



Zentralinstitut für Angewandte Mathematik

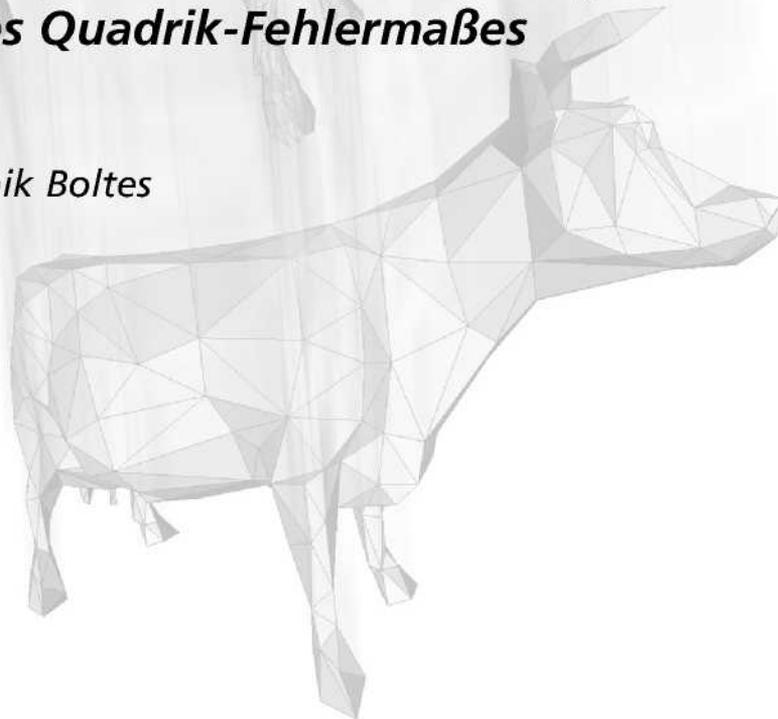
***Adaptive Netzvereinfachung auf der Basis
des Quadrik-Fehlermaßes***

Maik Boltes



***Adaptive Netzvereinfachung auf der Basis
des Quadrik-Fehlermaßes***

Maik Boltes



Berichte des Forschungszentrums Jülich ; 4157

ISSN 0944-2952

Zentralinstitut für Angewandte Mathematik Jül-4157

Zu beziehen durch : Forschungszentrum Jülich GmbH - Zentralbibliothek

D-52425 Jülich - Bundesrepublik Deutschland

☎ 02461 61-5220 - Telefax : 02461 61-6103 - e-mail : zb-publication@fz-juelich.de

Vollständig veröffentlicht im Internet

Persistent Identifier: [urn:nbn:de:0001-00234](https://nbn-resolving.org/urn:nbn:de:0001-00234)

<http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:0001-00234>

Copyright: Forschungszentrum Jülich 2004

Alle Rechte vorbehalten. Kein Teil des Werks darf in irgendeiner Form ohne schriftliche Genehmigung des Verlags reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Zusammenfassung

In der Computergrafik werden zur Erhöhung der Darstellungsqualität immer detailreichere Modelle verwendet. Der gewünschte Detailgrad ist aber je nach Anwendung unterschiedlich, so dass zur Einsparung von Prozessorleistung oder Speicherplatz die Modelle vorzugsweise automatisch vereinfacht werden müssen. Diese Vereinfachung der betrachteten Dreiecksnetze soll dabei die Originalnetze optisch möglichst gut wiedergeben.

Beim ausführlichen Vergleich existierender Verfahren fällt die Wahl für das in der Implementation herangezogene Maß des bei der Simplifizierung entstehenden Fehlers auf das Quadrik-Fehlermaß. Dieses in der Arbeit eingehend beschriebene Maß zeichnet sich durch seine Generalität, hohen Qualität und Geschwindigkeit aus. Die Simplifizierung geschieht dabei durch kontinuierliche Kantenkontraktion, deren Bewertung des geometrischen Fehlers über den quadratischen Abstand der Netzknoten des simplifizierten Netzes zu Ebenen durch die korrespondierenden Dreiecksflächen des Originalmodells geschieht.

Der Einsatz des implementierten Verfahrens in Virtual Reality-Umgebungen mit taktilen Ein- und Ausgabegeräten erfordert zudem einen adaptierbaren Detailgrad, um die Latenz zwischen Nutzer-Aktion und System-Reaktion so weit zu minimieren, dass der immersive Eindruck für alle unterstützten Sinnesorgane der Simulation erhalten bleibt.

Hierzu wird die kontinuierliche Netzvereinfachung reversibel durchgeführt und dabei eine progressive Netzrepräsentation erstellt, so dass eine schnelle Anpassung der Detailstufe an Randbedingungen der Simulation wie der Bild- oder Kraftwiederholrate automatisch möglich ist.

Abstract

In computer graphics, more and more detailed models are being used to increase the display quality. Depending on the application, the desired level of detail is, however, different so that the models should preferably be automatically simplified in order to save processor performance or memory space. This simplification of the triangle mesh under consideration should optically reproduce the original mesh as effectively as possible.

An extensive comparison of existing methods led to the choice of the quadric error metric as the metric used in the implementation for the error arising during simplification. This metric, described in detail here, is characterized by its generality, high quality and speed. The simplification is performed by continuous edge contraction that evaluates the geometric error via the quadratic distance of the mesh nodes of the simplified mesh to the planes through the corresponding triangular areas of the original model.

The application of the implemented method in virtual reality environments with tactile input and output devices also requires an adaptable level of detail in order to minimize the latency between user action and system reaction to such an extent that the immersive impression of the simulation is retained for all supported sensory organs.

To this end, continuous mesh simplification is reversibly performed and a progressive mesh representation created thus enabling a rapid automatic adaptation of the level of detail to the boundary conditions of the simulation such as the frame refresh or force update rate.

Inhaltsverzeichnis

1	Einleitung	1
2	Übersicht existierender Verfahren	3
2.1	Grundlagen	3
2.1.1	Netzaufbau	3
2.1.2	Einführung zum geometrischen Fehlermaß	6
2.2	Verfahrenseinteilung	17
2.3	Verfeinerung	18
2.3.1	Multiresolution Analysis of Arbitrary Meshes	18
2.4	Knotenhäufung	19
2.4.1	Multi-resolution 3D approximations for rendering complex scenes	19
2.5	Knotendezimierung	20
2.5.1	Decimation of triangle meshes	21
2.5.2	Simplification Envelopes	22
2.6	Kantenkontraktion	23
2.6.1	Mesh Optimization	24
2.6.2	Progressive Meshes	26
2.6.3	Surface Simplification Using Quadric Error Metrics	27
2.7	Vergleich und Bewertung der Verfahren	28
2.7.1	Eingabedaten	28
2.7.2	Datenstrukturen	30
2.7.3	Fehlermaße	32
2.7.4	Qualität und Geschwindigkeit	35
2.7.5	Zusammenfassung	38
2.8	Weitere Techniken	39
2.8.1	Geomorphs	39

2.8.2	Progressive Übertragung	40
2.8.3	Betrachterabhängigkeit	40
2.8.4	Modellierung an Vereinfachungen	42
2.8.5	Kompression	43
2.9	Aktuelle Forschungsgebiete	43
3	Vereinfachung mittels des Quadrik-Fehlermaßes	45
3.1	Verfahrensbeschreibung	45
3.1.1	Fortschreitende Knotenschmelzung	46
3.1.2	Wahl der Knotenschmelzungskandidaten	48
3.1.3	Kostenbestimmung der Kontraktionen	50
3.1.4	Bestimmung des Schmelzungspunktes	55
3.1.5	Einschränkungen	59
3.1.6	Speicherung der Simplifizierung	65
3.1.7	Adaptive Netzanpassung	68
3.1.8	Zusammenfassung des Algorithmus	70
3.2	Eigenschaften des Quadrik-Fehlermaßes	71
3.2.1	Konstruktion	73
3.2.2	Definitheit	76
3.2.3	Hauptachsentransformation	82
3.2.4	Allgemeiner Ebenenabstand als Norm	88
3.2.5	Klassifikation	91
3.2.6	Optimum	94
3.2.7	Oberflächenkrümmung	96
4	Ergebnisse	97
4.1	Einführung	97
4.2	Geschwindigkeit	99
4.2.1	Zeitkomplexität	99
4.2.2	Modelldaten	103
4.2.3	Empirische Laufzeiten	106
4.3	Speicherplatz	109
4.4	Qualität	112
4.4.1	Globale Fehler der Netzvereinfachung	117

4.4.2	Auswirkungen des Verschmelzungspunktes auf die globalen Fehler	119
4.4.3	Einfluss des Schwellwertes auf die globalen Fehler	124
4.4.4	Auswirkungen der Bestrafungen auf die globalen Fehler	126
4.5	Quadriken	128
4.6	Bestrafungen	131
4.7	Adaptive Netzanpassung	132
4.8	Probleme des Quadrik-Fehlermaßes	135
4.8.1	Unruhige offene Ränder	135
4.8.2	Spitzen	135
4.8.3	Imaginäre Kanten ohne Randbestrafung	136
4.8.4	Versetzte Dreiecke an einer Netzspalte	137
5	Zusammenfassung und Ausblick	139
5.1	Zusammenfassung	139
5.2	Ausblick	140
5.2.1	Selektive Vereinfachung	141
5.2.2	Gewichtung	142
5.2.3	Parallelisierung	143
Literatur		145
A	Implementationsbeschreibung	153
A.1	Voraussetzungen	153
A.1.1	SVT	153
A.1.2	Andere Softwarepakete	153
A.1.3	SENSITUS	154
A.1.4	Marching-Cube-Algorithmus	154
A.1.5	Hilfsmittel	155
A.1.6	Zeitschriften und Tagungen	155
A.2	Modulaufbau und Datenstrukturen	156
A.2.1	Modulaufbau	156
A.2.2	Datenstrukturen	159
A.3	Algorithmen	160
A.3.1	Abstand zwischen Punkt und Dreiecksnetz	160
A.4	Grafische Oberfläche	162

B	Beweise	165
B.1	Semidefinitheit der Untermatrix A des Quadrik-Fehlermaßes	165
C	Beispiele	167
C.1	Minima lokaler Fehlermaße	167
C.2	Konstruktion einer Ellipse	171
C.3	Berechnung des Quadrik-Fehlermaßes	172
C.4	Anzahl von zusätzlichen Knotenpaaren aufgrund eines Schwellwertes	173
D	Glossar	175
E	Danksagung	179

Abbildungsverzeichnis

1.1	Einpassung an einem Virtual Reality-System	1
2.1	Arten von Mannigfaltigkeiten	5
2.2	Approximationen mit Rauschen und starker Abweichung	7
2.3	Maximale Netzabstände	8
2.4	Knotenfreier maximaler Netzabstand	9
2.5	Strukturunverträglichkeit globaler Fehler	10
2.6	Abstandsberechnung ohne Geradenstücke	11
2.7	Ebenenabstand als Semi-Norm, Norm oder Fehlermaß	13
2.8	Isolinien lokaler Fehlermaße	15
2.9	Isolinien von globalen Fehlermetriken und dem Quadrik-Fehlermaß	16
2.10	Stufen des Vereinfachungsverfahrens durch Verfeinerung	18
2.11	Knotenhäufung im Zweidimensionalen	20
2.12	Abstand zur Durchschnittsebene oder zum Rand als Fehlermaß	21
2.13	Schnittebene zur Untersuchung der Knotenlage	21
2.14	Einhüllende Netze	22
2.15	Verringerung der Hüllenverschiebung	22
2.16	Elementartransformationen	25
2.17	Projektion und Minimierung der Abstandskvadratsumme und der Federkräfte	25
2.18	Kantenkontraktion und Knotenspaltung	26
2.19	Modellierte Oberfläche und Punktwolke eines 3D-Scans	28
2.20	Vereinfachung einer Geländeoberfläche mittels des Quadrik-Fehlermaßes	28
2.21	Knotenbaum einer kantenkontrahierenden Vereinfachung	31
2.22	Regularisierung eines Dreiecksnetzes	31
2.23	Optimale und positionstreue Vereinfachung	32
2.24	Bildfehler als Fehlermaß	34

2.25	Maximale und durchschnittliche Fehler fünf vorgestellter Verfahren	36
2.26	Zusammenhang von Geschwindigkeit und Qualität	37
2.27	Betrachterabhängige Triangulierung eines Höhenfeldes	41
2.28	Betrachterabhängige Netzverfeinerung im Bereich der Silhouette	42
2.29	Modellierung an Vereinfachungen	42
3.1	Kantenkontraktion und kantenungebundene Knotenverschmelzung	46
3.2	Vereinfachung dicht beieinanderliegender Würfel mit und ohne Schwellwert	48
3.3	Knotenpaarkandidaten	49
3.4	Kostenbestimmung über den Quadratabstand zu den Geraden	51
3.5	Detailstufen einer mit dem Quadrik-Fehlermaß simplifizierten Kuh	54
3.6	Würfelvereinfachung ohne Kosten	59
3.7	Vereinfachung einer gekrümmten Fläche mit und ohne Randbestrafung	61
3.8	Vereinfachung eines geschwungenen Blattes mit und ohne Randbestrafung	61
3.9	Kuhmodell mit gewichteten Hörnerspitzen	62
3.10	Bestrafung der Dreiecksidentität	63
3.11	Kantenkollabierung mit Oberflächenfaltung	64
3.12	Minimalbeispiel einer Normalendrehung	65
3.13	Knotenverschmelzung und Knotenspaltung	66
3.14	Tetraeder der Volumendifferenzmethode	73
3.15	Beispiel einer singulären Fehlerquadrik	78
3.16	Beispiel einer Transformation in Hauptachsenform	86
3.17	Ebenenkonstellationen für verschiedene Klassen von Quadriken	92
3.18	Fehlerellipsoide an vereinfachten Modellen	96
4.1	Vereinfachung mittels Marching-Cube-Algorithmus	98
4.2	Kanne mit unterschiedlichen Visualisierungsmethoden	98
4.3	Telefonhörer mit flacher und Gouraud-Schattierung	99
4.4	Schlimmster Fall für die Berechnungskosten der Vereinfachung	100
4.5	Verhältnis von Dreiecken und Kanten zu Knoten	101
4.6	Zeitunterschiede durch unterschiedlich aktivierter Bestrafungen	106
4.7	Geschwindigkeit der Vereinfachung	107
4.8	Geschwindigkeit des Lesens und Schreibens	108
4.9	Hauptspeicherbedarf von Datenstrukturen des Dreiecksnetzes	110

4.10	Hauptspeicherbedarf des Programms nach Einlesen und Vereinfachung	111
4.11	Dateigröße im SMF- und PM-Format	112
4.12	Detailstufen des “Stanford Bunnys”	114
4.13	Detailstufen eines Flugzeugs	114
4.14	Detailstufen einer Isofläche einer Elektronendichtewolke	114
4.15	Vergleich globaler Fehler eines einfachen und eines komplexen Modells	115
4.16	Detailstufen eines Drachens in unterschiedlichen Entfernungen	116
4.17	Globale Fehler der Detailstufen des Kuhmodells	117
4.18	Globale Fehler für ausgewählte Verschmelzungskandidaten	119
4.19	Vergleich globaler gerichteter Fehler verschiedener Verschmelzungskandidaten . . .	121
4.20	Prozentuale Fehlerdifferenz für die optimale Knotenwahl	122
4.21	Detailstufen des Kuhmodells mit unterschiedlicher Knotenwahl	123
4.22	Globaler Fehler und Knotenpaaranzahl in Abhängigkeit vom Knotenabstandswert . .	124
4.23	Vereinfachung menschlicher Fußknochen mit und ohne Schwellwert	124
4.24	2D-Knotenkonstellation mit Schwellwert bedingter Kante	125
4.25	Globale Fehler in Abhängigkeit von der Randbestrafung	127
4.26	Globale Fehler in Abhängigkeit von der Identitätsbestrafung	128
4.27	Anteil singulärer Quadriken	129
4.28	Bestrafungen pro Knoten der untersuchten Modelle	131
4.29	Knotenanzahl während des Ladens und der Vereinfachung	132
4.30	Knotenanzahl während der adaptiven Netzanpassung	133
4.31	Bildwiederholrate in verschiedenen Detailstufen	134
4.32	Schwer zu vereinfachende unruhige offene Ränder	135
4.33	Singuläre Quadriken an scharfen Ecken	136
4.34	Elimination einer Spitze	136
4.35	Von der Oberfläche entfernender Verschmelzungspunkt einer imaginären Kante . . .	137
4.36	Versetzte Netzspalte	137
5.1	Vereinfachung mit regionaler Gewichtung	141
5.2	Gewichtung des Quadrik-Fehlermaßes	142
A.1	Anwendung SENSITUS	154
A.2	Äquivalenzklassen zur Triangulierung	154
A.3	UML-Diagramm der grafischen Oberflächen	157

A.4	UML-Diagramm des Netzes und dessen Vereinfachung	158
A.5	Beziehungen wichtigster Klassen des Dreiecksnetzes	159
A.6	Raumaufteilung zur Abstandsbestimmung zwischen Punkt und Dreieck	160
A.7	Grafische Oberfläche der Anwendung SENSITUS	163
C.1	Geraden und Isolinien lokaler Fehlermaße	167
C.2	Konstruktion einer Ellipse	171
C.3	Beispiel zur Kostenbestimmung im Zweidimensionalen über den Quadratabstand . . .	172
C.4	Bestimmung der Knotenpaaranzahl in Abhängigkeit des Schwellwertes	173

Tabellenverzeichnis

2.1	Eigenschaften der Verfahren	38
3.1	Klassifikation der Quadriken	94
4.1	Eigenschaften ausgewählter Modelle	104
4.2	Modelleigenschaften während der Simplifizierung	104

1

Einleitung

Für möglichst realitätsnahe genaue Darstellungen in der Computergrafik werden immer komplexere, detailreichere Modelle herangezogen. Die gewünschte Detailliertheit schwankt aber von Anwendung zu Anwendung. Um Prozessorleistung oder Speicherplatz zu sparen, sollen daher die Modelle je nach Anforderung möglichst automatisch vereinfacht werden.

Vereinfachung heißt dabei, mit möglichst wenig Daten das Modell so gut wie möglich wiederzugeben. Das Modell wird vornehmlich durch seine Oberfläche und diese wiederum durch ein Netz von Dreiecken bestimmt. Da die menschliche Wahrnehmung sehr komplex ist, wird als Maß für die Qualität der geometrischen Vereinfachung meist

in irgendeiner wenn auch indirekten Form der geometrische Abstand des vereinfachten Netzes zum Originalnetz genommen. Bei betrachterabhängiger Vereinfachung kommen auch andere Fehlermaße zum Einsatz, die den Darstellungsfehler am Bildschirm mit möglichst wenig Dreiecken zu optimieren versuchen.

