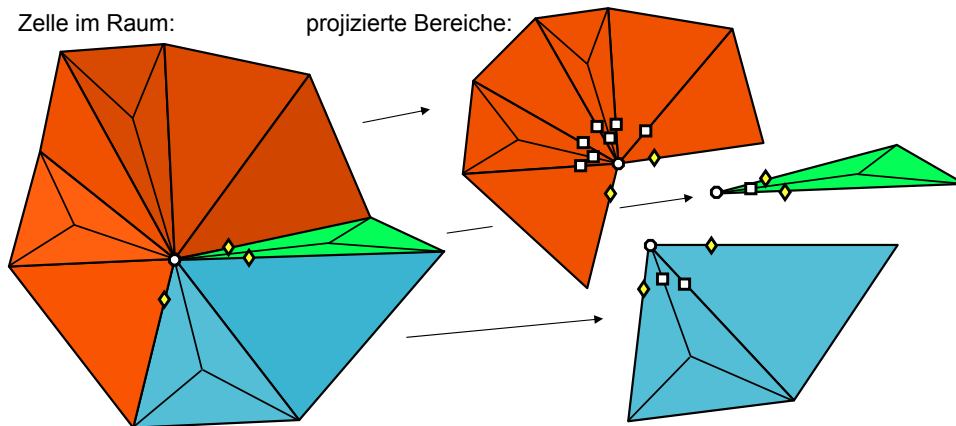
#### Algorithmus 4.11

Wir behandeln zunächst die Eckpunkte  $\mathbf{v}$  von  $\Delta$ , welche mehreren Teiltriangulierungen angehören: Im  $C^1$ -Ring  $\mathcal{C}$  von  $\underline{\Delta}$  jedes solchen Eckpunkts modifiziere man alle Bézier-Bernstein-Koeffizienten von  $\tilde{\mathbf{s}}$  wie folgt, siehe Abbildung 4.14:

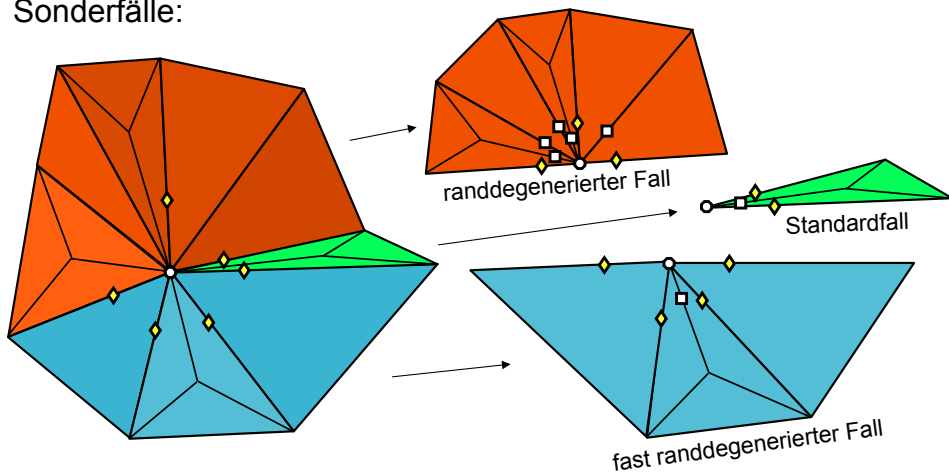


Glättung über einem Eckpunkt, welcher mehreren Bereichen angehört

Standardfall:



Sonderfälle:



- : vorgegebener Bézier-Bernstein-Koeffizient des Eckpunkts  $\mathbf{v}$
- ◆ : in die gemittelte Tangentialebene projizierte Koeffizienten
- : in den projizierten Ebenen komponentenweise angepasste Koeffizienten

Abbildung 4.14: Glättung über einem Eckpunkt  $\mathbf{v}$  mit mehreren Bereichen:

Im Standardfall werden in jedem Teilbereich genau zwei Koeffizienten des  $C^1$ -Rings um  $\mathbf{v}$  in die gemittelte Tangentialebene projiziert und die verbleibenden Koeffizienten komponentenweise wie im bivariaten Fall modifiziert.

Im randdegenerierten Fall wird dabei noch ein dritter projizierter Koeffizient benötigt.

Im fast randdegenerierten Fall werden alle Koeffizienten des  $C^1$ -Rings auf den Kanten der Makrodreiecke in die gemittelte Tangentialebene in  $\mathbf{v}$  projiziert und die verbleibenden Koeffizienten komponentenweise modifiziert.

1. Man bestimme die gemittelte Tangentialebene von  $\tilde{\mathbf{s}}$  in  $\mathbf{v}$  wie in Algorithmus 3.1 beschrieben.
2. Man projiziere alle Bézier-Bernstein-Koeffizienten aus  $\mathcal{C}$ , welche entweder in zwei Teiltriangulierungen enthalten sind oder auf einer Randkante von  $\Delta$  oder auf einer fast randdegenerierten Kante eines Makrodreiecks liegen, in die gemittelte Tangentialebene.