Die Modellkomplexität steht im direkten Zusammenhang mit den Rechenkosten für die Darstellung. Daher ist zum Beispiel für das interaktive Arbeiten eine Vereinfachung nötig, um die Bildwiederholrate je nach Bildänderung auf 10 bis 25 Bilder pro Sekunde erhöhen zu können, so dass die Bewegtdarstellung flüssig wird. Hingegen ist für die genaue Analyse von einer statischen Betrachterposition eine möglichst genaue Darstellung wünschenswert.

Die Notwendigkeit der Simplifizierung ergab sich bei der auf dem Grafikpaket SVT aufsetzenden

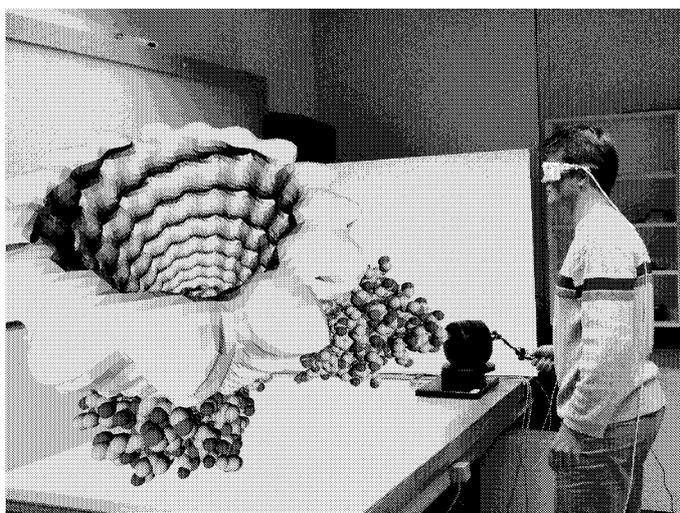


Abb. 1.1: Einpassung einer Molekülstruktur in eine Elektronendichtewolke an einem Virtual Reality-System mittels Krafrückkopplung (Fotomontage)

Anwendung SENSITUS, die mittels Krafrückkopplung das interaktive Einpassen von exakten Molekülstrukturen in niedrigaufgelöste Aufnahmen von Elektronendichtewolken ermöglicht. Zum einen generiert der zur Anwendung kommende Marching-Cube-Algorithmus bei feinen Messungen eine große Zahl von Dreiecken und zum anderen benötigt die flüssige Ausgabe der Kraft viel Rechenzeit, die der Visualisierung nicht mehr zur Verfügung steht. Des Weiteren geschieht die Visualisierung in einer Virtual Reality-Umgebung, für die die Latenz zwischen Nutzer-Aktion und System-Reaktion wichtig für den immersiven Eindruck der Simulation ist. Hierzu erweitert das in dieser Arbeit beschriebene Modul das Grafikpaket SVT und damit SENSITUS um die Möglichkeit der adaptiven automatischen Netzvereinfachung.

Zur Vereinfachung von Dreiecksnetzen gibt es eine Vielzahl von Verfahren. In dieser Arbeit wird die Reduktion der Netzkomplexität durch das iterative Zusammenziehen zweier auf einer Kante oder nah beieinander liegender Knoten auf einen Knoten bewirkt. Als Fehlermaß für diese Operation wird das Quadrik-Fehlermaß benutzt, das die Vereinfachung über den quadratischen Abstand der Netzknoten des simplifizierten Netzes zu Ebenen durch die Dreiecksflächen des Originalmodells bewertet. Um die Detailstufen des Modells, die durch die sukzessive Vereinfachung erzeugt werden, reversibel zu speichern, wird eine Liste der Netzoperationen angelegt. Da diese Liste auch wieder eine Verfeinerung erlaubt, kann das Modell adaptiv an die Bedürfnisse der Visualisierung angepasst werden.

Im ersten Teil dieser Arbeit werden einige der bekanntesten Ansätze zur Netzvereinfachung mit ihren Vor- und Nachteilen vorgestellt, um anschließend das wegen seiner Effizienz, Qualität und Generalität ausgewählte Verfahren und die eigenen Erweiterungen genauer zu beschreiben. Die Eigenschaften der für das Fehlermaß wichtigen Quadriken werden ausführlich analysiert. Im Weiteren wird auf die Resultate der ebenfalls erläuterten Implementation eingegangen. Der Ausblick auf mögliche Erweiterungen rundet die Arbeit ab.

2

Übersicht existierender Verfahren

Dieses Kapitel bespricht existierende Verfahren zur Erstellung von Netzvereinfachungen. Um diese zu verstehen und gemäß Abschnitt 2.2 eingruppiert zu können, werden in 2.1 grundlegende Begriffe vorgestellt. Daran anschließend werden einige wichtige Verfahren in den Abschnitten 2.3 bis 2.6 gruppiert vorgestellt, um in 2.7 verglichen zu werden. Abschnitt 2.8 gibt einen Einblick in weitere Techniken und 2.9 einen Ausblick auf aktuelle Forschungsgebiete.

2.1 Grundlagen

Für das bessere Verständnis seien im Folgenden Grundlagen und einige wiederkehrende Begriffe erläutert. Geometrische Fehlermaße werden eingeführt und motiviert. Weitere möglicherweise unbekannte Begriffe in dieser Arbeit sind mit einem hochgestellten “G” gekennzeichnet und werden im Glossar im Anhang D beschrieben.

2.1.1 Netzaufbau

Diese Arbeit beschränkt sich auf *diskrete Daten*, die heutzutage den Großteil der darzustellenden Objekte beschreiben. Die Approximationselemente zwischen den Daten können im allgemeinen stückweise linear, höhere polynomiale Funktionen oder andere Basisfunktionen sein [1]. Die nichtlinearen Approximationselemente erzeugen eine glattere Gesamtoberfläche. Bei diskreten Daten, über die hinaus man keine weiteren Informationen von der Oberfläche hat, täuscht die Glattheit aber nur exaktere Daten vor. Daher werden die nichtlinearen Basisfunktionen meist bei der Modellierung von Objekten bei einer eher geringen Zahl an Stützstellen genutzt. In der dieser Arbeit zugrundeliegenden Anwendung hingegen liegen hochaufgelöste diskrete Daten vor, so dass eine lineare Oberflächenapproximation zur Visualisierung herangezogen wird.

Bei den linearen Approximationselementen der Polygonnetze^G beschränkt sich diese Ausarbeitung auf *Dreiecksnetze*, da sich alle Polygonnetze in einem Vorverarbeitungsschritt auf Dreiecksnetze abbilden lassen (Triangulation^G) [2, 3]. Diese einfachste Art der Netzpräsentation ist am weitesten verbreitet und steht meist am Ende einer jeden Oberflächenvisualisierung, da diese Repräsentation von

fast aller Visualisierungssoftware und am besten von heutiger Grafikhardware unterstützt wird. Zudem existiert auf dieser Basis eine Vielzahl von Modellen.

Intuitiv richtig ist die Vorstellung eines Dreiecksnetzes als stückweise lineare Fläche, die aus an den Kanten verbundenen Dreiecken zusammengesetzt ist.

Theoretisch können Dreiecksnetze als *zweidimensionales schlichtes Simplex* (simplicial complex) eingeführt werden. Ein k -Simplex $\sigma^k = v_0 v_1 \dots v_k$ ist die kleinste konvexe Hülle^G (convex hull), die durch $k+1$ linear unabhängige Punkte bzw. Knoten v_i , $i \in \mathbb{N}_k = \{0, \dots, k\}$ definiert wird. Beispiele dafür sind Knoten (0-Simplex), Geradenstücke (1-Simplex) und Dreiecke (2-Simplex). Jeder Punkt p auf einem Simplex kann durch konvexe Kombination der Simplex-Knoten erreicht werden:

$$p = \sum_{i=0}^k t_i v_i \quad \text{mit} \quad \sum_{i=0}^k t_i = 1, \quad t_i \in [0, 1].$$

Jeder Simplex einer Untermenge an Knoten eines Simplex wird als Subsimplex bezeichnet. Ein zweidimensionales schlichtes Simplex S und damit ein allgemeines Dreiecksnetz besteht nun aus einer Menge von Knoten, Kanten (Geradenstücke) und Dreiecken, die folgenden Bedingungen genügen müssen:

1. Wenn $\sigma_i, \sigma_j \in S$, dann besitzen sie entweder keine Schnittmenge oder schneiden sich lediglich in einem gemeinsamen Subsimplex. Zum Beispiel können sich zwei Kanten nur in einem Knoten und zwei Dreiecke in einer Kante oder einem Knoten schneiden.
2. Wenn $\sigma_i \in S$, dann müssen auch alle Subsimplexe von σ_i in S liegen.

Ein *reiner Komplex* (pure complex) schränkt das zweidimensionale schlichte Simplex dahingehend ein, dass keine isolierten Kanten und Knoten zugelassen werden.

Zur Netzbeschreibung werden folgende Bezeichnungen herangezogen. $X = \{x_1, \dots, x_n\}$ bezeichnet eine Punktwolke, deren Knotenauswahl $V = \{v_1, \dots, v_m\}$ die *Geometrie* des Netzes bestimmt. K ist die *Topologie* und das aus Geometrie und Topologie zusammengesetzte Netz lautet $M = (K, V)$. $\phi_V(|K|)$ bezeichnet die geometrische Umsetzung der Topologie (geometric realization) des Netzes (K, V) , also die *Netzoberfläche*, und wird kurz auch $\phi(M)$ geschrieben. Die *Netzkomplexität* $|V|$ gibt die Anzahl der sich noch im Dreiecksnetz befindlichen Knoten wieder.

Für *orientierte Netze* hat die Oberfläche eine innere und äußere Seite. Es spielt bei diesen Netzen also eine wichtige Rolle, in welche Richtung die Dreiecksnormalen des Netzes zeigen. Oft wird die Orientierung der Dreiecke implizit durch die Reihenfolge der Dreiecksknoten angegeben, wobei von der äußeren Seite betrachtet die Knoten gegen den Uhrzeigersinn angegeben werden.

Bei den Simplifizierungsverfahren wird zwischen *topologieerhaltenden* und *topologieverändernden* Algorithmen unterschieden. Die Topologie beschreibt dabei die Netzstruktur und ist gegenüber affinen Transformationen invariant und damit losgelöst von der geometrischen Repräsentation. Eigenschaften der Topologie sind zum Beispiel der Zusammenhang von Knoten, die Zahl von zusammenhängenden Komponenten, die Inzidenz von Kanten an Knoten oder das Geschlecht. Das *Geschlecht* oder die Anzahl an Henkeln bezeichnet die Menge an Löchern im Netz. Eine Kugel hat zum Beispiel das Geschlecht 0 und ein Ring oder eine Tasse mit einem Henkel das Geschlecht 1. Körper mit dem gleichen Geschlecht sind *topologisch äquivalent* bzw. *homöomorph*. Die *lokale Topologie* der Dreiecke, Kanten und Knoten beschreibt die Netzeigenschaften in deren unmittelbaren Nachbarschaft. Topologieerhaltende Verfahren schließen somit keine Löcher, was topologieverändernde hingegen zulassen

und darüber hinaus auch unverbundene Netzteile zu einem Netz verschmelzen lassen können. *Topologietolerante* Methoden sind dadurch gekennzeichnet, dass sie als Eingabedaten beliebige Topologien wie Nicht-Mannigfaltigkeiten zulassen.

Eine *zweidimensionale Mannigfaltigkeit* (manifold) ist definiert als eine zusammenhängende Menge, in der jeder Punkt eine Umgebung der Menge besitzt, die topologisch äquivalent zu einer offenen Scheibe des \mathbb{R}^2 ist [4]. Die lokale Topologie jedes Punktes muss also diese Äquivalenz aufweisen. Da diese Arbeit sich ausschließlich mit zweidimensionalen Mannigfaltigkeiten befasst und die zitierte Literatur aus diesem Bereich die dimensionslose Formulierung benutzt, wird im Folgenden auf die Dimensionsangabe verzichtet und nur von Mannigfaltigkeit gesprochen. Ein Beispiel einer Mannigfaltigkeit ist in Abbildung 2.1 (a) zu sehen. Punkte einer Menge, die der lokalen Topologie einer Mannigfaltigkeit nicht genügen, werden *kritische Punkte* genannt. Solche Oberflächen werden *Nicht-Mannigfaltigkeiten* (non-manifold) genannt. Ein kritischer Punkt ist v in Abbildung 2.1 (c) und die Oberfläche aus (c) eine Nicht-Mannigfaltigkeit. Bei einer *Mannigfaltigkeit mit Rand* darf die Umgebung eines Punktes auch topologisch äquivalent zu einem Halbkreis sein. Auf der dicken Linie in Abbildung

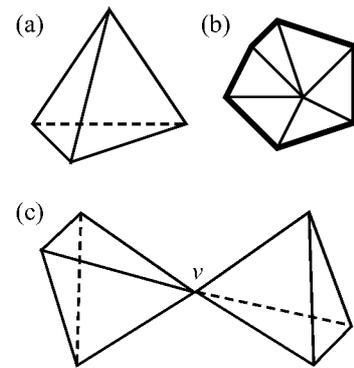


Abb. 2.1: (a) Mannigfaltigkeit, (b) Mannigfaltigkeit mit Rand und (c) Nicht-Mannigfaltigkeit mit kritischem Punkt v

2.1 (b) besitzen die Dreieckskanten keine Nachbardreiecke und erzeugen somit einen Rand, dessen Punkte Umgebungen besitzen, die topologisch äquivalent zu Halbkreisen sind. (b) ist somit eine Mannigfaltigkeit mit Rand. Für ein Dreiecksnetz heißt dies für die oben eingeführten Begriffe, dass jede Kante für eine Mannigfaltigkeit inzident genau zu zwei und für eine Mannigfaltigkeit mit Rand inzident zu ein oder zwei unterschiedlichen Dreiecken sein muss. Zudem darf der Ring der Dreiecke um jeden Knoten nur einmal mit Rand und keinmal ohne Rand unterbrochen werden. Die globale Topologie bleibt erhalten, wenn die lokale erhalten bleibt und es keine Durchdringungen des Netzes gibt.

Netze können neben den geometrischen Daten auch *Attribute oder Eigenschaften* (features, properties) besitzen. Hier lassen sich zwei Arten unterscheiden [5]. Zum einen die *diskreten Attribute*, die meist mit den Flächen verknüpft sind und für diese überall gleich sind (z.B. Materialeigenschaften), und zum anderen die *skalaren Attribute*, welche im einfachsten Fall mit den Knoten verknüpft sind (z.B. Normale, Farbe, Texturkoordinaten). Mit skalar ist nicht die Dimension der Eigenschaft gemeint, sondern soll im Gegensatz zu den diskreten Attributen ausdrücken, dass sich die Eigenschaft über die beteiligte Fläche verändern kann, zum Beispiel linear interpoliert von Netzknoten zu Netzknoten. Die skalaren Attribute können aber auch auf Ecken liegen. *Ecken* sind Tupel aus einem Knoten und einer Fläche. Dadurch können an demselben Knoten von verschiedenen Flächen aus unterschiedliche skalare Attribute vorkommen. Ein Beispiel dafür wären scharfe Kanten, an denen das Modell über zwei Ecken entlang einer Kante einen Normalensprung aufweist. Diskontinuität liegt also in Bereichen vor, deren Flächen unterschiedliche diskrete Attribute besitzen oder aber sich die skalaren Attribute in einem Knoten unterscheiden. Kanten, deren anliegenden Flächen sich stark in ihren Attributen unterscheiden, werden *innere Kanten* (interior edge) genannt. Knoten, an denen mehr als zwei innere Kanten aufeinandertreffen, heißen *spitze Knoten*. In der dieser Arbeit zugrundeliegenden Anwendung gibt es keine Netzeigenschaften, so dass diese in der weiteren Betrachtung nur eine untergeordnete Rolle spielen.

Für Dreiecksnetze, die topologisch äquivalent zu einer Kugel sind, gilt die einfache *Eulersche Formel*

$$\#V - \#E + \#F = 2,$$

die ganz allgemein für Polyeder^G gilt ($\#V$: Anzahl Knoten (vertices), $\#E$: Anzahl Kanten (edges), $\#F$: Anzahl Flächen (faces)). Da in diesem Fall jeder Kante genau zwei Dreiecke und jedem Dreieck drei Kanten zugeordnet sind, besitzt solch ein Dreiecksnetz in etwa zweimal so viele Flächen und dreimal so viele Kanten wie Knoten. Dies ist auch gut an der topologieerhaltenden inversen elementaren Operation zur Kantenkontraktion aus Abschnitt 2.6.2 zu erkennen, die genau einen Knoten, zwei Dreiecke und drei Kanten in ein bestehendes Netz einfügt.

Besitzt das Modell Henkel oder besteht es aus mehreren Zusammenhangskomponenten, ist aber eine Mannigfaltigkeit, gilt die *erweiterte Eulersche Formel* bzw. die *Euler-Poincaré Formel* [4, 6, 7]. Die *Zusammenhangskomponenten* (connected components) sind die miteinander verbundenen Teile eines Netzes. Diese einzelnen Teilnetze dürfen dabei nicht einmal über nur einen Knoten verbunden sein. Die oben gemachte Abschätzung für das Verhältnis von Knoten, Kanten und Dreiecken zueinander bleibt aber bei gewöhnlichen Netzen, deren Anzahl an Zusammenhangskomponenten und Henkeln um Größenordnungen unter der Anzahl an Netzprimitiven bleibt, bestehen. Werden die Euler-Formeln von einem Netz nicht erfüllt, so kann daraus geschlossen werden, dass es keine Mannigfaltigkeit ist, andersherum resultiert aber aus der Erfüllung der Formeln nicht, dass das Dreiecksnetz eine Mannigfaltigkeit beschreibt.

2.1.2 Einführung zum geometrischen Fehlermaß

Norm, Definition:

Sei V ein Vektorraum über \mathbb{R} , so ist die Abbildung $\|\cdot\| : V \rightarrow \mathbb{R}_0^+$ eine *Norm*, wenn folgende Bedingungen erfüllt sind:

1. $\|x\| > 0 \quad \forall x \in V \setminus \{0\}$ (Definitheit),
2. $\|\alpha x\| = |\alpha| \|x\| \quad \forall x \in V, \alpha \in \mathbb{R}$ (Homogenität),
3. $\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in V$ (Dreiecksungleichung).

Für den hier betrachteten \mathbb{R}^3 sind alle Normen äquivalent. Zwei Normen $\|\cdot\|$ und $\|\cdot\|^*$ heißen *äquivalent*, wenn

$$\exists \alpha, \beta \in \mathbb{R}^+ : \alpha \|x\| \leq \|x\|^* \leq \beta \|x\| \quad \forall x \in V.$$

Metrik, Definition:

Sei V ein Vektorraum über \mathbb{R} , so ist die Abbildung $d : V \times V \rightarrow \mathbb{R}_0^+$ eine *Metrik*, wenn folgende Bedingungen $\forall x, y, z \in V$ erfüllt sind:

1. $d(x, y) = 0 \Leftrightarrow x = y$ (Definitheit bzw. Identitätseigenschaft),
2. $d(x, y) = d(y, x)$ (Homogenität bzw. Symmetrie),
3. $d(x, z) \leq d(x, y) + d(y, z)$ (Dreiecksungleichung).

Jede Norm $\|\cdot\|$ induziert eine Metrik d gemäß

$$d(x, y) = \|y - x\|.$$

Die durch die L_p -Normen induzierten Metriken zwischen zwei auf dem Intervall $[a, b]$ stetigen Funktionen f und g ,

$$\|f - g\|_p = \sqrt[p]{\int_a^b |f(x) - g(x)|^p dx}, \quad p \in [1, \infty),$$

die wir hier kurz L_p -Metriken nennen, sollen als Ausgangsbasis für den *geometrischen Fehler* im Zweidimensionalen dienen. Neben der (*Betrag*-)Summen-Metrik ($p = 1$) sind die *Maximum-* bzw. *Supremum-Metrik* ($p = \infty$) und die *Euklidische Metrik* ($p = 2$) die gebräuchlichsten. Um den maximalen Abstand einer Funktion $f(x)$ von ihrer Approximation $g(x)$ auf dem Intervall $[a, b]$ zu bestimmen, dient die Maximum-Metrik, die aufgrund von Grenzwertbetrachtungen für $p = \infty$

$$\|f - g\|_\infty = \max_{a \leq x \leq b} |f(x) - g(x)|$$

lautet. Den durchschnittlichen Abstand gibt die Euklidische Metrik

$$|f - g| = \|f - g\|_2 = \sqrt{\int_a^b (f(x) - g(x))^2 dx}$$

wieder [8].

Hierbei ist die Maximum-Metrik die strengere Metrik, da sie eine globale absolute Grenze für den Abstand beider Funktionen angibt, hingegen die Euklidische Metrik besser das Verhalten der gesamten Approximation wiedergibt. Es gilt $L_\infty \geq L_2$. L_∞ ist anfällig gegenüber großen lokalen Abweichungen (peaks), wohingegen L_2 letzteren Fehler geringer bewertet als Rauschen bzw. gleichmäßige Störungen (noise). Abbildung 2.2 verdeutlicht dies: die Approximation (a) beeinflusst die Euklidische Metrik wesentlich mehr als die Maximum-Metrik ($L_2 > 0$, $L_\infty \approx L_2$); umgekehrt zeigt sich bei der Approximation (b) die Maximum-Metrik anfälliger und die Euklidische Metrik stabiler ($L_2 \approx 0$, $L_\infty \gg 0$).

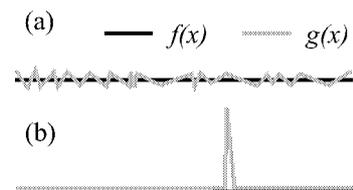


Abb. 2.2: Approximation $g(x)$ von $f(x)$ mit (a) Rauschen und (b) einer lokalen starken Abweichung

Beide Metriken haben also ihre Stärken und Schwächen. Eine Kombination wäre bei dem Einsatz zum Netzvergleich vorzuziehen: die Approximation sollte einen möglichst geringen L_2 -Fehler haben und einen Grenzwert, der über die L_∞ -Metrik definiert ist, nicht überschreiten. Für bestimmte Anwendungen wie der Kollisionserkennung (collision detection) hingegen ist die L_∞ die zu bevorzugende Metrik.

Metriken für Netzabstände

L_∞ und L_2 können auch auf Oberflächen übertragen werden. Da aber nicht wie bei den Funktionen eine eindeutige senkrechte Abstandrichtung vorliegt, muss der Abstand zwischen Netzen erst definiert werden. Sei M ein Netz und $\phi(M)$ dessen Oberfläche, so sei der Abstand eines Punktes v zu $\phi(M)$ gegeben durch die geringste Entfernung des Punktes zur Netzoberfläche

$$d_v(M) = \min_{w \in \phi(M)} \|v - w\|_2.$$

Die Berechnung von $d_v(M)$ für ein Dreiecksnetz wird im Anhang A.3.1 genauer beschrieben. Da der Abstand immer positiv ist, kann im Folgenden der Betrag innerhalb der Metriken entfallen.

Die zur L_∞ korrespondierende geometrische Fehlermetrik ist die *beidseitige Hausdorff-Metrik* [9]

$$\hat{E}_{max}(M_1, M_2) = \max \left(\max_{v \in \phi(M_1)} d_v(M_2), \max_{v \in \phi(M_2)} d_v(M_1) \right),$$

welche den maximalen Abstand beider Netze zueinander angibt und somit gewährleistet, dass die Netze sich jeweils in einer Umgebung von $\hat{E}_{max}(M_1, M_2)$ zueinander befinden. Sie wird auch *zweiseitiger* oder *symmetrischer Hausdorff-Abstand* genannt. Im Folgenden wird dieser auch ohne Zusatz zweiseitig, symmetrisch oder beidseitig benutzt. Der Hausdorff-Abstand wird als *einseitig* oder *gerichtet* bezeichnet, wenn er nur in eine Richtung betrachtet wird.

L_2 hat seine Entsprechung in dem *durchschnittlichen quadratischen geometrischen Abstand*

$$\hat{E}_{avg}(M_1, M_2) = \frac{1}{w_1 + w_2} \left(\int_{\phi(M_1)} d_v^2(M_2) dv + \int_{\phi(M_2)} d_v^2(M_1) dv \right),$$

wobei w_1 und w_2 die Oberflächengrößen von M_1 und M_2 sind. Die Normalisierung durch $1/(w_1 + w_2)$ könnte auch getrennt für jedes Integral mit der eigenen Oberflächengröße durchgeführt werden, was zu einer etwas anderen Gewichtung der Integralanteile führen würde.

Es ist zu beachten, dass es nicht genügt, den Abstand jedes Punktes auf $\phi(M_1)$ zu seinem nächsten Punkt auf M_2 zu bewerten, sondern dass auch der Abstand jedes Punktes auf $\phi(M_2)$ zu seinem nächsten Punkt auf M_1 zu bewerten ist. Im Zweidimensionalen verdeutlicht dies Abbildung 2.3. Hier ist

$$(a) \quad \max_{v \in \phi(M_1)} d_v(M_2) \quad \text{und} \quad (b) \quad \max_{v \in \phi(M_2)} d_v(M_1).$$

Es ist klar ersichtlich, dass die Betrachtung nur eines Maximums von beiden, den falschen Hausdorff-Abstand ergeben könnte, der in diesem Fall (b) ist. (a) ist im übrigen nicht eindeutig und hat auch zum rechten Knoten des unteren Netzes den gleichen Abstand. Ähnliche Überlegungen kann man auch für den durchschnittlichen Quadratabstand oder Abstand beider Netze anstellen.

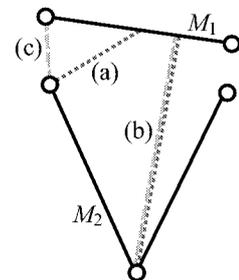


Abb. 2.3: Maximaler Netzabstand (a) von M_1 zu M_2 , (b) von M_2 bzw. von den Knoten von M_2 zu M_1 und (c) von den Knoten von M_1 zu M_2

Vereinfachte Maße

Da die exakte Berechnung der Netzabstände sehr aufwendig ist, wird oft ein *vereinfachtes Maß* genommen. Statt ein gesamtes Netz mit einem anderen zu vergleichen, werden nur repräsentierende diskrete Punkte auf dem Netz zur Abstandsbestimmung herangezogen. Daher werden die Maße auch *diskret* genannt. Spezielle Implementierungen zur Abstandsberechnung von Netzen wie die Programme Mesh [9] oder Metro [10] erlauben benutzerdefinierte Einteilungen der Dreiecksflächen zur Bestimmung der diskreten Messpunkte. Im einfachsten Fall sind die repräsentierenden Punkte die Knoten $V \subset \phi(M)$ des Netzes. Diese wurden auch bei der eigenen Implementation zur Angabe der geometrischen Fehler in Kapitel 4 herangezogen. Sei $k_i = |V_i|$, $i \in \{1, 2\}$ die Knotenanzahl, dann lauten die vereinfachten Fehlermaße

$$E_{max}(M_1, M_2) = \max \left(\max_{v \in V_1} d_v(M_2), \max_{v \in V_2} d_v(M_1) \right)$$

und

$$E_{avg}(M_1, M_2) = \frac{1}{k_1 + k_2} \left(\sum_{v \in V_1} d_v^2(M_2) + \sum_{v \in V_2} d_v^2(M_1) \right).$$

In Abbildung 2.3 sind auch die beiden Teilergebnisse des vereinfachten Hausdorff-Abstandes eingetragen:

$$(c) \max_{v \in V_1} d_v(M_2) \quad \text{und} \quad (b) \max_{v \in V_2} d_v(M_1).$$

(b) ist hier identisch mit dem Teil des generellen Hausdorff-Abstandes. $E_{max}(M_1, M_2)$ ist in diesem Fall wiederum (b), was aber nicht immer der Fall sein muss.

Hierzu betrachte man in Abbildung 2.4 den durch eine gestrichelte Linie angegebenen maximalen Abstand beider Netze, der nicht zwischen zwei Netzknoten liegt und damit nicht über die vereinfachten Maße erreicht werden kann.

Einen weiteren Abstandswert gibt

$$E_{sum}(M_1, M_2) = \frac{1}{k_1 + k_2} \left(\sum_{v \in V_1} d_v(M_2) + \sum_{v \in V_2} d_v(M_1) \right)$$

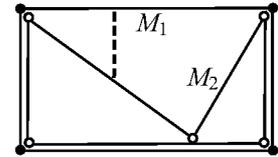


Abb. 2.4: Gestrichelte Linie als maximaler Netzabstand, der nicht zwischen zwei Knoten liegt

wieder, was einem *gemittelten Summen-Maß* entspricht. Dieses stimmt mit der geometrischen Vorstellung des durchschnittlichen geometrischen Abstandes überein.

Anhand Abbildung 2.4 ist auch zu erklären, warum von den generellen Metriken, aber von vereinfachten Maßen gesprochen wird. In dieser Arbeit werden die Begriffe Norm und Metrik im mathematischen Sinn benutzt, wohingegen *Maß*, *Abstand*, *Distanz* und *Entfernung* im Sprachsinn zu verstehen sind. Würden in der Abbildung die nah beieinanderliegenden parallelen Kanten identisch sein, und auch der einzelne Knoten des feineren Netzes genau auf der Kante liegen, würde für die vereinfachten Maße

$$E_{max}(M_1, M_2) = E_{avg}(M_1, M_2) = E_{sum}(M_1, M_2) = 0$$

gelten. Da für Metriken Null nur für $M_1 = M_2$ auftreten darf, sind E_{max} , E_{avg} und E_{sum} keine Metriken und werden in dieser Arbeit wie auch alle folgenden heuristischen Bewertungen von Netzabständen als Maße bezeichnet.

Für die Metriken \hat{E}_{max} und \hat{E}_{avg} ist es wichtig, dass der beidseitige Abstand betrachtet wird, da sonst die Symmetrie bzw. Kommutativität der Metriken verletzt würde. Zudem ist die Definition identischer Netze wichtig. Damit \hat{E}_{max} und \hat{E}_{avg} Metriken im mathematischen Sinn sind, müssen alle Netze als identisch betrachtet werden, deren Oberflächen identisch sind, ansonsten würden unterschiedliche Triangulationen^G der gleichen Oberfläche den Abstand Null besitzen und als unterschiedlich zu bewertende Netze aufgefasst der Bedingung einer Metrik widersprechen, nur Null für identische Objekte sein zu dürfen. Ganz korrekt müssten die Metriken also auf der geometrischen Repräsentation definiert werden ($\hat{E}_{max}(\phi_{V_1}(|K_1|), \phi_{V_2}(|K_2|))$ bzw. $\hat{E}_{avg}(\phi_{V_1}(|K_1|), \phi_{V_2}(|K_2|))$) und nicht auf den Netzen selbst, was hier der Einfachheit halber durch \hat{E}_{max} bzw. \hat{E}_{avg} ausgedrückt werden soll. Einen Beweis dafür, dass es sich um Metriken handelt, findet man in [11].

Bewertungen des implementierten Vereinfachungsalgorithmus mittels der vereinfachten Maße stehen in Abschnitt 4.4.1 und 4.4.3.

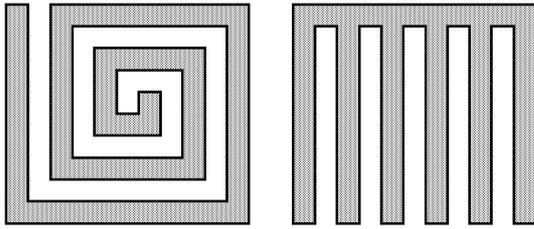


Abb. 2.5: Zwei in ihrer Struktur und Oberfläche aber nicht in ihren globalen Fehlern sehr unterschiedliche Netze

Die nebenstehende Abbildung soll im Zweidimensionalen zeigen, dass die globalen Fehler und hier offensichtlich insbesondere der maximale keinen Aufschluss über die Gleichartigkeit von Netzen zulassen, sondern reine Abstände messen. Dies gilt für die vereinfachten Maße wie auch für die genauen Metriken. Werden die beiden in ihrer Oberfläche und Struktur sehr unterschiedlichen Modelle der Abbildung genau übereinandergelegt, so haben sie nur einen globalen maximalen Fehler einer halben Breite der grauen Streifen. Aber auch die globalen durchschnittlichen

Fehler werden nicht den sehr unterschiedlichen Netzen gerecht. Besitzen die Netze nur Knoten an den Modellecken, so sind die globalen vereinfachten Maße sogar Null. Für die Simplifizierungsverfahren spielen solche Extrembeispiele aber keine Rolle, da kein Vereinfachungsoperator bei konkreten Anwendungen solch große Netzänderungen herbeiführt.

Lokale Fehlermaße

Die vorgestellten bereits vereinfachten Maße können noch weiter vereinfacht werden, wenn wie bei den Simplifizierungsverfahren die miteinander zu vergleichenden Netze Entsprechungen auf den jeweiligen Oberflächen haben. Einige auf diesen Maßen beruhende Vereinfachungsverfahren [12, 13, 14, 15] beschleunigen die Berechnung, indem der Abstand der Knoten vom Netz nicht über das gesamte Netz berechnet wird, sondern nur die durch die Vereinfachung korrespondierenden Flächen zur Abstandsberechnung herangezogen werden. Dies hat zwar mögliche kleine Fehler zur Folge, vergleicht aber genau die Regionen, die sich in den Netzen entsprechen sollen. Diese Klasse von Fehlermaßen werden *lokale Fehlermaße* genannt.

Diese weitere Vereinfachung der Fehlermaße besitzt immer einen größeren Fehler als E_{max} , E_{avg} bzw. E_{sum} , da durch die Einschränkung von $d_v(M)$ zur minimalen Abstandsuche nur ein Teil des Netzes betrachtet wird. Wird also konkret die Vereinfachung durch Verschieben von Knoten herbeigeführt, zum Beispiel wie bei der in dieser Arbeit im Mittelpunkt stehenden Knotenverschmelzung, wo zwei Knoten auf die gleiche Position verschoben werden und dabei verschmelzen, so sind die lokalen Fehler, die den Abstand der verschobenen Knoten zu den Dreiecken messen, an denen die ursprünglichen Knoten angelegen haben, obere Schranken für die entsprechenden globalen Fehler. Um in fortlaufenden Vereinfachungsstufen weiterhin eine obere Schranke für den global vereinfachten Fehler darzustellen, müssen die lokalen Fehler für eine Knotenverschiebung immer den Vergleich mit dem Originalnetz suchen und sich für eine lokale Abstandsberechnung die zu einem Knoten korrespondierenden Dreiecke merken.

Ebenenabstand als Fehlermaß

Die lokale Abstandbetrachtung ist vordringlich zur Geschwindigkeitssteigerung eingeführt worden. Möchte man die Geschwindigkeit noch weiter erhöhen, so könnte die aufwendige Abstandsberechnung von Knoten zu Dreiecken, die im Anhang A.3.1 genauer beschrieben wird, durch eine Entfernungsbestimmung von Knoten zu Ebenen ersetzt werden, wobei die Ebenen jeweils einem Dreieck zugeordnet sind und durch dessen Knoten gehen.

Abbildung 2.6 zeigt im Zweidimensionalen den großen Unterschied in der Fehlerbestimmung, der durch die Betrachtung der Geraden bzw. Ebene anstatt des Geradenstücks bzw. des Dreiecks zustande kommen kann. Alle Punkte auf der gepunkteten Linie sind gleich weit von der gestrichelten Geraden entfernt und besitzen somit den gleichen Fehler. Für den nicht senkrecht über dem Geradenstück liegenden Punkt würde also statt des eigentlichen Fehlers, der durch das helle Geradenstück angezeigt wird, das dunkle Geradenstück zur Fehlerberechnung herangezogen werden. Werden aber bei der lokalen Fehlerbestimmung die korrespondierenden Dreiecke zum Aufspannen der Vergleichsebenen herangezogen und ändert sich zum Beispiel wie bei der Knotenverschmelzung die Position der vereinfachten Knoten nur geringfügig, so hält sich der Fehler der Fehlerbestimmung in Grenzen.

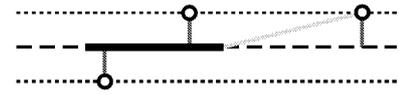


Abb. 2.6: Der helle Punktastand zum dunklen Geradenstück wird bei der Abstandsberechnung über eine Gerade durch den wesentlich kleineren Abstand senkrecht zur Geraden ersetzt

Für einen Knoten v lautende *lokale Fehlermaße auf Basis von Ebenenabständen*, die sich an die oben beschriebenen globalen Maße anlehnen, sind folgende:

$$E_{max}^l(v, E) = \max_{i \in \mathbb{N}_k} D_i(v),$$

$$E_{avg}^l(v, E) = \sum_{i=1}^k D_i^2(v),$$

$$E_{sum}^l(v, E) = \sum_{i=1}^k D_i(v).$$

Dabei ist E die Menge der zum Knoten v zugeordneten k Ebenen E_i . Sind die Ebenen gegeben durch die Hessesche Normalform $E_i : n_i^t w + d_i = 0$, $w \in \mathbb{R}^3$, $i \in \mathbb{N}_k$ mit n_i als normierte Normalenvektoren und $|d_i|$ als Abstände der Ebenen zum Nullpunkt, dann lautet der Ebenenabstand von v zu einer Ebene E_i

$$D_i(v) = |n_i^t v + d_i|.$$

Auf eine Durchschnittsberechnung bei der Summe der quadratischen Abstände und der einfachen Summe der Abstände ist verzichtet worden, da das Ziel dieser Fehlermaße kein Vergleich zweier kompletter Netze ist, sondern eine einzelne Knotenposition zu den korrespondierenden Ebenen bewerten soll. Daher ist es sinnvoll, die Anzahl der Ebenen negativ in den Fehler einfließen zu lassen, da ein Knoten auch alle seine korrespondierenden Ebenen vertreten soll.