Anschließend betrachte man für jeden Koeffizienten  $\mathbf{c}$  aus  $\mathcal{C}$  auf einer fast randdegenerierten Kante  $[\mathbf{v}, \mathbf{v}_0]$  eines Clough-Tocher-unterteilten Dreiecks  $T$  mit Baryzentrum  $\mathbf{v}_0$  die Projektion  $\bar{T}$  von  $T$  und modifiziere  $\mathbf{c}$  nach (1.17) komponentenweise, sodass  $\tilde{\mathbf{s}}|_T$  differenzierbar in  $\mathbf{v}$  ist.

3. Für jeden Bereich  $\Delta_l$  mit randdegenerierten Kanten im Eckpunkt  $\mathbf{v}$ , wähle man eine solche Kante und projiziere den darauf liegenden Bézier-Bernstein-Koeffizienten aus  $\mathcal{C}$  in die gemittelte Tangentialebene.

Dabei wähle man eine Kante, welche in der Projektion  $\bar{\Delta}_l$  einen möglichst großen Winkel zu den kollinearen Randkanten bildet.

4. Für jeden weiteren Koeffizienten  $(c_1, c_2, c_3)$  in  $\mathcal{C}$  betrachte man die projizierten Dreiecke der entsprechenden Teiltriangulierung. Dabei existieren drei in den Schritten 2 und 3 modifizierte Koeffizienten in der  $C^1$ -Scheibe um  $\mathbf{v}$ , welche  $c_\tau$  für jede Komponente  $\tau = 1, 2, 3$  wie in (1.17) bestimmen.

Liegt  $(c_1, c_2, c_3)$  dabei auf einer randdegenerierten Kante, so bemerken wir, dass vier in den Schritten 2 und 3 modifizierte Koeffizienten der Teiltriangulierung in der  $C^1$ -Scheibe um  $\mathbf{v}$  existieren, wovon wegen der Kollinearität der Randkanten jedoch einer für die  $C^1$ -Bedingung nicht von Bedeutung ist.

Wir bezeichnen alle in dieser Weise durchlaufenen Eckpunkte und auch die entsprechenden projizierten Eckpunkte als *markiert*.

Weiterhin betrachten wir jeden Eckpunkt  $\mathbf{v}$  in  $\Delta$ , welcher nur einer einzigen Teiltriangulierung angehört. Für jeden projizierbaren Bereich  $\Delta_l$ ,  $l = 1, \dots, N$ , mit solchen Eckpunkten betrachten wir die entsprechende projizierte Teiltriangulierung  $\bar{\Delta}_l$  und verfahren wie folgt:

Für  $m = 3, 2, 1$  gehe man wie folgt vor: Solange ein Makrodreieck  $\bar{T} \in \bar{\Delta}_l$  mit  $m$  nicht markierten Eckpunkten  $v_1, \dots, v_m$  in  $\bar{\Delta}_l$  existiert, markiere man diese Eckpunkte und glätte  $\tilde{\mathbf{s}}$  nach folgender Methode:

Für jeden Eckpunkt  $v_i$ ,  $i = 1, \dots, m$ , und jede Komponente  $\tau = 1, 2, 3$  modifiziere man die nicht auf dem Rand von  $\bar{T}$  liegenden Bézier-Bernstein-Koeffizienten von  $\tilde{\mathbf{s}}_\tau$  mit Abstand eins zu  $v_i$ , sodass Gleichung (1.17) erfüllt ist.

Den in dieser Weise modifizierten Interpolanten bezeichnen wir mit  $\hat{\mathbf{s}}$ . Offensichtlich ist  $\hat{\mathbf{s}}$  differenzierbar auf jedem Eckpunkt  $\mathbf{v}$  von  $\Delta$ . Weiterhin gilt das folgende

**Lemma 4.12**

*Jede Abbildung  $\hat{s}_\tau|_{\Omega_{\bar{\Delta}_l}}$ ,  $l = 1, \dots, N$ ,  $\tau = 1, 2, 3$ , ist differenzierbar in jedem Eckpunkt von  $\bar{\Delta}_l$ , welcher nicht zwei fast kollinearen Randkanten und mehreren Teiltriangulierungen angehört.*

Diese Aussage folgt direkt aus der Modifikation der Koeffizienten des  $C^1$ -Rings um jeden Eckpunkt  $\mathbf{v}$  von  $\Delta$ . Damit gilt insbesondere

**Korollar 4.13**

Gegeben seien zwei benachbarte Dreiecke  $T_1, T_2$  eines projizierbaren Bereichs  $\Delta_l$ , die nicht demselben Makrodreieck aus  $\Delta_l$  angehören. Die gemeinsame Kante  $E$  sei dabei nicht fast randdegeneriert in einem Eckpunkt, welcher mehreren Bereichen angehört. Dann lässt sich der Bézier-Bernstein-Koeffizient  $\mathbf{c}_{111}^{[T_1]}$  komponentenweise modifizieren, sodass  $\hat{\mathbf{s}}$  differenzierbar auf  $E$  ist.