Ebenenabstand zu Ursprungsebenen als Semi-Norm

Die Ebenenabstände sind für Ursprungsebenen *Semi-Normen* bzw. *Halb-Normen*. *Ursprungsebenen* sind Ebenen, die durch den Ursprung verlaufen. Semi-Normen unterscheiden sich von Normen nur in ihrer Semidefinitheit, die die Definitheit der Normen ablöst. Es wird also die erste Bedingung der Definition der Norm ersetzt durch:

$$1. \|x\| \geq 0 \quad \forall x \in V \quad (\text{Semidefinitheit}).$$

Ebenenabstand ist Semi-Norm, Beweis:

Für alle $v, w \in \mathbb{R}^3$, $\alpha \in \mathbb{R}$ gilt:

1. $D(v) \geq 0$, da Betrag immer nicht negativ ist,
2. $D(\alpha v) = |n_i^t(\alpha v)| = |\alpha(n_i^t v)| = |\alpha||n_i^t v| = |\alpha|D(v)$,
3. $D(v + w) = |n_i^t(v + w)| = |n_i^t v + n_i^t w| \leq |n_i^t v| + |n_i^t w| = D(v) + D(w)$. ■

Da $D(v) = 0 \forall v \perp n$, handelt es sich bei dem einfachen Ebenenabstand nicht um eine Norm.

Da die Summe oder das Maximum von Semi-Normen wiederum eine Semi-Norm ist, sind E_{max}^l und E_{sum}^l Semi-Normen. Auch $\sqrt{E_{avg}^l}$ ist eine Semi-Norm. Dass die Wurzel der Summe der Quadratabstände eine Semi-Norm ist, soll im Folgenden bewiesen werden. Der Beweis kann auch auf die anderen Semi-Normen entsprechend angewandt werden.

$\sqrt{E_{avg}^l}$ ist Semi-Norm, Beweis:

Für alle $v, w \in \mathbb{R}^3$, $\alpha \in \mathbb{R}$ gilt:

1. $\sqrt{E_{avg}^l(v, E)} \geq 0$, da $D_i(v)$ nicht negativ ist, sind auch Quadrate, Summen und Wurzeln von $D_i(v)$ nicht negativ,

2.

$$\begin{aligned} \sqrt{E_{avg}^l(\alpha v, E)} &= \sqrt{\sum_{i=1}^k D_i^2(\alpha v)} = \sqrt{\sum_{i=1}^k (|\alpha|D_i(v))^2} \quad (\text{da } D_i(v) \text{ Semi-Norm}) \\ &= \sqrt{\sum_{i=1}^k \alpha^2 D_i^2(v)} = \sqrt{\alpha^2 \sum_{i=1}^k D_i^2(v)} \\ &= |\alpha| \sqrt{\sum_{i=1}^k D_i^2(v)} = |\alpha| \sqrt{E_{avg}^l(v, E)}, \end{aligned}$$

3.

$$\begin{aligned} E_{avg}^l(v + w, E) &= \sum_{i=1}^k D_i^2(v + w) \leq \sum_{i=1}^k (D_i(v) + D_i(w))^2 \quad (\text{da } D_i(v) \text{ Semi-Norm}) \\ &= \sum_{i=1}^k (D_i^2(v) + 2D_i(v)D_i(w) + D_i^2(w)) \\ &= \sum_{i=1}^k D_i^2(v) + 2 \sum_{i=1}^k D_i(v)D_i(w) + \sum_{i=1}^k D_i^2(w) \\ &\leq \sum_{i=1}^k D_i^2(v) + 2 \sqrt{\sum_{i=1}^k D_i^2(v) \sum_{i=1}^k D_i^2(w)} + \sum_{i=1}^k D_i^2(w) \quad (\text{CSU}^1) \\ &= E_{avg}^l(v, E) + 2\sqrt{E_{avg}^l(v, E)E_{avg}^l(w, E)} + E_{avg}^l(w, E) \\ &= \left(\sqrt{E_{avg}^l(v, E)} + \sqrt{E_{avg}^l(w, E)} \right)^2 \\ &\Leftrightarrow \sqrt{E_{avg}^l(v + w, E)} \leq \sqrt{E_{avg}^l(v, E)} + \sqrt{E_{avg}^l(w, E)}. \quad \blacksquare \end{aligned}$$

¹Cauchy-Schwarzsche Ungleichung, CSU: $|a_1 b_1 + a_2 b_2| \leq \sqrt{a_1^2 + a_2^2} \sqrt{b_1^2 + b_2^2}$

E_{avg}^l ist keine Semi-Norm, da die Homogenität verletzt ist. Wie schon bei den globalen Maßen ist beim Quadratabstand die in der entsprechenden Euklidischen Metrik vorkommende Wurzel weggelassen worden, da schon durch den Verzicht auf den Durchschnitt das Fehlermaß keine anschauliche Bedeutung mehr hat und es bei der Bewertung von Knoten über das Fehlermaß ausschließlich auf eine relative Bewertung ankommt. Die relative Ordnung des Maßes wird durch das Weglassen der Wurzel nicht gestört, da die Wurzel streng monoton wachsend ist.

Ein auf E_{max}^l aufsetzendes Verfahren ist von Ronfard und Rossignac in [16] vorgestellt worden. Die Summe der Quadratabstände E_{avg}^l ist Basis des in dieser Arbeit im Mittelpunkt stehenden *Quadrik-Fehlermaßes* und wird ausführlich im nächsten Kapitel behandelt.

Ebenenabstand als Norm

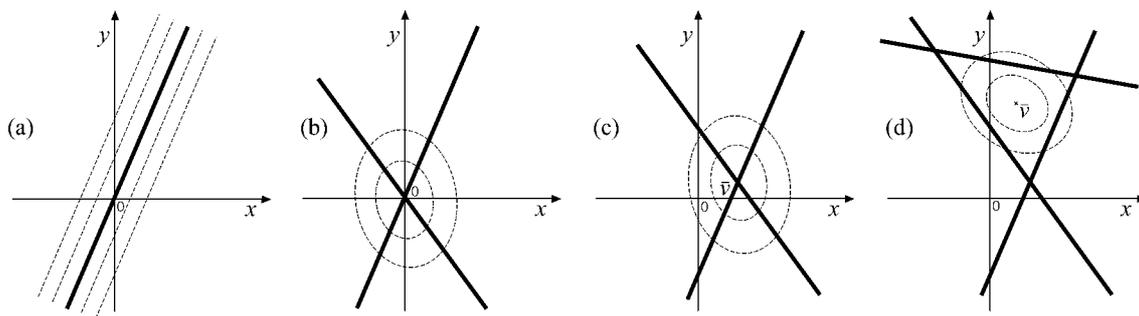


Abb. 2.7: (a) Der Abstand zu einer Ursprungsgeraden erzeugt eine Semi-Norm, (b) zu nicht parallelen Ursprungsgeraden eine Norm und zu Geraden, von denen mindestens eine nicht durch den Ursprung geht, ein Fehlermaß, das (c) minimal Null oder (d) echt positiv ist

Abbildung 2.7 zeigt in (a) beispielhaft im Zweidimensionalen, warum der generelle Ebenenabstand ohne weitere Bedingungen nur eine Semi-Norm ist. Die dicken durchgezogenen Linien sind Geraden und stehen im Dreidimensionalen für Ebenen. Die dünnen gestrichelten Linien sind *Isolinien* im Zweidimensionalen und *Isflächen* im Dreidimensionalen. Sie bilden die Punktmenge mit dem gleichen lokalen Fehler. Für Isolinie wird auch der Begriff *Niveaulinie* verwendet. Die Isolinien in (a) sind parallel zur Geraden, bilden für denselben Fehler immer ein Linienpaar und nähern sich der Geraden bei sinkendem Fehler. Für den Fehler Null ist die Gerade selbst Isolinie, da $D(v) = 0 \forall v \perp n$. Somit ist das Fehlermaß bei einer Geraden bzw. Ebene nur eine Semi-Norm.

In (b) wird aber aus der Semi-Norm eine Norm im Zweidimensionalen, da die Geraden genau nur einen gemeinsamen Schnittpunkt haben und genau nur dort identisch Null sind:

$$E_{\star}^l(v, E) = 0 \Leftrightarrow v = 0.$$

E_{\star}^l steht dabei für alle eingeführten Semi-Normen. Dies ist auch gut an den Isolinien von E_{avg}^l zu sehen, die hier stellvertretend in (b), (c) und (d) eingezeichnet wurden. Sie ziehen sich für kleiner werdende Fehler immer weiter zusammen, bis sie in (b) für den Fehler Null auf den Nullpunkt zusammenfallen. Die Definitheit ist im Dreidimensionalen erst durch drei sich nur im Nullpunkt schneidende Ebenen gegeben, da mit weniger als drei Ebenen oder mehr Ebenen, die sich nicht nur im Nullpunkt schneiden, mehr als nur der Nullpunkt die Gleichung $E_{\star}^l(v, E) = 0$ erfüllt.

Bisher wurde sich auf Ursprungsebenen beschränkt. In der Praxis sollen aber beliebige Ebenen im Fehlermaß berücksichtigt werden können. In Abbildung 2.7 (c) und (d) sind solch allgemeine Fälle

gezeigt. (c) beschränkt sich dabei auf den Fall genau eines gemeinsamen Schnittpunktes, und (d) steht für eine Konstellation ohne einen gemeinsamen Schnittpunkt, aber mit genau einem Punkt, an dem der minimale Fehler angenommen wird. Bei Ebenen sind für den Fall (c) drei und für (d) vier Ebenen nötig, um die Bedingungen zu erfüllen. Abstände zu allgemeinen Ebenenkonstellationen sind keine Normen mehr, da zum Beispiel die Homogenität, die besagt, dass der Abstand linear wachsen muss, schon allein für eine Ebene, die nicht durch den Nullpunkt geht ($d \neq 0$), verletzt ist:

$$D(\alpha v) = |n^t(\alpha v) + d| = |\alpha(n^t v) + d|$$

$$\stackrel{\text{i.a.}}{\neq} |\alpha|(n^t v) + |\alpha|d = |\alpha||n^t v + d| = |\alpha|D(v).$$

Der Fall (c) lässt sich aber leicht durch Verschieben des Koordinatensystems in eine Norm überführen. Sei \bar{v} der einzige Schnittpunkt der Ebenen, so lauten die ursprünglichen Knoten $v \in \mathbb{R}^3$ nach Verschiebung im neuen Koordinatensystem:

$$\underline{v} = v + \bar{v} \Leftrightarrow v = \underline{v} - \bar{v}.$$

Die lokalen Fehlermaße sind im neuen Koordinatensystem wieder Normen, da der Ebenenabstand

$$D^{\bar{v}}(\underline{v}) = |n^t \underline{v} + d| = |n^t(v + \bar{v}) + d| = |n^t v + n^t \bar{v} + d|$$

$$\stackrel{(*)}{=} |n^t v - d + d| = |n^t v|$$

im neuen System wieder dem Abstand von Ursprungsebenen entspricht. Die Identität (*) gilt, da n normiert ist und damit das Skalarprodukt mit allen Ebenenpunkten $v \in E$ den gerichteten Abstand $-d$ zum Nullpunkt ergibt ($n^t v = -d$).

Im Falle (d) ist die Transformation zu einer Norm nicht mehr möglich, da der Punkt \bar{v} , an dem das Fehlermaß das Minimum annimmt, je nach Fehlermaß variiert und das echt positive Minimum ($E_{\star}^l(\bar{v}, E) > 0$) für die Homogenität auf Null abgebildet werden muss. Die verschiedenen \bar{v} der Fehlermaße sind gut in der Abbildung 2.8 zu erkennen. Für E_{avg}^l wird aber in Abschnitt 3.2.4 eine Rechenvorschrift vorgestellt, die das Maß in eine Norm überführt.

Vergleich der Fehlermaße

Zum Ende der Einführung über geometrische Fehlermaße sollen zum einen die lokalen Maße miteinander und zum anderen das in dieser Arbeit im Mittelpunkt stehende Quadrik-Fehlermaß E_{avg}^l mit den globalen Metriken E_{max} und E_{avg} im Zweidimensionalen verglichen werden.

Abbildung 2.8 zeigt das Verhalten der lokalen Fehlermaße (a) E_{sum}^l , (b) E_{avg}^l und (c) E_{max}^l über ihre Isolinien, an denen der Fehler identisch ist. Die Fehlerwerte der einzelnen Isolinien wurden äquidistant gewählt, so dass für E_{sum}^l und E_{max}^l die Isolinien gleich weit voneinander entfernt sind. Da der Abstand in E_{avg}^l quadratisch einfließt, wird die Entfernung der Isolinien zueinander immer geringer, um so weiter man sich vom Ellipsenmittelpunkt entfernt. E_{avg}^l bestraft also sinnvollerweise das Entfernen vom Minimum am stärksten.

Es sind jeweils drei Geraden g_1, g_2, g_3 gegeben, die nicht alle durch den Nullpunkt gehen und sich auch nicht in einem gemeinsamen Punkt schneiden. Daher gilt für alle lokalen Fehlermaße

$$E_{\star}^l(v, G) > 0 \quad \forall v \in \mathbb{R}^2,$$

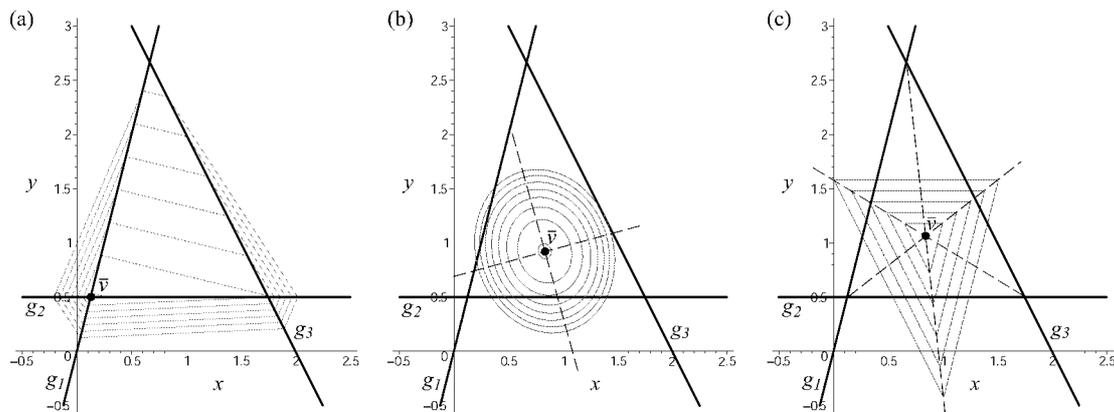


Abb. 2.8: Isolinien der lokalen Fehlermaße (a) E_{sum}^l , (b) E_{avg}^l und (c) E_{max}^l für den Abstand von drei Geraden

also insbesondere auch $E_{*}^l(\bar{v}, G) > 0$. $G = \{g_1, g_2, g_3\}$ ist dabei die Menge der Geraden.

Die Form der Isolinien sagt etwas über die Komplexität des Fehlermaßes aus. Für (a) ergeben sich schon für drei Geraden nicht mehr geschlossen darzustellende Isolinien, deren Formen von einem Viereck bis zu einem Sechseck variieren. Die Komplexität der Form steigert sich bei Hinzunahme weiterer Geraden. Das Gleiche gilt auch für (c), wo für drei Geraden noch Dreiecke die Form der Isolinien in der Nähe des Minimums angeben, bei Hinzunahme von Geraden aber auch Vielecke entstehen. Zudem werden durch den Betrag des Abstandes die Ecken der Dreiecke ab einer gewissen Entfernung vom Minimum abgeschnitten, so dass am Ende auch hier ein Sechseck als Isolinie entsteht. Nur für (b) wird die Form auch für beliebig viele Geraden eine Ellipse bleiben, was in Abschnitt 3.1.3 gezeigt wird, und damit eine ebenfalls im gleichen Abschnitt beschriebene kompakte Darstellung des Fehlermaßes erlaubt. Für den Spezialfall zweier nicht paralleler Geraden ist im Anhang C.2 eine Konstruktion einer Ellipse als Isolinie der Summe der Quadratabstände durchgeführt worden. Die Begründung für die Komplexität des lokalen Summen-Fehlers und Maximum-Fehlers liegt in der nicht möglichen Zusammenfassung von Betragssummen und dem Maximum einer Aufzählung.

Für (b) sind die Hauptachsen der Ellipsen als gestrichelte Linien eingezeichnet, für (c) sind die gestrichelten Linien die Winkelhalbierenden des durch die Geraden eingeschlossenen Dreiecks. Die Winkelhalbierenden sind die Geraden, die von den jeweiligen Schenkeln maximal weit weg sind. Schneiden sich alle Winkelhalbierenden in einem Punkt, so ist dieser Punkt genau die Stelle mit dem maximalen Fehler für alle Geraden. In einem Dreieck schneiden sich wie in diesem Beispiel die Winkelhalbierenden immer in einem Punkt, an dem E_{max}^l somit sein Minimum annimmt. Für ein beliebiges Vieleck schneiden sich die Winkelhalbierenden für gewöhnlich nicht mehr in einem Punkt. Der lokale Summen-Fehler besitzt sein Minimum in diesem Beispiel an dem Geradenschnittpunkt, der an der am Schnittpunkt unbeteiligten Geraden am nächsten dran liegt. Das Minimum von E_{avg}^l liegt im Zentrum der Ellipsen, also im Schnittpunkt ihrer Hauptachsen.

Die Geraden im Beispiel sind möglichst beliebig gewählt worden. Würde das durch die Geraden eingeschlossene Dreieck gleichseitig sein, so würden die Minima aller Maße auf demselben Punkt angenommen werden und damit auf dem Schnittpunkt der Winkelhalbierenden liegen. Wäre das Dreieck nur gleichschenkelig, würden alle Minima der Maße auf der Winkelhalbierenden liegen, deren Schenkel gleich lang sind.

Wären zwei der abgebildeten Geraden parallel, so würde für (b) das Minimum auf der dritten nicht parallelen Geraden in der Mitte der parallelen Geraden angenommen werden. An der gleichen Stelle sind auch (a) und (c) minimal. (c) besitzt aber auch auf einem Geradenstück durch diesen Punkt parallel zu den parallelen Geraden das gleiche Minimum und ist damit nicht mehr eindeutig. Ebenso besitzt (b) mehrere Lösungen, da auf dem Geradenstück auf der nicht parallelen Geraden zwischen den parallelen Geraden die absolute Abstandssumme identisch ist.

Konkrete Werte der Minima für das Beispiel aus Abbildung 2.8 sind in Anhang C.1 berechnet worden.

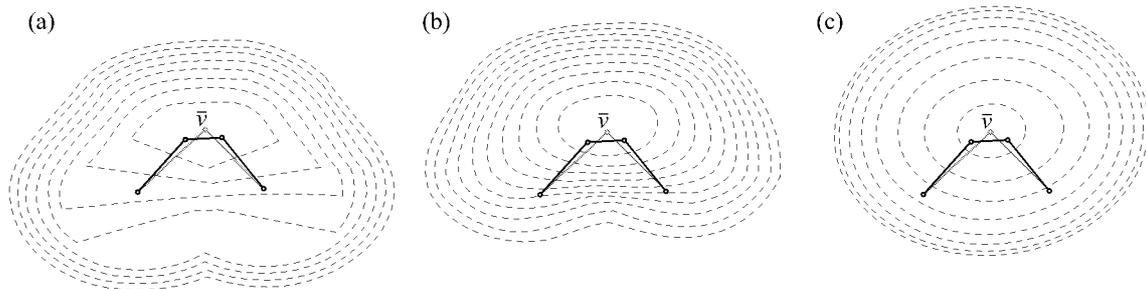


Abb. 2.9: Isolinien der globalen Fehlermetriken (a) E_{max} und (b) E_{avg} und des lokalen Fehlermaßes (c) E_{avg}^l für den Fehler eines Vereinfachungsschrittes

Um ein Gefühl für die globalen Metriken im Vergleich zu dem im implementierten Algorithmus im Mittelpunkt stehenden Quadrik-Fehlermaß E_{avg}^l zu bekommen, soll hier das Verhalten dieser Maße über ihre Isolinien an einem Beispiel skizziert werden. In Abbildung 2.9 ist zudem angedeutet, was die Fehlermaße im Bereich der Vereinfachung aussagen sollen. In diesem Beispiel soll ein Geradenstück mit seinen beiden Endpunkten auf einen Punkt reduziert, also das schwarze Netz auf das helle Netz vereinfacht werden. Das dunklere Netz mit den drei Geradensegmenten wird also auf das blässere Netz mit den zwei Geradenstücken vereinfacht, indem das obere Segment auf einen Knoten zusammenfällt.

Für einen Vergleich ist die Konzentration auf einen Netzabschnitt nötig, bei dem ein Vereinfachungsschritt bewertet werden soll, weil sonst ein Vergleich der globalen Fehler für das ganze Netz mit dem Quadrik-Fehlermaß nicht sinnvoll wäre, da dieses nur eine lokale Bewertung eines Punktes zum ursprünglichen Netz vornimmt.

Das Quadrik-Fehlermaß E_{avg}^l in (c) liefert die typischen Ellipsen, in deren Zentrum der optimale Verschmelzungspunkt \bar{v} dieses Verfahrens liegt (siehe Abschnitt 3.1.4). Als optimaler Verschmelzungspunkt wird der Punkt bezeichnet, der den geringsten Fehler in dem verwendeten Maß verursacht. Die Konturlinien von E_{avg} in (b) sind gegenüber denen von E_{max} in (a) glatter. Der optimale Verschmelzungspunkt der beiden globalen Verfahren liegt unwesentlich über dem von E_{avg}^l . Die Linien aus (a) und (b) sind komplizierter, was auch der Komplexität der Berechnung entspricht, haben aber Ähnlichkeit mit dem Quadrik-Fehlermaß, insbesondere die Linien des inneren Bereiches von E_{avg} . Diese Ähnlichkeit wird aber mit steigender Netzkomplexität abnehmen und sich immer weiter auf den Verschmelzungspunkt zurückziehen.

Die Einführung zu geometrischen Fehlermaßen ist hiermit abgeschlossen. Eine ausführliche Behandlung des in der Implementation im Mittelpunkt stehenden Quadrik-Fehlermaßes E_{avg}^l findet im Kapitel 3 statt.

2.2 Verfahrenseinteilung

Am Anfang der Generierung von vereinfachten Netzen stand die Erzeugung von *Detailstufen* (level of detail, LOD) von Hand. Ein Designer entwarf seine Charaktere bzw. Objekte in einigen wenigen Auflösungen. Dieser Vorgang war schwer, sehr zeitaufwendig und beschränkte sich auf nur eine geringe Anzahl an Stufen, die dadurch keinen visuell fließenden Übergang zwischen zwei Detailstufen ermöglichten. Diese Arbeit wurde von Modellierungssoftware immer besser unterstützt bis hin zur automatischen Simplifizierung, die durch im Folgenden vorgestellte Verfahren möglich wird.

Die Einteilung der Verfahren in Kategorien für folgende Abschnitte geschieht anhand der *elementaren Operation*, die auf das Netz ausgeführt wird, um es zu vereinfachen. Die Operationen können das Netz rein geometrisch und/oder lokal topologisch verändern. *Topologische Operationen* verändern nicht die Geometrie V des Netzes $M = (K, V)$ und damit die Position der Netzknoten, sondern schaffen durch Topologieänderungen eine neue Netzstruktur. Dabei können Knoten ihre Verbindung zum Dreiecksnetz verlieren und das Netz dadurch an Komplexität abnehmen. *Geometrische Operationen* hingegen zeichnen sich durch die Positionsänderung der Netzknoten aus und verändern nicht die Netzkomplexität. Zum Beispiel kann sich bei der später beschriebenen Operation der Kantenkontraktion die Kante auf einen der bereits bestehenden Knoten zurückziehen, dann ist die Operation rein topologisch, oder die Endknoten der Kante verschmelzen in einen neuen Knoten, dann handelt es sich um eine topologische und geometrische Operation.

Das Fehlermaß steht in engem Zusammenhang mit dem Operator eines Verfahrens und könnte auch zur Klassifizierung der Verfahren herangezogen werden. In Abschnitt 2.7.3 und in der abschließenden Tabelle 2.1, die die Arbeiten auch in anderen Bereichen zusammenfassend charakterisiert, werden verschiedene Weisen der Einteilung aufgezeigt. Eine weitere Klassifizierung ließe die Behandlung der Netztopologie zu, auf die in Abschnitt 2.7.1 näher eingegangen wird.

Ganz andere Einteilungen werden von Luebke in [17] gemacht. Zum einen schlägt er eine Aufteilung durch die im nächsten Absatz beschriebenen algorithmischen Verfahrensweisen und zum anderen durch die ebenfalls weiter unten beschriebene Richtung der Netzoperationen vor.

Beispiele von *Verfahrensweisen* sind Greedy-, Lazy-, Schätz- oder nicht-optimierte Algorithmen. *Greedy-Algorithmen* arbeiten zur Findung einer möglichst optimalen Lösung eine Menge sukzessive so ab, dass das bestbewertete Element zu dem aktuellen Zeitpunkt zuerst verarbeitet wird [18]. Eine Neubewertung der veränderten Menge ist zumindest lokal nötig. Bei *Lazy-Algorithmen* hingegen wird statt der Neuberechnung nur ein Zeichen (flag) gesetzt, dass bestimmte Werte ungenau sind. Erst bei der Benutzung eines ungenauen Elementes wird der exakte Fehler berechnet. Dies ist zwar meist ungenauer, aber beschleunigt das Verfahren. *Schätz-Algorithmen* arbeiten statt einer andauernden Neuberechnung mit Schätzwerten. Auch dies führt zu Ungenauigkeiten, aber auch zu Geschwindigkeitsgewinnen. Ganz ohne Bewertung kommen die *nicht-optimierten Algorithmen* aus, wie es zum Beispiel das Verfahren von Rossignac und Borrel aus Abschnitt 2.4.1 eines ist. Die meisten der vorgestellten Verfahren und auch das zur Implementation ausgewählte sind Greedy-Algorithmen.

Die *Richtung der Netzoperation* kann zum einen vom Originalnetz eine fortlaufende *Vereinfachung* sein (top-down framework) oder die Operation bewirkt zum anderen eine *Detailsteigerung* ausgehend von einem Basisnetz (bottom-up framework). Außer dem ersten vorgestellten Verfahren gehen alle Methoden den Weg der Vereinfachung.

2.3 Verfeinerung (refinement)

Diese Gruppe von Verfahren gehört zu der kleinen Klasse von Methoden, die nicht ein Ausgangsnetz vereinfachen, sondern ein grobes zu generierendes *Basisnetz* bis zum gewünschten Detailgrad verfeinern (adaptive subdivision) und dabei die Topologie nicht verändern. Da diese Vorgehensweise selten ist, wird für diese Klasse an Verfahren nur eine Arbeit vorgestellt.

2.3.1 Multiresolution Analysis of Arbitrary Meshes

Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery and Werner Stuetzle [19]

Dieses fortschreitende Unterteilungsverfahren benutzt für den rekursiven Aufteilungsprozess eine kompakte Wavelet-Repräsentation^G und erweitert die Arbeit von Lounsbery et al. [20] hauptsächlich um die automatische Erzeugung eines zum Originalnetz topologisch äquivalenten Basisnetzes.

Der größte Teil der Arbeit besteht in der Findung dieses Basisnetzes. Dies ist für ein Höhenfeld einfach ein Rechteck, bedarf für topologisch komplexe Objekte aber auch entsprechend aufwendige Basisnetze. Die Erzeugung geschieht über wachsende Voronoi-Regionen^G, die, wenn sie sich nicht mehr weiter ausdehnen können, im zweiten Schritt der Delaunay-Triangulierung^G dienen. In der Abbildung 2.10 ist in (b) für das Originalnetz (a) die Aufteilung in Voronoi-Regionen zu sehen. Die Delaunay-Triangulation in (c) erzeugt das Basisnetz (d).

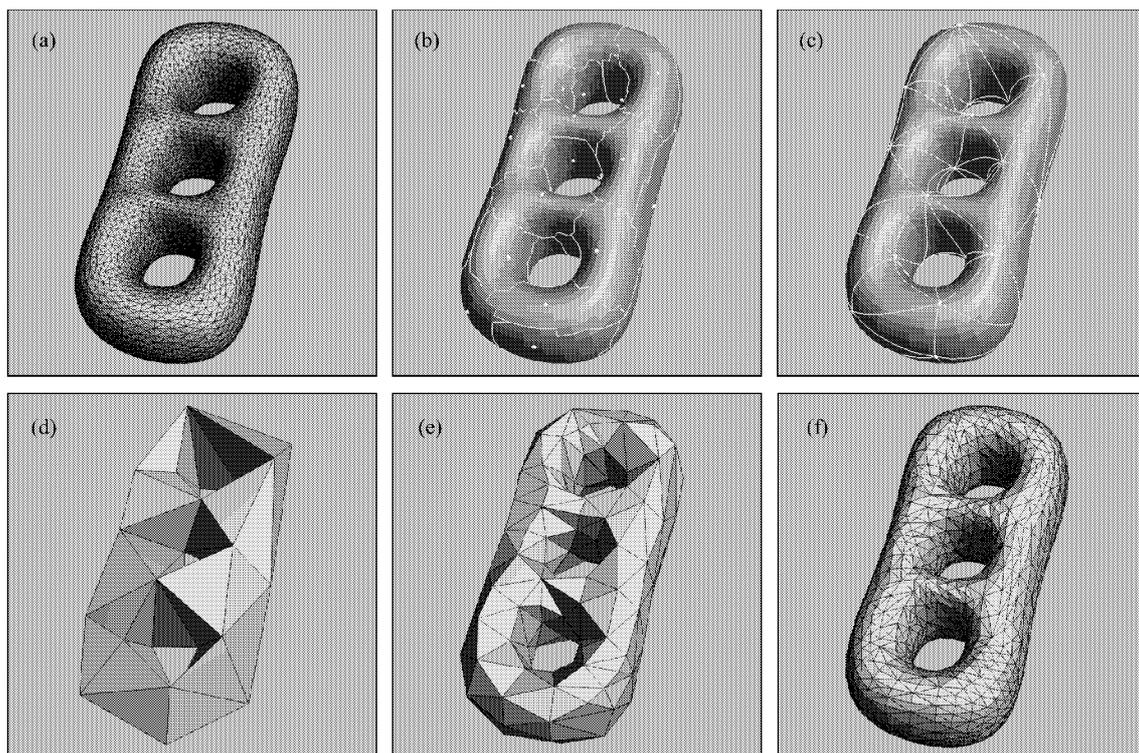


Abb. 2.10: Stufen des Vereinfachungsverfahrens durch Verfeinerung: (a) Originalnetz, (b) Aufteilung in Voronoi-Regionen, (c) Delaunay-Triangulation, (d) Basisnetz, (e) Approximation mit Fehler von 4,5% und (f) von 1%, basiert auf Bildern aus [19]

Nach der Erzeugung des Basisnetzes wird dieses durch rekursive Vierteilung der Dreiecke verfeinert (remeshing) und eine Zuordnung der Knoten des Originalnetzes zur verfeinerten Unterteilung über eine harmonische Abbildung (harmonic maps) hergestellt (subdivision connectivity). Wavelet-Koeffizienten geben dabei die Ausrichtung der verfeinerten Unterteilung zum Originalnetz an (multiresolution analysis, MRA). Der gewünschte globale Approximationsfehler als Abstand zum Originalnetz kann angegeben werden und bewirkt eine angemessene Aufteilung mittels entsprechend vieler Wavelet-Koeffizienten. Die 70 Dreiecke des Basisnetzes sind in Abbildung 2.10 (e) auf 366 angestiegen und die Approximation hat einen Fehler von 4,5% der maximalen Ausdehnung des Originalnetzes, in (f) sind es 2614 Dreiecke, dessen Netz einen Fehler von 1% aufweist.

Die Netzrepräsentation nach der Vereinfachung besteht somit aus einem Basisnetz und einer Folge von Wavelet-Koeffizienten, die entsprechend der gewünschten Approximationsqualität am Ende unbeachtet bleiben.

Diese mathematisch elegante Methode birgt aber viele Nachteile. Die Berechnung des Basisnetzes und der Wavelet-Repräsentation dauert lange. Die Methode kann nur auf Mannigfaltigkeiten angewandt werden. Die Topologieerhaltung bringt den damit verbundenen Nachteil mit sich, dass die Vereinfachung bei komplexen Modellen stark limitiert ist. Das Verfahren arbeitet wie ein Tiefpass-Filter und kann daher Unstetigkeiten scharfer Details wie spitze Ecken und scharfe Kanten schwer erfassen, wenn sie nicht gerade durch das Basisnetz bereits erfasst wurden. Es eignet sich stattdessen gut für weiche Formen. Das Originalnetz kann für gewöhnlich nicht exakt mit der Wavelet-Repräsentation wiedergegeben werden, der Fehler aber beliebig gesenkt werden, auch wenn dadurch mehr Dreiecke als im Original nötig sind.

2.4 Knotenhäufung (vertex clustering)

Die folgenden Methoden dieses Kapitels gehen von einem feinen Netz aus und simplifizieren dieses. Die *Knotenhäufung* besitzt als elementare Operation zur Verringerung der Netzkomplexität das Zusammenziehen von Knoten auf einen gemeinsamen Verschmelzungspunkt. Das in diesem Abschnitt vorgestellte Verfahren, das auf der Knotenhäufung aufsetzt, diente vielen späteren Verfahren der gleichen Klasse als Ausgangsbasis. Eine Änderung wurde zum Beispiel bei der Raumaufteilung empfohlen, die nicht mehr äquidistant sondern adaptiv beispielsweise über einen den Raum rekursiv aufteilenden Octree^G realisiert werden sollte.

2.4.1 Multi-resolution 3D approximations for rendering complex scenes

Jarek Rossignac and Paul Borrel [21]

Diese von den vorgestellten Verfahren älteste Arbeit lässt sich auf jede Art von Dreiecksnetz anwenden, setzt also keine Mannigfaltigkeit voraus, erhält dafür aber auch keine Mannigfaltigkeit oder sonstige topologische Eigenschaften des Netzes.

Zwei Schritte bestimmen die Vorgehensweise. Im ersten Schritt werden alle Knoten bewertet, um die Gewichtung für den zweiten Schritt zu bestimmen. Das Gewicht setzt sich wiederum aus zwei Komponenten ω_1 und ω_2 mit der Einschränkung $0 \leq \omega_1, \omega_2 \leq 1$ zusammen. Die erste bewertet Knoten aufgrund der Größe anliegender Dreiecke - große Dreiecke hohe, kleine Dreiecke geringe Relevanz - die zweite Komponente lässt die Oberflächenkrümmung über den Kehrwert des größten Winkels zwischen zwei an einem Knoten anliegenden Kanten in das Gewicht einfließen.

Als zweiter Schritt wird ein dreidimensionales äquidistantes Gitter über das gesamte Netz gelegt und für jede Zelle bzw. Voxel^G ein Verschmelzungspunkt \bar{v} bestimmt, der sich aus den gewichteten k Knoten v_i , $i \in \mathbb{N}_k$, die innerhalb der jeweiligen Zelle liegen, zusammensetzt:

$$\bar{v} = \frac{1}{\sum_{i=1}^k \omega_{1i} \omega_{2i}} \sum_{i=1}^k \omega_{1i} \omega_{2i} v_i .$$

Die Auflösung des Gitters bestimmt damit die Qualität der resultierenden Simplifizierung.

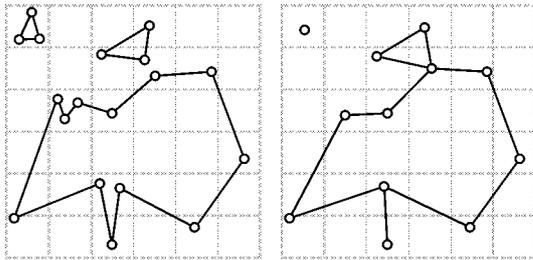


Abb. 2.11: Knoten des linken Originals im Zweidimensionalen werden innerhalb einer Zelle auf einen Knoten entsprechend der rechten Zeichnung zusammengefasst, Bild aus [22]

Die sich möglicherweise verändernde Topologie wird nachgezogen, indem dem Verschmelzungspunkt alle Verbindungen der Knoten seiner Zelle zugewiesen werden.