Dabei wird Bedingung (1.16) für die einzelnen Komponenten von  $\hat{\mathbf{s}}$  über den projizierten Dreiecken  $\bar{T}_1, \bar{T}_2$  betrachtet.

### Glättung auf den Kanten

Wir verwenden Korollar 4.13 nun für die Glättung von  $\hat{\mathbf{s}}$  über den Kanten der Triangulierung:

**Algorithmus 4.14**

Man glätte  $\hat{\mathbf{s}}$  über den Kanten von  $\Delta$  wie folgt und bezeichne den geglätteten Spline-Interpolanten mit  $\mathbf{s}$ :

1. Für jedes Paar  $T_1, T_2$  schwarzer Dreiecke mit gemeinsamer Kante  $E$  modifiziere man den Bézier-Bernstein-Koeffizienten  $\mathbf{c}_{111}^{[T_1]}$  wie folgt: Falls  $T_1$  und  $T_2$  in derselben Teiltriangulierung liegen und  $E$  nicht fast randdegeneriert in einem mehreren Bereichen angehörenden Eckpunkt ist, so modifiziere man  $\mathbf{c}_{111}^{[T_1]}$ , sodass  $\hat{\mathbf{s}}$  differenzierbar über der gemeinsamen Kante ist. Ansonsten modifiziere man  $\mathbf{c}_{111}^{[T_1]}$  wie in Algorithmus 3.3, sodass  $\hat{\mathbf{s}}$  im Mittelpunkt der gemeinsamen Kante differenzierbar ist.
2. In jedem weißen Mikrodreieck  $T_1$  mit schwarzem Nachbardreieck  $T_2$  modifiziere man  $\mathbf{c}_{111}^{[T_1]}$  wie in Schritt 1 und betrachte dabei statt dem schwarzen Dreieck  $T_1$  das weiße Mikrodreieck  $T_1$ .

3. Für jedes Paar  $T_1, T_2$  benachbarter weißer Mikrodreiecke in unterschiedlichen Makrodreiecken modifiziere man  $\mathbf{c}_{111}^{[T_1]}$  wie in Schritt 1, wobei statt den schwarzen Makrodreiecken die weißen Mikrodreiecke betrachtet werden.
4. Für jedes weiße Makrodreieck  $T$  mit Projektion  $\bar{T}$  seien für  $i = 1, 2, 3$  mit  $\bar{T}_i$  die projizierten Mikrodreiecke bezeichnet, und es sei  $\bar{v}_0$  das Baryzentrum von  $\bar{T}$ . Für jede Komponente  $\tau = 1, 2, 3$  lassen sich gemäß (1.16) und (1.17) zunächst die Koeffizienten des  $C^1$ -Rings um  $\bar{v}_0$  und schließlich der zu  $\bar{v}_0$  gehörende Koeffizient so modifizieren, dass  $s_\tau|_{\bar{T}}$  differenzierbar ist.

Den in dieser Weise geglätteten Interpolanten bezeichnen wir mit  $\mathbf{s}$ .

## 4.3 Eigenschaften der Methode

### 4.3.1 Differenzierbarkeit der geglätteten Interpolanten

Folgendes Theorem zeigt die Differenzierbarkeitseigenschaften des geglätteten Interpolanten:

**Theorem 4.15**

*Der geglättete Spline-Interpolant  $\mathbf{s} : \Delta \rightarrow \mathbb{R}^3$  ist differenzierbar über allen Eckpunkten und Kanten von  $\Delta$  mit Ausnahme folgender Kanten:*

1. *Jeder Kante, die zwei Teiltriangulierungen angehört.*
2. *Jeder Kante von  $\Delta$ , welche fast randdegeneriert in einem Eckpunkt ist, der mehreren Teiltriangulierungen angehört.*

*Auf diesen Kanten ist  $\mathbf{s}$  differenzierbar an den Eckpunkten und Kantenmittelpunkten.*

**Beweis:** Wir verwenden die Notationen der Algorithmen 4.11 und 4.14. Da  $\hat{\mathbf{s}}$  und somit  $\mathbf{s}$  wie gesehen über jedem Eckpunkt  $\mathbf{v}$  von  $\Delta$  differenzierbar ist, genügt es, die folgenden Aussagen zu zeigen:

- (i) Jede Abbildung  $s_\tau|_{\Omega_{\bar{\Delta}_l}}$ ,  $l = 1, \dots, N$ ,  $\tau = 1, 2, 3$ , ist differenzierbar in jedem Punkt  $v \in \Omega_{\bar{\Delta}_l}$ , der nicht auf einer Kante eines Makrodreiecks liegt, welche fast randdegeneriert in einem Eckpunkt ist, der mehreren Bereichen angehört.

- (ii)  $\mathbf{s}$  ist differenzierbar im Mittelpunkt jeder Kante aus  $\Delta$ , welche entweder zwei Teiltriangulierungen angehört oder fast randdegeneriert in einem mehreren Bereichen angehörenden Eckpunkt ist.