Bei der Knotenverschmelzung innerhalb der Zelle kann es dazu kommen, dass Dreiecke zu Kanten oder Knoten degenerieren. Im Zweidimensionalen ist dies in Abbildung 2.11 beispielhaft gezeigt. Diese entarteten Dreiecke werden nicht aus dem Netz gelöscht, sondern als Kante bzw. Knoten der Grafikhardware übergeben, wobei zuvor doppelte Primitive herausgefiltert werden. Bei starker Vereinfachung helfen diese Primitive das Objekt besser zu erkennen.

Neben der Degeneration der Dreiecke zu Kanten und Knoten ist in Abbildung 2.11 auch gut zu erkennen, dass feine Strukturen, die kleiner als die Zellausdehnung sind, verschwinden können, getrennte Netze verschmelzen können und der Verschmelzungspunkt nicht in der Zellenmitte liegen muss.

Dieses Verfahren ist sehr robust und verarbeitet beliebig entartete Netze, liefert aber auch entsprechend entartete Resultate, die visuell mit zu den schlechtesten Ergebnissen aller Simplifizierungsverfahren führen. Dafür ist es das schnellste aller Verfahren, die in dieser Arbeit vorgestellt werden. Da kein Knoten weiter als der Durchmesser eines Voxel verschoben wird, ist der Durchmesser eine Schranke für den Hausdorff-Abstand als Approximationsfehler.

2.5 Knotendezimierung (vertex decimation)

Das zuvor vorgestellte Verfahren hat für das gesamte Netz gleichmäßig eine Operation durchgeführt, die in ihren Ausprägungen untereinander nicht bewertet wurden und damit keine Reihenfolge zu beachten war. Die weiteren in diesem Kapitel vorgestellten Methoden arbeiten mit *lokalen Operationen* auf dem Netz, deren Ausprägungen durch eine Bewertung zueinander in eine Abarbeitungsreihenfolge gebracht werden. Es könnte zwar auch bei den lokalen Operationen rein zufällig vorgegangen oder nur auf eine gleichmäßige Verteilung auf dem Netz geachtet werden, was den Vorteil einer einfachen Implementation und einer schnellen Vereinfachung hätte. Diese Vorgehensweise hätte aber ein schlechtes Vereinfachungsergebnis zur Folge. Stattdessen wird meist der geometrische Abstand in irgendeiner Form zur Schaffung einer Abarbeitungsreihenfolge zur Hilfe genommen.

Bei der *Dezimierung* zur Netzverfeinerung wird iterativ ein Knoten entfernt und das entstehende Loch mittels Triangulierung geschlossen.

In dieser Klasse von Verfahren werden zwei vorgestellt, von denen das erste die fortschreitende Dezimierung der Knoten im Mittelpunkt stehen hat und diese inkrementell bewertet. Die zweite Methode hingegen nutzt die Dezimierung nur zur abschließenden Knotenverringern, die Knotenauswahl aber geschieht über zuvor geschaffene globale Grenzen.

2.5.1 Decimation of triangle meshes

Schroeder et al. [23]

Wie zuvor erwähnt, wird bei dieser Art der Netzverfeinerung als Netzoperation ein Knoten entfernt. Dabei wird der Knoten gelöscht, der bei seiner Löschung einen möglichst geringen Fehler verursacht. Das dadurch entstandene Loch der Form eines geschlossenen Polygons^G wird in dieser Arbeit über eine zum Ende dieses Abschnitts vorgestellte Methode geschlossen, indem optisch vorteilhafte Dreiecke in das Netz eingefügt werden.

Zur Fehlerbestimmung wird eine Durchschnittsebene (average plane) entlang der Punkte gelegt, die mit dem zu löschenden durch eine Kante verbunden sind. Der Abstand zur Ebene spiegelt dabei den Fehler wieder, den der zu entfernende Punkt verursachen würde. Bei Randpunkten (boundary vertices) wird statt der Ebene eine Gerade durch die benachbarten Randpunkte zur Abstandsbestimmung gelegt. Um Nicht-Mannigfaltigkeiten und Eigenschaften wie Materialien und Normalen eines Netzes zu erhalten oder zumindest besonders zu berücksichtigen, wird zwischen verschiedenen Knotenarten unterschieden. Kritische Knoten einer Nicht-Mannigfaltigkeit werden daher nicht gelöscht, um diese wichtige topologische Eigenschaft zu erhalten. Bei Materialübergängen oder Knoten, bei denen umliegende Dreiecke sich stark in ihren Normalenrichtungen unterscheiden, wird statt dem Ebenen-Knoten Abstand der Abstand des Knotens von dieser inneren Kante entsprechend den Knoten im Randbereich als Fehler gewertet. Spitze Knoten, also Knoten an denen mehr als zwei innere Kanten aufeinanderstoßen, werden meist von der Vereinfachung ausgenommen. Ist im Datensatz hingegen mit Rauschen zu rechnen, so kann die Sonderbehandlung der inneren Kanten auch entfallen. Ist nun der Knoten mit dem geringsten Fehler gefunden worden, so wird dieser gelöscht, das entstandene Loch trianguliert und der Fehler angrenzender Knoten neu bestimmt.

Bei der Triangulierung sollte auf jeden Fall eine vorkommende innere Kante als Kante berücksichtigt werden. Um eine gute Triangulierung zu erhalten, sollten die erzeugten Dreiecke ein ausgeglichenes Seitenverhältnis (aspect ratio) haben und sich nicht überschneiden.

Hierzu wird ein Verfahren vorgeschlagen, bei welchem dem Knotenring um einen zu löschenden Knoten eine Kante eingefügt und damit geteilt wird und mit den entstandenen zwei Teilen ebenfalls so verfahren wird (divide and conquer). Die Kante sollte dabei entlang einer Schnittgeraden (split line) dort eingefügt werden, wo die Knoten einen geringen Abstand zur Durchschnittsebene haben und die Kante kurz ist. Zudem muss zur Vermeidung von Überschneidungen die Knotenlage in den beiden Teilgebieten untersucht werden. Dazu wird eine Schnittsebene (split plane) senkrecht auf der Durchschnittsebene entlang der Schnittgeraden gelegt und überprüft, ob

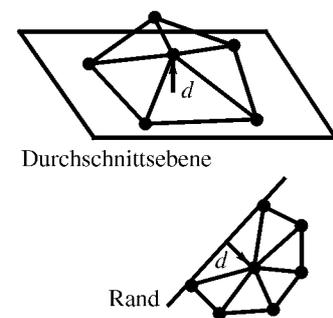


Abb. 2.12: Abstand d zur Durchschnittsebene oder zum Rand als Fehlermaß, Bild aus [23]

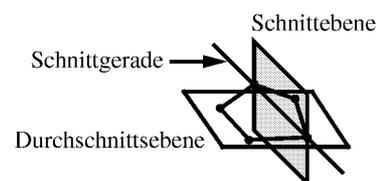


Abb. 2.13: Schnittsebene senkrecht zur Durchschnittsebene entlang der Schnittgeraden zur Untersuchung der Knotenlage, Bild aus [23]

die Knoten der beiden Teilgebiete jeweils auf der anderen Seite dieser Ebene liegen. Ist dies nicht der Fall, kann eine andere Kante als Kandidat untersucht werden oder der Knoten zur Löschung zurückgestellt werden.

2.5.2 Simplification Envelopes

Cohen et al. [24]

Die Arbeit von Cohen et al. ist weniger ein Verfahren als vielmehr eine Ideenskizze, der zwei Beispielumsetzungen beigelegt sind.

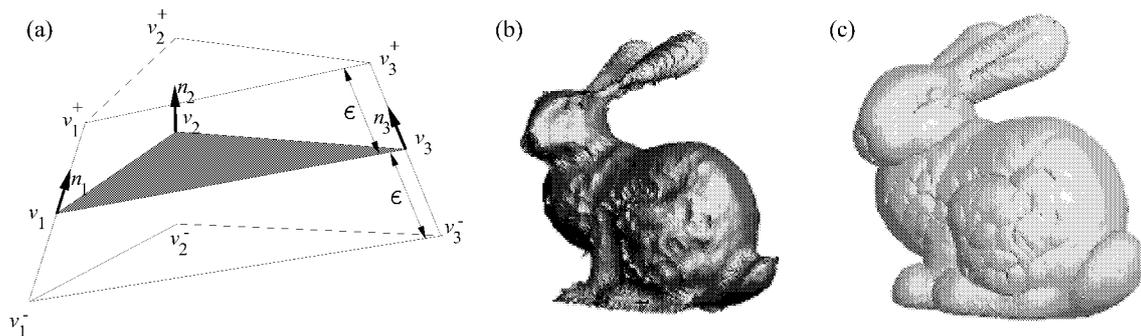


Abb. 2.14: (a) Verschiebung der Netzknoten in Knotennormalenrichtung um ϵ bzw. $-\epsilon$ zur Erstellung der umschließenden Hüllen, (b) innere und (c) äußere Hülle des "Stanford Bunnys"; basiert auf Bildern aus [24]

Die Idee besteht darin, das Netz mit zwei umschließenden Hüllen zu versehen, aus denen das Netz während der Vereinfachung nicht austreten darf. Die einhüllenden Netze werden möglichst in einer Distanz von ϵ vom Originalnetz aufgespannt. Die Erzeugung der inneren und äußeren Hülle geschieht über Kopien des Originalnetzes, dessen Knoten in Knotennormalenrichtung um ϵ bzw. $-\epsilon$ entsprechend Abbildung 2.14 (a) verschoben werden. In Randbereichen wird die Einhüllung mittels Röhren vorgenommen. Ein Beispiel für die innere und äußere Hülle für das "Stanford Bunny" ist in Abbildung 2.14 (b) bzw. (c) zu sehen.

Da es den Hüllen nicht erlaubt ist, sich selbst zu durchdringen, wird an diesen kritischen Stellen das ϵ so weit verringert, bis die Überschneidung unterbunden ist. Um eine Durchdringung der Hüllen zu verhindern, muss entsprechend Abbildung 2.15 darauf geachtet werden, dass Knoten der Hülle sich nicht aus den Voronoi-Regionen^G der adjazenten Dreiecke des ursprünglichen Knotens bewegen. In dem skizzierten zweidimensionalen Beispiel landen beide um ϵ verschobenen Knoten der Originalfläche in einer Voronoi-Region, die nicht zu denen der adjazenten Kanten (Dreiecke im Dreidimensionalen) gehören. Das ϵ muss für diese Knoten soweit verringert werden, dass der verschobene Knoten innerhalb der richtigen Voronoi-Regionen bleibt. Da die Bestimmung der Voronoi-Regionen der Dreiecke nicht trivial und zudem rechenaufwendig ist, werden in der Arbeit von Cohen et al. dafür ein analytisches und ein numerisches Verfahren vorgestellt.

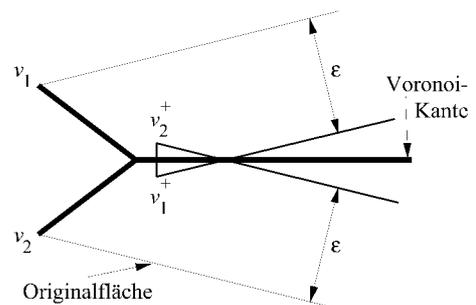


Abb. 2.15: Voronoi-Regionen der Netzdreiecke bestimmen die Verringerung der Hüllenverschiebung, basiert auf Bild aus [24]

Nach der Erstellung der Hüllen werden zwei Vereinfachungsansätze vorgestellt, die Dreiecke oder Knoten eliminieren und das entsprechende Loch wieder triangulieren^G. Dabei ist ohne weitere Einschränkung nur darauf zu achten, dass die vereinfachte Oberfläche nicht die einschließende Hülle verlässt. Diese Vereinfachungsansätze können durch andere ersetzt werden, solange sie die Einhüllung berücksichtigen. Varshney schlägt in [25, 26] ein globales Verfahren vor, das alle möglichen Dreiecke mit Eckpunkten aus dem Originalnetz und Lage innerhalb der Hülle für die Vereinfachung in der Reihenfolge heranzieht, in der sie möglichst viele Originaldreiecke ersetzen.

Durch die Hüllen wird garantiert, dass das Netz sich nicht weiter als ϵ von der Originalposition entfernt und damit der Hausdorff-Abstand maximal ϵ ist. Da gerade an scharfen Kanten und spitzen Ecken eine Durchdringung der Hüllen wie in Abbildung 2.15 zu erwarten ist, wird dort durch die Verringerung des ϵ genau an diesen wichtigen Orten besonders behutsam vereinfacht. Zudem erhalten die sich nicht überschneidenden Hüllen die Topologie des Netzes. Durch die Topologieerhaltung und der strengen Fehlergrenze ist die Netzgüte insbesondere für behutsame Vereinfachungen groß, wird aber die drastische Vereinfachung erschwert. Auch topologisch unwichtige Eigenschaften des Originalnetzes schränken den Vereinfachungsprozess ein und machen den Prozess weniger robust. Zur Erzeugung der Hüllen ist ein orientiertes Netz im Gegensatz zu allen anderen in dieser Arbeit vorgestellten Verfahren nötig. Insbesondere die Erzeugung der Hüllen bewirkt einen hohen Rechenaufwand und eine schwere Implementierbarkeit.

2.6 Kantenkontraktion (edge collapse, edge contraction)

Die *Kantenkontraktion* kann als Spezialfall der Knotenhäufung aufgefasst werden, bei dem nicht alle Knoten in einem Voxel^G auf einen Knoten zusammenfallen, sondern genau zwei auf einer Kante liegende Knoten. Daher sind diese Gruppen in der Tabelle 2.1 bei den Mechanismen unter Knotenverschmelzung zusammengefasst. Einen weiteren Spezialfall unter den knotenverschmelzenden Verfahren stellen die hier nicht weiter behandelten *flächenkontrahierenden* dar. Hierbei werden genau drei Knoten eines Dreiecks auf einen gemeinsamen zusammengezogen. Wie bei der Kantenkontraktion werden dabei die Dreiecke aus dem Netz entfernt, die zu einem Punkt oder zu einem Geradenstück degenerieren. Statt also der im Normalfall zwei eliminierten Dreiecke bei der Kantenkontraktion würden dabei für gewöhnlich gleich drei Dreiecke je Operation aus dem Netz entfernt werden. Die Qualität verringert sich aber bei der Erhöhung der Anzahl zusammenfallender Knoten.

Die Aufteilung der Verfahren geschieht in dieser Arbeit nach ihren im Verfahren verwandten elementaren Operationen. Auch in diesem Abschnitt steht die Kantenkontraktion im Vordergrund, die eine Kante auf einen Knoten zusammenfallen lässt. Jedoch beinhaltet der erste Ansatz "Mesh Optimization" weitere Netzoperationen, die aber nicht vordringlich der Vereinfachung dienen und der dritte Ansatz von Garland und Heckbert erweitert die Idee der Kantenkontraktion auf imaginäre Kanten. *Imaginäre Kanten* werden dabei Knotenpaare genannt, die in einem wählbaren Abstand zueinander liegen.

Diese Klasse von Verfahren, zu denen auch die wichtigen von Ronfard und Rossignac [16], Popović und Hoppe [27] und Lindstrom und Turk [28] gehören, erzeugen die beste Qualität an Vereinfachungen.

2.6.1 Mesh Optimization

Hoppe et al. [29]

Das Verfahren der Netzoptimierung (mesh optimization) dient zur Verbesserung von Netzen, kann aber bei geeigneter Wahl der Parameter zur Vereinfachung genutzt werden. Die Optimierung geht nur, wenn neben dem Dreiecksnetz weitere Informationen vorliegen. Hier wird von einer Punktwolke (scattered data points) im Raum ausgegangen, die insbesondere von 3D-Scannern gemessen werden.

Um Eigenschaften des Netzes zu quantifizieren wird eine Energie mit mehreren die verschiedenen Netzeigenschaften bewertenden Termen verwendet. Ziel ist es nun, die Energie des Ausgangsnetzes ohne Änderung der Mannigfaltigkeiten zu optimieren, indem das Netz sich der Punktwolke mit möglichst wenig Dreiecken anpasst. Ohne Punktwolke kann das Verfahren auch zur reinen Vereinfachung eingesetzt werden. Hierbei werden dann die Knoten des Ausgangsnetzes und möglicherweise weitere Punkte auf dem Netz statt der Punktwolke herangezogen.

Die zu minimierende Energie E besteht dabei aus drei konkurrierenden Teilen.

$$E = E_{dist} + E_{rep} + E_{spring}$$

E_{dist} beschreibt die Abstandsenergie, zur Bestrafung der Knotenzahl dient E_{rep} und zur besseren Findung von lokalen Minima steht E_{spring} ebenfalls im globalen Energieterm und fördert zudem eine glatte Oberfläche und Dreiecke mit ausgeglichenem Seitenverhältnis.

Sei wie in Abschnitt 2.1.1 eingeführt $X = \{x_1, \dots, x_n\}$ die Punktwolke als Ausgangsbasis zur Abstandsbestimmung, $V = \{v_1, \dots, v_m\}$ die Geometrie des Netzes $M = (K, V)$ und $\phi_V(|K|)$ die Netzoberfläche.

$$E_{dist}(K, V) = \sum_{i=1}^n d^2(x_i, \phi_V(|K|))$$

ist dann die Summe der Abstandsquadrate (sum of squared distances) der Punktwolke X von der Netzoberfläche $\phi_V(|K|)$ und ist entsprechend gering, umso näher das Netz der Punktwolke ist. Dieser Energieterm ist ähnlich dem vereinfachten quadratischen Fehlermaß aus Abschnitt 2.1.2. E_{dist} ist aber nicht gemittelt und unsymmetrisch, da die Punktwolke kein Netz repräsentiert.

$$E_{rep}(K) = c_{rep}m$$

bestraft durch eine zur Knotenanzahl m proportionale Energieerhöhung und steht somit der Abstandsenergie E_{dist} entgegen, die für gewöhnlich bei Verringerung der Knotenzahl größer wird. Der Parameter c_{rep} dient also der Gewichtung der Netzvereinfachung gegenüber der geometrischen Genauigkeit.

$$E_{spring}(K, V) = \sum_{\{i,j\} \in K} \kappa \|v_i - v_j\|_2^2$$

summiert die Kantenlängenquadrate des aktuellen Netzes und arbeitet dementsprechend wie eine Feder der Länge 0 an jeder Kante. Sie steht der Verzerrung der Dreiecke entgegen. Empirisch hat sich herausgestellt, dass κ bei Verringerung der Knotenzahl fortlaufend an die Netzkomplexität angepasst heruntergesetzt werden sollte, um den Einfluss nach Findung eines lokalen Minimums herabzusetzen.

Um nun die Energie zu verringern, sind zwei verschachtelte Probleme zu lösen. Zum einen ist bei fester Topologie K die Knotenplatzierung V zu optimieren und zum anderen bei festem V die Topologie K so anzupassen, dass $E(V, K)$ möglichst gering ausfällt.

Für die Topologieveränderung werden drei Elementartransformationen entsprechend Abbildung 2.16 zugelassen: Kantenkontraktion (edge collapse), bei der eine Kante auf einen Punkt zusammenfällt und für gewöhnlich mit der Kante zwei Dreiecke eliminiert, Kantenteilung (edge split), die das Netz sich der Punktwolke besser anpassen lässt und Kantenwechsel (edge swap), der zwar über Kantenteilung und Kantenkontraktion nachgebildet werden kann, aber bei Hinzunahme bessere lokale Minima findet. Dieses Verfahren steht unter der Überschrift Kantenkontraktion, da die Netzvereinfachung hauptsächlich auf dieser Topologieveränderung beruht.

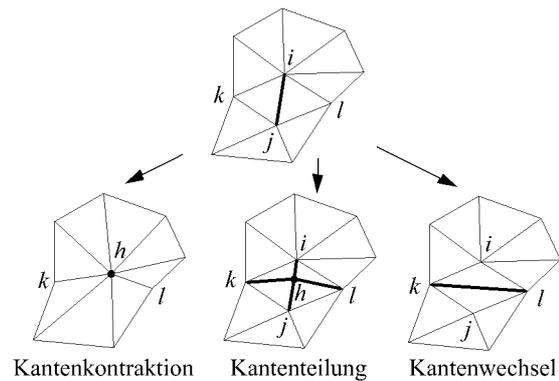


Abb. 2.16: Elementartransformationen, Bild aus [29]

Um die Mannigfaltigkeiten zu bewahren, ist eine Kantenkontraktion nur erlaubt, wenn die Nachbarknoten von i und j eine gemeinsame Fläche mit ihnen aufspannen, und ein Kantenwechsel nur, wenn die neue Kante $\{k, l\}$ nicht schon eine Kante des aktuellen Netzes ist. Eine Transformation wird durchgeführt, wenn $E(K, V)$ sich verringert. Zur stochastischen Optimierung werden die Orte der Transformationstests zufällig ausgewählt.

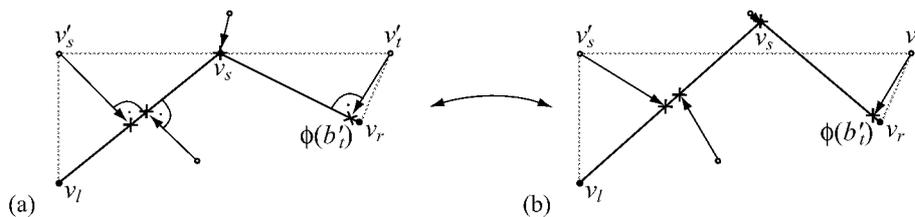


Abb. 2.17: (a) Bestimmung der nächstliegenden Projektion aller Punkte (b'), (b) Minimierung der Abstandsquadrate und der Federkräfte der variablen Kanten durch Verschiebung von v_s

Die Energieminimierung über die Geometrie ist wiederum eine Iteration über zwei Aktionen gemäß Abbildung 2.17. Zuerst werden die Punkte X mittels baryzentrischer Koordinaten optimal auf das Netz mit festen Knoten abgebildet und dann bei fester Projektion die Knoten so verschoben, dass der quadratische Projektionsabstand minimal ist. Beide Schritte werden bis zur Konvergenz wiederholt.

Abschließend lässt sich festhalten, dass das Verfahren der Netzoptimierung sehr gute Ergebnisse liefert, dies aber auch unter Ausnutzung von Lokalitäten, die Punkte festen Flächen zuordnet, mit einem sehr hohen Rechenaufwand. Aber auch so wird nicht das globale, sondern nur ein lokales Minimum gefunden. Bei dieser Methode bleiben Mannigfaltigkeiten erhalten. Es dient neben der Netzvereinfachung auch der Optimierung eines Netzes unter Berücksichtigung von geometrischer Genauigkeit und Komplexität.

2.6.2 Progressive Meshes

Hoppe [5, 30]

Dieses Verfahren setzt auf dem zuvor beschriebenen auf, beschränkt sich dabei auf die Vereinfachung und führt eine Datenstruktur zur Verwaltung von Detailstufen ein.

Zur Netzvereinfachung wird die Elimination von Kanten herangezogen (edge collapse, *ecol*), bei der an einer Kante zwei Knoten auf einen zusammenfallen und somit mit dieser Kante zwei Dreiecke aus dem Netz verschwinden (an Rändern nur ein Dreieck, in entarteten Bereichen auch mehrere). Da hier eine reine Vereinfachung und keine Optimierung des Netzes wie im vorangegangenen Artikel angestrebt wird, ist dies auch die einzige Transformation zur Vereinfachung des Netzes.

Das Gegenstück zur Elimination einer Kante ist das Aufspalten eines Punktes (vertex split, *vsplit*). Hierbei entsteht aus einem Knoten ein zweiter und damit die verbindende Kante mit ihren Dreiecken. Der Vorgang ist damit reversibel. Grafisch ist dies in Abbildung 2.18 dargestellt.

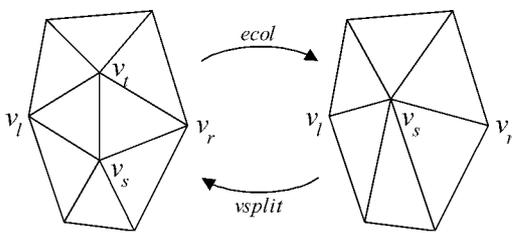


Abb. 2.18: Kantenkontraktion (*ecol*) und Knotenspaltung (*vsplit*), Bild aus [5]

Sei \hat{M} das Ausgangsnetz, dann lässt sich die Vereinfachung als Sequenz

$$(\hat{M} = M_n) \xrightarrow{ecol_{n-1}} \dots \xrightarrow{ecol_1} M_1 \xrightarrow{ecol_0} M_0$$

vorstellen, bei der M_0 das grobe Anfangsnetz ist. $ecol_i(v_s, v_t)$ würde dann zum Beispiel wie in obiger Abbildung v_t auf v_s fallen lassen und damit neben dem Knoten v_t die Dreiecke $\{v_s, v_t, v_l\}$ und $\{v_t, v_s, v_r\}$ eliminieren.

Der inverse Vorgang der Punktspaltung stellt sich als

$$M_0 \xrightarrow{vsplit_0} M_1 \xrightarrow{vsplit_1} \dots \xrightarrow{vsplit_{n-1}} (M_n = \hat{M})$$

dar und ergibt am Ende wieder das Ausgangsnetz. Für die Spaltung muss $vsplit_i$ der neue Knoten v_t , möglicherweise die neuen Koordinaten von v_s und die entstehenden Dreiecke bekannt sein. Zudem sind die Dreiecke anzugeben, bei denen der Knoten v_s durch v_t ersetzt wird. Falls dem Netz Eigenschaften mitgegeben wurden, sind auch diese Daten nötig.

Die “Progressive Mesh”-Repräsentation (PM) ist also

$$(M_0, \{vsplit_0, \dots, vsplit_{n-1}\}).$$

Die Kantenauswahl geschieht wie im vorherigen Verfahren über dieselbe Energiefunktion. Lediglich der Term E_{rep} wird nicht mehr benötigt, da keine Netzoptimierung vorgesehen ist und somit jeweils die Kante ausgewählt wird, die die geringste Gesamtenergie nach Löschung hervorbringt. Um Netzattribute zu berücksichtigen, ist hingegen ein diesen Gesichtspunkt bewertender Term eingeführt worden.

Kontinuierliche Attribute können mittels einer weiteren zu addierenden Energie in der Energieberechnung berücksichtigt werden, die wie die geometrische Abstandsenergie die gesamten Attributunterschiede zum Originalnetz aufsummiert. Ein Faktor gewichtet diese Energie relativ zu den anderen Energien. Die Projektion wird der Einfachheit halber von der geometrischen Betrachtung übernommen.

Über eine weitere Energie wird auch dem geometrischen Verlauf der scharfen Kanten Rechnung getragen. Eine Kantenkollabierung wird aber verhindert, wenn die Topologie dieser in den Attributänderungen unstetigen Bereiche verändert wird.

2.6.3 Surface Simplification Using Quadric Error Metrics

Michael Garland and Paul S. Heckbert [31]

Auch dieses Verfahren soll hier kurz vorgestellt werden, um im Vergleich auf Seite 28 berücksichtigt werden zu können. Eine ausführliche Beschreibung findet man in Abschnitt 3.1, da dieses das für die dieser Arbeit zugrundeliegende Anwendung als optimal erscheinende Verfahren darstellt und somit als Basis für den Vereinfachungsalgorithmus implementiert wurde.

Die grundlegende Operation zur Netzvereinfachung ist ähnlich der im vorangegangenen Abschnitt verwandten Kantenkontraktion (edge collapse). Auch hier werden zwei Knoten zu einem zusammengeführt. Dabei müssen diese aber nicht unbedingt auf einer Kante liegen, sondern brauchen nur nah genug beieinander zu liegen. Aus diesem Grund spricht man auch nicht von einer Kantenkontraktion, sondern von einer Knotenverschmelzung (vertex pair contraction) als elementare Operation zur Simplifizierung. Daraus resultiert, dass auch nicht zusammenhängende Gebiete verschmelzen können, also Mannigfaltigkeiten nicht unbedingt erhalten bleiben.

Zur Bestimmung des geometrischen Fehlers werden Quadriken an den Knoten eingesetzt, die es zum einen ermöglichen, den optimalen Platz für den aus einer Verschmelzung resultierenden Knoten zu berechnen und über diesen auch den Fehler anzugeben (quadric error metric). Quadriken ergeben sich aus der Summe der quadratischen Abstände zu den an einen Knoten angrenzenden Flächen. Zur Vereinfachung werden Quadriken nur initial für jeden Knoten berechnet und dann nach einer Verschmelzung summiert, was zwar Flächen häufiger berücksichtigen kann, dafür aber die Vereinfachung schneller durchführen lässt.

Die Geschwindigkeit ist dann auch schneller als bei den qualitativ hochwertigen Verfahren von Hoppe (Abschnitt 2.6.1, 2.6.2), die Netzqualität nach Vereinfachung aber generell besser als bei den schnellen Verfahren von Rossignac und Schroeder (Abschnitt 2.4.1, 2.5.1).

Der Ablauf des Algorithmus lässt sich wie folgt zusammenfassen:

1. Berechne für jeden Knoten seine Quadrik
2. Bestimme alle erlaubten Knotenverschmelzungen
3. Berechne für jedes Knotenverschmelzungspaar den optimalen Verschmelzungspunkt und bestimme darüber den dann resultierenden Fehler
4. Sortiere alle Knotenpaare nach ihren Kosten für eine Verschmelzung
5. Füge iterativ Knotenpaare mit dem minimalsten Fehler zusammen und bestimme in deren Umgebung Quadriken, optimalen Verschmelzungspunkt und Fehler neu

2.7 Vergleich und Bewertung der Verfahren

2.7.1 Eingabedaten

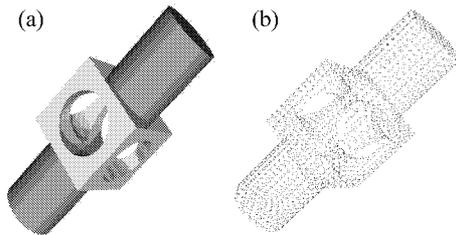


Abb. 2.19: (a) Modellierter Oberfläche und (b) Punktwolke eines 3D-Scans des gleichen Objekts, Bild aus [32]

Die Daten, aus denen Oberflächen in der Computergrafik entspringen, sind vielfältiger Herkunft. In der nebenstehenden Abbildung 2.19 ist in (a) eine Oberfläche modelliert worden und in (b) über eine *Punktwolke* erst noch zu erzeugen. Die Punktwolke resultiert aus einem 3D-Scan eines realen Objekts und muss durch ein Verfahren erst noch in eine Oberfläche überführt werden [32]. Andere *Eingabedaten* könnten *analytische Funktionen*, *Messwerte* (zum Beispiel Volumendaten^G oder Höhenfelder (height fields)) oder Daten aus *Computersimulationsrechnungen* sein.

Die *Höhenfelder*, die zum Beispiel bei Flugsimulatoren zur Geländemodellierung (terrain rendering) eingesetzt werden, haben die spezielle Eigenschaft, dass sie eine Abbildung einer Teilmenge des \mathbb{R}^2 in die reellen Zahlen darstellen und dadurch sehr effizient unter anderem mit Methoden der Bildverarbeitung vereinfacht werden können. Für diese Art von Daten gibt es auch schnelle betrachterabhängige Vereinfachungsverfahren (siehe Abschnitt 2.8.3). Ein Beispiel solch einer Geländeoberfläche zeigt Abbildung 2.20 in sechs verschiedenen Detailstufen. Zur Vereinfachung ist aber kein spezielles Verfahren für Höhenfelder angewandt worden, sondern das in dieser Arbeit ausführlich beschriebene und implementierte generelle Verfahren (siehe Abschnitt 3.1). Die Abbildung soll einen Vorgeschmack auf die Qualität dieses Verfahrens basierend auf dem Quadrik-Fehlermaß geben.

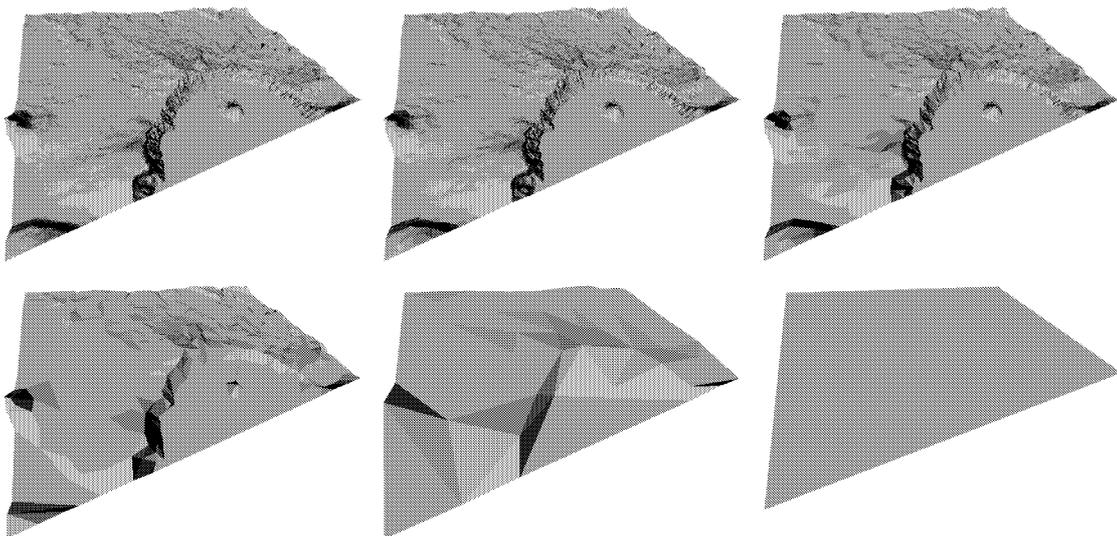


Abb. 2.20: Eine Geländeoberfläche aus ursprünglich 100.000 Knoten ist mittels des in dieser Arbeit ausführlich in Abschnitt 3.1 beschriebene Verfahrens auf 10.000, 2.000, 500, 50 und 4 Knoten vereinfacht worden

Wie die Höhenfelder, so besitzen auch die allgemeineren *parametrisierten Flächen*, die über eine Abbildung einer Teilmenge des \mathbb{R}^2 in den reellen Raum \mathbb{R}^3 angegeben werden, spezialisierte effiziente Simplifizierungsverfahren, die die Einfachheit der Repräsentation und die kompakte Speicherung solcher Daten ausnutzen. Aus diesem Grund sind einige Verfahren entwickelt worden, die allgemeine

Dreiecksnetze in stückweise parametrisierte Flächen überführen, um sie als solche besser und schneller vereinfachen zu können.

Aus den für die dieser Arbeit zugrundeliegende Anwendung wichtigen *Volumendaten*^G wird erst über bestimmte Verfahren eine Isofläche als Oberfläche gewonnen. Weit verbreitet ist dafür der Marching-Cube-Algorithmus, der in Abschnitt A.1.4 kurz beschrieben wird.

Analytische Funktionen können besser bei der Erzeugung approximiert und vereinfacht werden. Eine Kugel sollte also nicht erst als feines Netz trianguliert werden, um dann nach Bedarf vereinfacht zu werden, sondern sollte bei veränderten Randbedingungen neu trianguliert werden.

Diese Arbeit beschränkt sich auf diskrete Daten, die heutzutage den Großteil der darzustellenden Objekte beschreiben. Bei den Daten wird unterschieden, ob sie auf einem *regelmäßigen, unregelmäßigen* oder gar keinem *Gitter* liegen und welche weiteren Attribute neben der Geometrie vorliegen.

Bei sehr großen Datenmengen, die durch feine Abtastung entstehen können, kann es vorkommen, dass die Daten insbesondere mit den zusätzlichen Strukturen für die Simplifizierung nicht mehr in den Hauptspeicher passen. Verfahren, die eine Bewertung aller Eingabedaten vornehmen, um dann ortsungebunden im Netz zu vereinfachen, sind für solch große Datenmengen nicht geeignet. Es gibt speziell für diesen Zweck ausgelegte Verfahren, die die Eingabedaten nicht im Hauptspeicher halten müssen (out-of-core simplification), sondern nur einmal durchlaufen und dabei sofort das vereinfachte Netz konstruieren. Sie setzen zum Beispiel auf das gebietszerlegende Verfahren der Knotenhäufung auf [33]. Durch ihre direkte Vereinfachung sind solche Verfahren auch geeignet, sofort im Anschluss der Datenerfassung zum Beispiel eines 3D-Scanners eine Vereinfachung ohne Zwischenspeicherung durchzuführen.

Die allgemeine Form der Netzdefinition aus 2.1.1 wird nur von dem in Abschnitt 2.4.1 vorgestellten Verfahren von Rossignac und Borrel als Ausgabedaten unterstützt, da nur dieses auch dreiecksungebundene Kanten und Knoten eines zweidimensionalen schlichten Simplex ausgibt. Die Arbeiten von Hoppe aus Abschnitt 2.6.1 und 2.6.2 berücksichtigen zumindest in den Eingabedaten isolierte Knoten, die bei überabgetasteten Oberflächen zur Optimierung des Netzes und dessen Simplifizierung herangezogen werden. Das implementierte Verfahren lässt nur reine Komplexe zu, die, wie in Abschnitt 2.1.1 beschrieben, keine isolierten Kanten und Knoten erlauben.