Bei (i) zeigen wir zunächst die Differenzierbarkeit in jeder Teiltriangulierung  $\bar{\Delta}_l$ ,  $l = 1, \dots, N$ , für jede Komponente  $\tau = 1, 2, 3$  über den inneren Kanten, welche nicht fast randdegeneriert in einem mehreren Bereichen angehörenden Eckpunkt sind. Dies folgt aus der Tatsache, dass für jedes Paar solcher Dreiecke der Teiltriangulierung die Bézier-Bernstein-Koeffizienten so modifiziert werden, dass Gleichung (1.16) für  $r = 1$  erfüllt ist, siehe auch Korollar 4.13. Schließlich werden in jedem projizierten unterteilten Dreieck  $\bar{T} = [v_1, v_2, v_3]$  die Koeffizienten der  $C^1$ -Scheibe um das Baryzentrum  $v_0$  von  $\bar{T}$  in jeder Komponente  $\tau = 1, 2, 3$  so angepasst, dass  $s_\tau$  über jeder Kante  $[v_0, v_i]$ ,  $i = 1, 2, 3$ , und damit auch insbesondere in  $v_0$  differenzierbar ist. Aussage (ii) folgt mit Lemma 3.4, da  $\mathbf{s}$  über jeder Kante von  $\Delta$ , welche entweder zwei Bereichen angehört oder fast randdegeneriert in einem mehreren Bereichen angehörenden Eckpunkt ist, wie in Algorithmus 3.3 geglättet wird.  $\square$

### 4.3.2 Lokalität und optimale Approximationsordnung

Abschließend betrachten wir Lokalitätseigenschaften und Approximationsordnung der Interpolanten.

#### Theorem 4.16

*Gegeben sei eine feine Oberfläche und der zugehörige geglättete Interpolant  $\mathbf{s} : \Omega \rightarrow \mathbb{R}^3$ . Bei Veränderung eines Interpolationswerts  $\mathbf{f}_z$ ,  $\mathbf{z} \in \mathcal{P}_q(\Delta)$ , verändert sich  $\mathbf{s}$  maximal in einem  $\text{star}^3(T)$ ,  $T \in \Delta$ .*

**Beweis:** Wir betrachten zunächst die Veränderung des stetigen Interpolanten  $\tilde{\mathbf{s}}$ . Da für die Bestimmung jeder polynomialen 3D-Oberfläche  $\tilde{\mathbf{s}}|_T$ ,  $T \in \Delta$ , lediglich die zu  $T$  gehörenden Interpolationswerte benötigt werden, verändert sich der stetige Interpolant ausschließlich auf jedem Dreieck, welches  $\mathbf{z}$  enthält. Bei der Unterteilung von  $\Delta$  wirkt sich die Veränderung nur auf die Mikrodreiecke der veränderten Makrodreiecke aus. Wir untersuchen nun die Veränderung für den geglätteten Interpolanten.

#### Fall 1:

Liege  $\mathbf{z}$  zunächst im Inneren eines Dreiecks  $T_1$  von  $\Delta$ . Da auf jeder Kante von  $\Delta$  bei der stetigen Interpolation vier Interpolationswerte vorliegen, bleibt der stetige und somit auch der geglättete Spline-Interpolant auf den Kanten der Makrodreiecke unverändert.

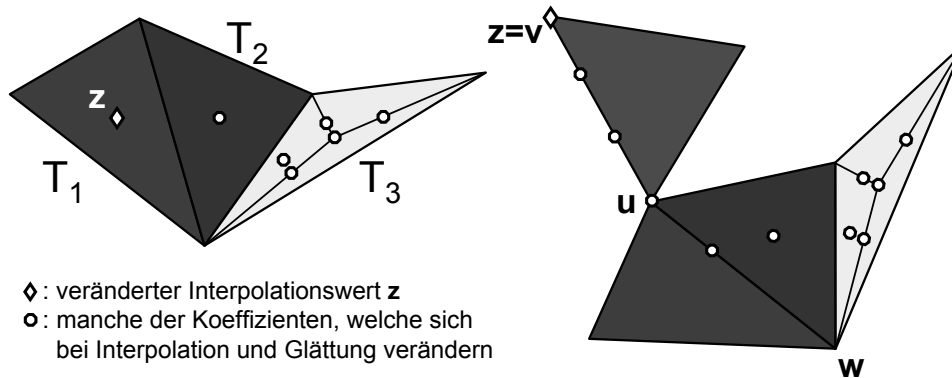


Abbildung 4.15: Maximale Veränderung von  $s$  bei der Veränderung eines Interpolationswertes im Inneren (links) oder auf einer Kante (rechts) eines Dreiecks

Im hinsichtlich Trägergröße ungünstigsten Fall ist  $T_1$  ein schwarzes Dreieck mit schwarzem Nachbarn  $T_2$ , sodass  $\mathbf{c}_{111}^{[T_2]}$  bezüglich der gemeinsamen Kante  $T_1 \cap T_2$  modifiziert wird. Mit dem Interpolationswert  $\mathbf{f}_z$  ändert sich dadurch der Koeffizient  $\mathbf{c}_{111}^{[T_1]}$  und damit auch  $\mathbf{c}_{111}^{[T_2]}$ . Im ungünstigsten Fall besitzt  $T_2$  eine gemeinsame Kante  $E$  mit einem weißen Makrodreieck  $T_3$ , sodass sich im Mikrodreieck von  $T_3$  mit gemeinsamer Kante zu  $T_2$  der Koeffizient  $\mathbf{c}_{111}^{[T_3]}$  verändert. Durch die weitere Glättung in  $T_2$  verändern sich dann auch die Koeffizienten der  $C^1$ -Scheibe um das Baryzentrum von  $T_2$ . Insgesamt verändert sich  $s$  im ungünstigsten Fall daher auf dem  $\text{star}^1(T_1)$ . Ein Beispiel zeigt Abbildung 4.15 (links).

#### Fall 2:

Liege  $\mathbf{z}$  nun auf einer Kante von  $\Delta$ . Im ungünstigsten Fall liegt  $\mathbf{z}$  dann auf einem Eckpunkt  $\mathbf{v}$  von  $\Delta$  der Iteration  $m = 3$ . Durch den veränderten Interpolationswert verändern sich die Bézier-Bernstein-Koeffizienten von  $s$  auf allen Kanten von  $\Delta$  mit Eckpunkt  $\mathbf{v}$ . Im ungünstigsten Fall existiert eine Kante  $[\mathbf{u}, \mathbf{w}]$ ,  $\mathbf{u} \neq \mathbf{v}$ ,  $\mathbf{w} \neq \mathbf{v}$ , der Iteration  $m = 2$  mit einem zu  $\mathbf{v}$  benachbarten Eckpunkt  $\mathbf{u}$ . Im ungünstigsten Fall verändert sich der zu  $\frac{2}{3}\mathbf{u} + \frac{1}{3}\mathbf{w}$  gehörende Koeffizient, und  $[\mathbf{u}, \mathbf{w}]$  ist gemeinsame Kante zweier schwarzer Dreiecke, sodass sich bei der Glättung von  $\hat{s}$  über  $E_2$  in einem dieser Dreiecke der im Baryzentrum liegende Koeffizient verändert. Sei dieses Dreieck mit  $T_3$  bezeichnet. Im ungünstigsten Fall besitzt  $T_3$  ein weißes Nachbardreieck  $T_4$ , auf dem sich Koeffizienten wie in Fall 1 beschrieben verändern.

Bei Berücksichtigung beider Fälle liegt  $\mathbf{z}$  im ungünstigsten Fall folglich auf einer Kante von  $\Delta$ , und der geglättete Spline-Interpolant  $s$  verändert sich in einem  $\text{star}^3(T)$ ,  $T \in \Delta$ , siehe Abbildung 4.15 (rechts).  $\square$

Für ebene Triangulierungen bringt das Verfahren Interpolanten von optimaler Approximationsordnung hervor. Die entsprechenden Argumente aus dem Beweis zu Theorem 3.16 sind analog anwendbar. Dabei ist insbesondere von Bedeutung, dass in Schritt 2 von Algorithmus 4.11 der Fall von fast kollinearen Randkanten gesondert betrachtet wird. Die Konstante  $L$  aus Abschätzung (3.25) hängt dabei neben  $\mathbf{f}$  und  $\alpha$  auch von der Konstanten  $\epsilon$  ab.

Zuletzt bemerken wir, dass sich die entwickelten Methoden auch für Splines auf Quadrangulierungen verwenden lassen:

**Bemerkung 4.17 (3D-Splines auf projizierten Quadrangulierungen)**

Die in diesem Kapitel entwickelten Methoden lassen sich in ähnlicher Weise für quadrangulierte 3D-Triangulierungen wie in Kapitel 3 anwenden. Man ermittelt einen stetigen Interpolanten  $\tilde{\mathbf{s}}$ , definiert auf einer quadrangulierten 3D-Triangulierung  $\diamond$ , und projiziert die Vierecke und Dreiecke von  $\diamond$  wie in Abschnitt 4.1 in geeignete Ebenen. Basierend auf Färbungs- und Prioritätsprinzipien modifiziert man anschließend das Bézier-Bernstein-Netz von  $\tilde{\mathbf{s}}$  ähnlich wie in Abschnitt 4.2.2.

Auf diese Weise erhält man lokal bestimmbare Splineoberflächen, welche auf allen Eckpunkten und fast allen Kanten differenzierbar sind. Das Verfahren besitzt optimale Approximationsordnung, falls Ausgangstriangulierungen der Ebene betrachtet werden.

# Literaturverzeichnis

- [1] P. Alfeld, B. Piper und L. L. Schumaker. An explicit basis for  $C^1$  quartic bivariate splines. *SIAM Journal of Numerical Analysis*, 24:891–911, 1987.
- [2] P. Alfeld, B. Piper und L. L. Schumaker. Minimally supported bases for spaces of bivariate piecewise polynomials of smoothness  $r$  and degree  $d > 4r + 1$ . *Computer Aided Geometric Design*, 4:105–123, 1987.
- [3] P. Alfeld und L. L. Schumaker. The dimension of bivariate spline spaces of smoothness  $r$  for degree  $d \geq 4r + 1$ . *Constructive Approximation*, 3:189–197, 1997.
- [4] P. Alfeld und L. L. Schumaker. Smooth macro elements based on clough-tocher triangle splits. *Numerical Mathematics*, 2002.
- [5] P. Alfeld und L. L. Schumaker. Smooth macro elements based on powell-sabin triangle splits. *Advances in Computational Mathematics*, 2002.
- [6] J. Barilleaux. *3D User Interfaces With Java 3D<sup>TM</sup>*. Manning Publications, 2000.
- [7] B. G. Baumgart. Winged edge polyhedron representation. <http://www.ncstrl.org:8900/ncstrl/servlet/search?formname=detail&id=oai%3Ancstrlh%3Astan%3ASTAN%2F%2FCS-TR-72-320>, Stanford University, Stanford, CA, USA, Technischer Bericht: CS-TR-72-320, 1972.
- [8] W. Boehm, G. Farin und J. Kahmann. A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design*, 1:1–60, 1984.
- [9] P. Bézier. Essay de définition numérique des courbes et de surfaces expérimentales, 1977.
- [10] C. K. Chui, G. Hecklin, G. Nürnberger und F. Zeilfelder. Optimal lagrange interpolation with quartic  $C^1$  splines on triangulations. *Journal of Computational and Applied Mathematics*. Submitted.
- [11] C. K. Chui und M.-J. Lai. Multivariate vertex splines and finite elements. *Journal of Approximation Theory*, 60:245–343, 1990.



- 
- [12] C. K. Chui und M.-L. Lai. On bivariate super vertex splines. *Constructive Approximation*, 6, 1990.
- [13] P. G. Ciarlet und P. A. Raviart. General lagrange and hermite interpolation in  $\mathbb{R}^n$  with applications to finite element methods. *Arch. Rational Mech. Anal.*, 46:177–199, 1972.
- [14] P. Cignoni, C. Montani, P. Pinci, C. Rocchini und R. Scopigno. A suite of tools for the management of 3D scanned data. *Italian-Canadian Workshop on 3D Digital Imaging and Modeling Conference Proc.*, Seiten 1–8, April 2001.
- [15] P. Cignoni, C. Montanini und R. Scopigno. A comparison of mesh simplification algorithms. *Computer Graphics*, 22, 1:37–54, 1998.
- [16] R. W. Clough und J. L. Tocher. Finite element stiffness matrices for analysis of plates bending. *Conference on Matrix Methods in Structural Mechanics*, 1965.
- [17] O. Davydov, G. Nürnberger und F. Zeilfelder. Approximation order of bivariate spline interpolation for arbitrary smoothness. *Journal of Computational and Applied Mathematics*, 90:117–134, 1998.
- [18] O. Davydov, G. Nürnberger und F. Zeilfelder. Interpolation by cubic splines on triangulations. *Approximation Theory*, IX:17–24, 1998.
- [19] O. Davydov, G. Nürnberger und F. Zeilfelder. Interpolation by splines on triangulations. *New Developments in Approximation Theory*, ISNM 132:49–70, 1999.
- [20] O. Davydov, G. Nürnberger und F. Zeilfelder. Bivariate spline interpolation with optimal approximation order. *Constructive Approximation*, 17:181–208, 2000.
- [21] O. Davydov, G. Nürnberger und F. Zeilfelder. Cubic spline interpolation on nested polygon triangulations. *Curve and Surface Fitting: St. Malo 1999*, Seiten 161–170, 2000.
- [22] C. de Boor. B-form basics. *Geometric Modeling*, Seiten 131–148, 1987.
- [23] C. de Boor. *A Practical Guide to Splines, revised edition*. Springer-Verlag, 2002.
- [24] C. de Boor und K. Höllig. Approximation order from bivariate  $C^1$  cubics: a counterexample. *Proceedings of the American Mathematical Society*, 87:649–655, 1983.
- [25] C. de Boor und K. Höllig. Approximation power of smooth bivariate PP functions. *Mathematische Zeitschrift*, 197:343–363, 1988.

- 
- [26] C. de Boor und Q. Jia. A sharp upper bound on the approximation order of smooth bivariate pp functions. *Journal of Approximation Theory*, 72:24–33, 1993.
- [27] P. de Casteljaou. Outillage méthodes calcul. *Automobiles SA, Paris*, 1959.
- [28] P. de Casteljaou. Courbes et surfaces à poles. *Automobiles SA, Paris*, 1963.
- [29] Michael Deering, Kevin Rushforth und H. Sowizral. *The Java 3D<sup>TM</sup> API Specification (2nd Edition)*. Addison-Wesley Publications, 2000.
- [30] A. D. DeRose, T. Duchamp, M. Eck, H. Hoppe, M. Lounsberry und W. Stuetzle. Multiresolution analysis of arbitrary meshes. *SIGGRAPH*, Seiten 173–182, 1995.
- [31] T. DeRose, T. Duchamp, H. Hoppe, J. McDonald und W. Stuetzle. Surface reconstruction from unorganized points. *SIGGRAPH*, Seiten 71–78, 1992.
- [32] A. Engelmann. Bernstein-Bézier-methoden und Interpolation mit bivariaten Splineräumen (Inauguraldissertation). Vorliegend an der Universität Mannheim, Lehrstuhl für Mathematik IV, 2003.
- [33] G. Farin. Bézier polynomials over triangles and the construction of piecewise  $C^r$ -polynomials. *Technical Report*, TR/91, 1980.
- [34] G. Farin. A construction fo the visual  $C^1$  continuity of polynomial surface patches. *Computer Graphics and Image Processing*, 20:272–282, 1982.
- [35] G. Farin. A modified clough-tocher-interpolant. *Computer Aided Geometric Design*, 2:19–27, 1985.
- [36] G. Farin. Triangular bernstein-bézier patches. *Computer Aided Geometric Design*, 3:83–127, 1986.
- [37] D. Flanagan. *Java in a Nutshell. Deutsche Ausgabe der 4. Auflage*. O'Reilly, 2003.
- [38] Michael Garland. homepage. <http://graphics.cs.uiuc.edu/~garland/>.
- [39] D. Gehringer und A. E. Walsh. *Java 3D API Jump-Start*. Prentice Hall PTR, 2001.
- [40] C. Gotsman, S. Gumhold und L. Kobbelt. Simplification and compression of 3D meshes. *Tutorials on Multiresolution in Geometric Modelling*, Seiten 319–361, 2002.
- [41] J. Grüntjes. Ein neuer ansatz zum display von oberflächen beliebigen topologie-typs: 3D-splines, Diplomarbeit. Vorliegend an der Universität Mannheim, Lehrstuhl für Mathematik IV, 2004.

- [42] I. Guskov, P. Schröder, W. Sweldens und K. Vidimce. Normal meshes. *SIGGRAPH*, Seiten 95–102, 2000.
- [43] G. Hecklin. Interpolation mit bivariaten und trivariaten Splineräumen (Inauguraldissertation). Vorliegend an der Universität Mannheim, Lehrstuhl für Mathematik IV, 2005.
- [44] G. Hecklin, G. Nürnberger, L. L. Schumaker und F. Zeilfelder. A local lagrange interpolation method based on  $C^1$  cubic splines on freudenthal partitions. *Computational Mathematics*. Submitted.
- [45] G. Hecklin, G. Nürnberger und F. Zeilfelder. Structural analysis of  $C^1$  spline spaces on freudenthal partitions. *SIAM Journal of Mathematical Analysis*. To appear.
- [46] N. Kohlmüller. Lagrange und Hermite Interpolation mit Splineräumen auf Triangulierungen (Inauguraldissertation). Vorliegend an der Universität Mannheim, Lehrstuhl für Mathematik IV, 2001.
- [47] N. Kohlmüller, G. Nürnberger und F. Zeilfelder. Construction of cubic 3D spline surfaces by lagrange interpolation at selected points. *Curve and Surface Design, Saint Malo 2002*, Seiten 235–245, 2003.
- [48] N. Kohlmüller, G. Nürnberger und F. Zeilfelder. Optimal approximation order of interpolation by cubic spline surfaces. *Curve and Surface Design, Saint Malo 2002*, Seiten 235–245, 2003.
- [49] G. Krüger. *Handbuch der Java-Programmierung*. Addison-Wesley Longman, 2002.
- [50] M. Laghchim-Lahlou und P. Sablonnière. Composite quadrilateral finite elements of class  $C^3$ . *Mathematical Methods in CAGD*, Seiten 413–418, 1989.
- [51] M. Laghchim-Lahlou und P. Sablonnière.  $C^r$  finite elements of HCT, PS and FVS types. *Proceedings of the Fifth International Symposium on Numerical Methods in Engineering*, 2:163–168, 1989.
- [52] M. Laghchim-Lahlou und P. Sablonnière. Triangular finite elements of HCT type and class  $C^p$ . *Advances in Computational Mathematics*, 2:101–122, 1994.
- [53] M. Laghchim-Lahlou und P. Sablonnière.  $C^r$  finite elements of powell-sabin types on the three directional mesh. *Advances in Computational Mathematics*, 6:191–206, 1996.
- [54] M.-J. Lai. Approximation order from bivariate  $C^1$ -cubics on a four-directional mesh is full. *Computer Aided Geometric Design*, 11:215–223, 1994.

- 
- [55] M.-J. Lai. Scattered data interpolation and approximation using bivariate  $C^1$  piecewise cubic polynomials. *Computer Aided Geometric Design*, 13, 1996.
- [56] M.-J. Lai und L. L. Schumaker. Macro elements and stable bases for splines on clough-tocher triangulations. *Numerical Mathematics*, 88:105–119, 2001.
- [57] M.-J. Lai und L. L. Schumaker. On the approximation power of bivariate splines. *Adv. Comp. Math.*, 9:251–279, 2001.
- [58] M.-J. Lai und L. L. Schumaker. Macro elements and stable bases for splines on powell-sabin triangulations. *Mathematics of Computation*, 2002.
- [59] A. Meyer. Interpolation mit bivariaten splines, theorie und praxis einer java-anwendung, Diplomarbeit. Vorliegend an der Universität Mannheim, Lehrstuhl für Mathematik IV, 2003.
- [60] Sun Microsystems. Java™ 2 platform standard edition v 1.4.1 API specification. <http://java.sun.com/j2se/1.4.1/docs/api/index.html>, 2003.
- [61] Sun Microsystems. Java™ 3D platform v 1.3 API specification. <http://java.sun.com/products/java-media/3D/forDevelopers/j3dapi/index.html>, 2006.
- [62] G. Nürnberger. *Approximation by Spline Functions*. Springer, Berlin, 1989.
- [63] G. Nürnberger. Approximation order of bivariate spline interpolation. *Journal of Approximation Theory*, 87:117–136, 1996.
- [64] G. Nürnberger und Th. Rießinger. Lagrange and hermite interpolation by bivariate splines. *Numer. Funct. Anal. Optim.*, 13:75–96, 1992.
- [65] G. Nürnberger und Th. Rießinger. Bivariate spline interpolation at grid points. *Numerical Mathematics*, 71:91–119, 1995.
- [66] G. Nürnberger, L. L. Schumaker und F. Zeilfelder. Local lagrange interpolation by bivariate  $C^1$  cubic splines. *Mathematical Methods for Curves and Surfaces, Oslo*, III:393–404, 2000.
- [67] G. Nürnberger, L. L. Schumaker und F. Zeilfelder. Lagrange interpolation by  $C^1$  cubic splines on triangulations of separable quadrangulations. *Approximation Theorie, X: Splines, Wavelets and Applications*, Seiten 405–424, 2002.
- [68] G. Nürnberger, L. L. Schumaker und F. Zeilfelder. Lagrange interpolation by  $C^1$  cubic splines on triangulated quadrangulations. *Advances in Computational Mathematics*, 21:381–419, 2004.

- [69] G. Nürnberger, L. L. Schumaker und F. Zeilfelder. Two lagrange interpolation methods based on  $C^1$  splines on tetrahedral partitions. *Approximation Theory XI: Gatlinburg 2004*, Seiten 101–118, 2005.
- [70] G. Nürnberger und G. Walz. Error analysis in interpolation by bivariate  $C^1$ -splines. *IMA Journal of Numerical Analysis*, 18:485–508, 1998.
- [71] G. Nürnberger und F. Zeilfelder. Developments in bivariate spline interpolation. *Journal of Computational and Applied Mathematics*, 121:125–152, 2000.
- [72] G. Nürnberger und F. Zeilfelder. Lagrange interpolation by bivariate  $C^1$ -splines with optimal approximation order. *Advances in Computational Mathematics*, 21:381–419, 2004.
- [73] University of Stanford. The stanford 3D scanning repository. <http://graphics.stanford.edu/data/3Dscanrep/>, 2006.
- [74] B. Piper. Tangent pane continuity between triangular bézier patches. *Geometric Modeling*, 1986.
- [75] M. J. D. Powell und M. A. Sabin. Piecewise quadratic approximation on triangles. *ACM Transactions on Mathematical Software*, 3:316–325, 1977.
- [76] Th. Rießinger. Interpolation by bivariate quadratic splines on a four-directional mesh. *Computing*, 49:129–137, 1992.
- [77] P. Sablonnière. Composite finite elements of class  $C^2$ . *Topics in Multivariate Approximation*, Seiten 207–217, 1987.
- [78] I. J. Schoenberg und A. Whitney. On polya frequency functions: The positivity of translation determinants with application to the interpolation problem by spline curves. *Trans. Amer. Math. Soc.*, 74:246–259, 1953.
- [79] L. L. Schumaker. *Basic Theory*. Wiley Interscience, New York, 1980.
- [80] L. L. Schumaker. Bounds on the dimension of spaces of multivariate piecewise polynomials. *Rocky Mountain Journal of Mathematics*, 14:251–264, 1984.
- [81] D. Selman. *Java 3D Programming*. Manning Publications, 2002.
- [82] Zh. Sha. On interpolation by  $\mathcal{S}_2^1(\Delta_{m,n}^{(2)})$ . *Approx. Theory & Appl.*, 2:71–82, 1985.
- [83] Zh. Sha. On interpolation by  $\mathcal{S}_3^1(\Delta_{m,n}^{(1)})$ . *Approx. Theory & Appl.*, 4:1–18, 1985.
- [84] R. Steyer. *Java 2 - Kompendium. Professionelle Programmierung mit J2SE Version 1.3*. Markt+Technik, 2001.

- 
- [85] C. Ullenboom. *Java ist auch eine Insel*. Galileo Press, 2003.
- [86] T. Wang. A  $C^2$ -quintic spline interpolation scheme on triangulations. *Computer Aided Geometric Design*, 9:379–386, 1992.
- [87] D. R. Wilhelmsen. A markov inequality in several dimensions. *J. Approx. Theory*, 11:216–220, 1974.

## Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig angefertigt habe und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Mannheim, im Oktober 2006

Andreas Dinh