Die Einschränkung des implementierten Verfahrens ist für die zugrundeliegende Anwendung unkritisch, da der verwendete Marching-Cube-Algorithmus nur Dreiecke und keine isolierten Elemente ausgibt. Viel wichtiger ist die Generalität des implementierten Verfahrens im Bereich der Mannigfaltigkeiten. Zwar erzeugt der Marching-Cube-Algorithmus Mannigfaltigkeiten oder Mannigfaltigkeiten mit Rand, aber die Verschmelzung unverbundener Regionen ist nur über kritische Knoten einer Nicht-Mannigfaltigkeit möglich (siehe zum Beispiel Knoten \bar{v} in Abbildung 3.1 (b)).

Das in der Implementation verwendete Verfahren setzt daher keine Mannigfaltigkeiten voraus, was die Eingabedaten weniger einschränkt und bei der Vereinfachung mehr Freiheiten bei den Netzoperationen bietet und somit optimaler arbeiten kann. Die Vereinfachung erhält mehr Spielraum und kann dadurch das Netz geometrisch exakter wiedergeben. Starke Vereinfachungen topologisch komplexer Modelle sind nur unter Zulassung von Topologieveränderungen möglich. Das einzige weitere vorgestellte Verfahren, das diesbezüglich unkritisch ist, ist das von Rossignac und Borrel aus Abschnitt 2.4.1.

Die Erhaltung von Mannigfaltigkeiten ist hingegen wichtig für Anwendungen im Bereich der Finiten Elementen Analysis oder der Visualisierung mittels Radiosity-Verfahren.

Eine zusammenfassende Einteilung der vorgestellten Verfahren in topologieerhaltende, -verändernde und -tolerante Methoden ist in Tabelle 2.1 auf Seite 38 zu sehen.

Zusätzlich zur Darstellungsmöglichkeit von aus Dreiecken bestehenden Isoflächen des Marching-Cube-Algorithmus wurde die Anwendung auch um die Möglichkeit erweitert, direkt Dreiecksnetze einzulesen und zu speichern. Neben *Datenformaten* kommerzieller Produkte zur Speicherung von Dreiecksnetzen haben sich das freie SMF- und PLY-Format etabliert. Die Programmimplementation unterstützt zur Zeit das generelle SMF-Format [34] und hat ein eigenes eingeführt, um die berechnete kontinuierliche Vereinfachung zu sichern.

2.7.2 Datenstrukturen

Die Art der Eingabedaten werden von den Messapparaturen, Simulationsrechnungen oder der Modellierungssoftware bestimmt, die *Datenstruktur* zur Speicherung der Vereinfachung im Hauptspeicher oder auch extern als Ausgabedaten wird von dem Simplifizierungsverfahren und der Anwendung geprägt, für die die Vereinfachung erstellt wird. Die Datenstruktur hängt stark davon ab, ob die Simplifizierung statisch, dynamisch oder betrachterabhängig ist.

Statische Simplifizierung und deren Datenstruktur entspricht den ursprünglichen handerzeugten *diskreten Vereinfachungsstufen* (level of detail, LOD). Die Visualisierung entscheidet nach der Erzeugung über die für die Darstellung angemessene vorberechnete Detailstufe. Die Stufen liegen alle vor der eigentlichen Visualisierung vor und entkoppeln die Vereinfachung von der Visualisierung, so dass die Vereinfachung nicht echtzeitfähig sein muss. Diese einfache Art von Datenstruktur mittels diskreter Detailstufen wird von aktueller Grafikhardware unterstützt und ermöglicht den schnellen Zugriff auf die verschiedenen Modellapproximationen zum Beispiel über separate Display-Listen^G und ergibt dadurch eine schnelle Visualisierung auch größerer Datenmengen. Diese statische bzw. diskrete Datenstruktur ist mit allen vorgestellten Verfahren zu erzeugen, bietet sich aber insbesondere für das einhüllende Verfahren aus Abschnitt 2.5.2 an, da es eine globale Fehlergrenze mitbringt und keinen kontinuierlichen Vereinfachungsverlauf besitzt. Bei den Ausgabedaten unterstützt unter anderem die bekannte offene standardisierte Beschreibungssprache allgemeiner dreidimensionaler Szenen namens *VRML (Virtual Reality Modeling Language)* [35] statische Simplifizierungsstufen.

Die *dynamische* oder auch *kontinuierliche Vereinfachung* unterscheidet sich von dem statischen Ansatz dadurch, dass die Datenstruktur keine festen Approximationsstufen des Modells sondern ein kontinuierliches Spektrum an Netzapproximationen beinhaltet (multiresolution modeling). Die gewünschte Stufe kann zur Laufzeit bestimmt werden und bietet eine sehr viel feinere Granularität an Detailstufen als bei dem statischen Ansatz, was eine exaktere Abstimmung an Geschwindigkeits- und Qualitätswünschen erlaubt und das visuell störende Umspringen zweier Vereinfachungsstufen stark vermindert (siehe auch Abschnitt 2.8.1). Der bekannteste Vertreter dieser Datenstruktur ist das Progressive Mesh (PM), das in Abschnitt 2.6.2 genauer vorgestellt wurde. Die Klasse der Verfahren, die für die Erzeugung solcher Datenstrukturen in Frage kommen, sind vordringlich die, die je Detailstufe nur geringe Veränderungen am Netz vornehmen und damit leicht umkehrbar sind. Zu dieser Klasse gehören also insbesondere die kantenkontrahierenden Ansätze aus Abschnitt 2.6, da die elementare Operation nur lokale Modifikationen vornimmt und dabei wenig am Netz verändert. Auch in der dieser Arbeit zugrundeliegenden Implementation ist solch eine Datenstruktur Grundlage für die adaptive Netzanpassung (siehe dazu 3.1.6). Zudem bieten sich diese Datenstrukturen für die in Abschnitt 2.8.2 erläuterte progressive Übertragung an.

Die Datenstrukturen zu *betrachterabhängigen Vereinfachungen* sind die kompliziertesten, sollen sie doch zur Laufzeit kontinuierliche Approximationen liefern, die für die aktuelle Betrachterposition und -blickrichtung optimiert sind. Sie stellen damit eine Untergruppe der kontinuierlichen Datenstrukturen dar. Der Betrachterabhängigkeit ist ein eigener Abschnitt 2.8.3 gewidmet, in dem auch Beispiele betrachterabhängiger Verfahren kurz vorgestellt werden. Eine Skizze für eine betrachterabhängige Datenstruktur stellt Luebke in [36] vor, die eine *hierarchische dynamische Vereinfachung* (hierarchical dynamic simplification, HDS) ermöglicht. Diese Datenstruktur kann mit allen knotenverschmelzenden Verfahren, zu denen die Klasse der Verfahren aus Abschnitt 2.4 aber auch die kantenkontrahierenden Ansätze aus Abschnitt 2.6 zählen, aufgebaut werden. Wichtig ist die eindeutige Abhängigkeit mehrerer Knoten einer vorangegangenen Detailstufe zu genau einem Knoten in der vereinfachten Stufe, zum Beispiel der eindeutige Bezug der Kantenknoten zu ihrem Verschmelzungsknoten. Nach dem Aufbau der Datenstruktur als Knotenbaum kann mittels verschiedener Fehlermaße, die Entfernung, Silhouette oder Bildfehler bewerten, dynamisch eine Detailstufe erzeugt werden, die sich in Abhängigkeit von der Betrachterposition und -blickrichtung ändert. Hierzu wird im Abhängigkeitsbaum an den Knoten weiter verzweigt, an denen das gewünschte Fehlermaß einen noch zu hohen Wert aufweist. Abbildung 2.21 zeigt einen solchen Knotenbaum für eine kantenkontrahierende Vereinfachung, deren aktive Knoten, also die Knoten, die zur Zeit mit ihren Kanten angezeigt werden, hinterlegt sind. Der Baum muss dabei nicht wie in der Zeichnung Blätter in gleicher Höhe besitzen.

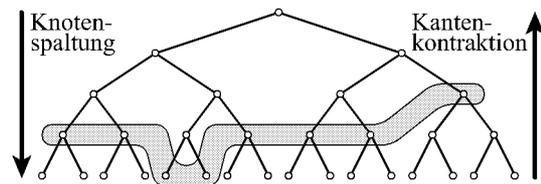


Abb. 2.21: Knotenbaum einer kantenkontrahierenden Vereinfachung mit hinterlegten aktiven Knoten

Die Netzvereinfachung wird oft zur flüssigen Darstellung herangezogen. Ein weiterer Aspekt ist aber auch die Speicherplatzersparnis wie sie auch die in Abschnitt 2.8.5 vorgestellte Kompression zum Ziel hat. Eine bekannte Datenstruktur dafür ist der *Dreiecksstreifen*, bei dem pro Dreieck nur ein Knoten abgelegt werden muss. Die beiden anderen Knoten werden dem vorangegangenen Dreieck entnommen.

Für die Erzeugung solcher speicherplatzeffizienten Datenstrukturen ist ein möglichst *gleichmäßiges Netz* erforderlich. Gleichmäßig bzw. *regulär* heißt dabei, dass jeder Knoten genau sechs Valenzknoten besitzt. Als *Valenzknoten* eines Knotens werden dabei die bezeichnet, die mit ihm auf einer gemeinsamen Kante liegen. Diese sechs über eine Kante verbundenen Nachbarknoten ergeben sich daraus, dass nach [6] jede Voronoi-Region^G im Mittel sechs Kanten hat und dadurch jeder Knoten in der dazugehörigen Triangulierung^G in sechs Kanten vorkommt. Solch ein reguläres Gitter, welches auch möglichst vom Vereinfachungsverfahren aus Abschnitt 2.3.1 erzeugt wird, existiert nur für Netze, die topologisch äquivalent zu einer Kugel oder einem Ring sind. Ist das Dreiecksnetz nicht dazu homöomorph, müssen auch Knoten mit weniger Nachbarknoten zugelassen werden und es entsteht ein so genanntes *semi-reguläres Netz*.

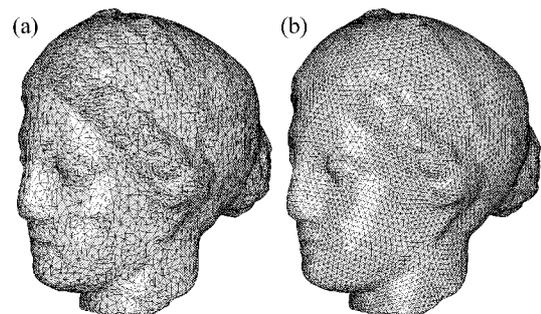


Abb. 2.22: (a) Originalnetz ist in (b) regulärisiert worden, Bild aus [37]

Wird ein Dreiecksnetz wie in Abbildung 2.22 in ein semi-reguläres überführt (remeshing, resampling, retiling), so werden für gewöhnlich nicht die Knotenposition und -anzahl erhalten, so dass dadurch

Fehler eingebracht werden, die aber durch Heruntersetzen der Maschenweite beliebig verringert werden können [37].

Diese semi-regulären Netze haben für die Vereinfachung weitere wichtige Bedeutungen. Simplifizierungsverfahren für parametrisierte Flächen sind wie die der Höhenfelder sehr effizient in Geschwindigkeit und Speicherplatz. Durch die *Regularisierung* der Netze besteht nun die Möglichkeit einer einfacheren *Parametrisierung* bzw. stückweisen Abbildung auf parametrisierte Flächen, auf die die entsprechenden Vereinfachungsansätze angewendet werden können [38]. Die Parametrisierung wird für die Detailstufen hierarchisch angewandt, da semi-reguläre Netze eine einfache Zuordnung von Gebieten einzelner Detailstufen erlauben. Bei der stückweisen Abbildung mit anschließender getrennter Simplifizierung besteht das größte Problem in der anschließenden Zusammenführung der vereinfachten Netzteile.

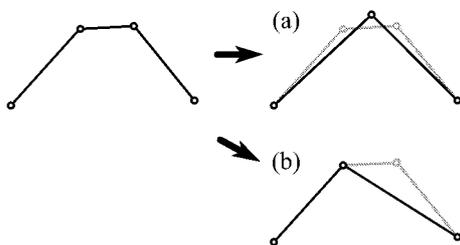


Abb. 2.23: Netz im Zweidimensionalen wird durch Kantenkontraktion in (a) mit optimiertem Verschmelzungspunkt und in (b) positionstreu vereinfacht

Wie bei den regulären Netzen, deren Knoten auch in der feinsten Detailstufe nicht die Knotenposition des Originalnetzes annehmen müssen, ist die *Positionstreue* der Netzknoten ein weiterer Punkt, der die Datenstruktur prägt. Werden Knoten während der Simplifizierung verschoben, sind also die Knoten einer Detailstufe keine Teilmenge der Originalknoten mehr, müssen die Knotenpositionen für die kontinuierlichen und betrachterabhängigen Datenstrukturen für jede Stufe mit abgelegt werden. Bleiben sie hingegen fest, würde zum Beispiel das Originalnetz mit seiner Geometrie ausreichen und nur topologische Änderungen abgelegt werden müssen. Positionstreuere Datenstrukturen sind kompakter, bieten aber nicht so eine gute Qualität, da zum Beispiel

durch die Verschiebung des Verschmelzungspunktes bei knotenkontrahierenden Verfahren die Netzgeometrie, wie in Abbildung 2.23 zu sehen, besser wiedergegeben werden kann.

2.7.3 Fehlermaße

In diesem Abschnitt geht es um die heuristischen Fehlermaße der vorgestellten Verfahren und nicht um globale Fehler zur Qualitätsbestimmung. Die heuristischen Fehlermaße werden hier klassifiziert und bewertet. Zusammenfassend ist dies in Abschnitt 2.7.5 nachzulesen.

Der Fehler bzw. die Kosten eines Simplifizierungsverfahrens sollen widerspiegeln, wie stark sich die Netzvereinfachung bzw. die elementaren Operationen auf das Modell auswirken. Die Berücksichtigung von Netzattributen, wie Farben und Texturkoordinaten, spielen für die Anwendung, die hinter dieser Arbeit steht, keine Rolle. Daher wird darauf auch nur am Rande eingegangen. Das Verfahren von Hoppe aus Abschnitt 2.6.2 bezieht die Attribute über einen weiteren Energieterm in seinen Gesamtfehler ein. Wie das in der Implementation verwendete Quadrik-Fehlermaß elegant zur Berücksichtigung von Attributen erweitert werden kann, wird in [39, 40] beschrieben. Wieder ganz andere Ansätze sind für betrachterabhängige Vereinfachungen gefragt (siehe Abschnitt 2.8.3). Da aber unter anderem die konkrete Anwendung in einer Virtual Reality-Umgebung^G eingesetzt wird und sich dort die Betrachterposition fortlaufend ändert, kommen auch diese Ansätze für die Implementation nicht in Frage. Bei Außerachtlassung von Netzeigenschaften und der betrachterunabhängigen Vorgehensweise ist der geometrische Fehler ein gutes Maß zur Qualitätskontrolle. Die auf dem geometrischen Fehler aufsetzenden Verfahren versuchen die im Abschnitt 2.1.2 vorgestellten globalen Fehlermaße wie den

maximalen oder durchschnittlichen Abstand zweier Netze als Bewertungsgrundlage heranzuziehen, indem sie die globalen Maße auf verschiedene Weise abschätzen.

Will man den nahezu exakten geometrischen Fehler zur Kostenberechnung heranziehen wie es Hoppe am ehesten macht (siehe Abschnitt 2.6.1), ist die Netzqualität der Vereinfachung sehr gut, aber die Rechenzeit durch die vielen Abstandsberechnungen entsprechend groß. Aber auch der Ansatz von Hoppe ist ein Greedy-Verfahren, welches nicht das Fehlermaß global minimiert, sondern nur nach jedem Schritt eine Kostenanalyse durchführt und den Schritt mit den zu diesem Zeitpunkt geringsten Kosten ausführt.

Eine schnellere Kostenberechnung bieten Verfahren, die den Abstand zu Oberflächenapproximationen als Fehlermaß heranziehen. Eine einfache Approximation wäre eine Ebene durch Nachbarknoten eines Knotens wie bei Schroeder (siehe Abschnitt 2.5.1), aber auch Flächen höherer Ordnung sind denkbar, wenn auch rechenintensiver. Diese Verfahren haben den Nachteil, dass die reine Simplifizierung generell nur auf Mannigfaltigkeiten ausgeführt werden kann. Noch schwerer wiegt aber, dass dieses Fehlermaß als einziges der hier vorgestellten nur den Fehler zwischen zwei Iterationsstufen misst, also das, was bei vielen Verfahren eine elementare Operation verursacht, und somit den Bezug zum Originalnetz verliert. Es wird zwar für jeden Schritt die Operation mit dem geringsten Fehler ausgeführt, aber es kann nicht verhindert werden, dass diese Fehler sich über mehrere Iterationsstufen in bestimmten Bereichen ungünstig aufsummieren.

Verfahren, die nur aufgrund der aktuellen Netzbeschaffenheit die nächste Aktion bestimmen und den vorherigen Iterationsprozess unberücksichtigt lassen, sind *transitiv*. Sie könnten unterbrochen und für das bis dahin vereinfachte Netz neu aufgesetzt werden und würden zum gleichen Ergebnis wie die Simplifizierung kommen, die ohne Unterbrechung erfolgt. Fehlermaße dieser Verfahren nennt man *inkrementell*. Hingegen ist unter anderem das implementierte Verfahren nicht transitiv, da das Fehlermaß ein "Gedächtnis" hat und die Fehler vorheriger Iterationsstufen geschickt speichert und bei aktuellen Bewertungen berücksichtigt. Es soll damit der Fehler einer Detailstufe zum Originalnetz bestimmt werden. Eine Unterbrechung und ein Neuaufsetzen der Simplifizierung mit dem bis dahin simplifizierten Netz als originales Ausgangsnetz ergibt somit ein anderes Ergebnis wie die Vereinfachung in einem Zuge. Die für gewöhnlich genaueren Fehlermaße dieser Klasse an Verfahren nennt man *total*. Dass aber auch inkrementelle Verfahren für moderate Vereinfachungen gute Ergebnisse liefern, ist dem Fehlerverlauf des inkrementellen Verfahrens der Dezimierung von Schroeder et al. in Diagramm 2.25 zu entnehmen.

In der Fachliteratur wird statt der Begriffe inkrementeller und totaler auch lokaler und globaler Fehler benutzt. Da dies zu Verwechslungen mit den exakten globalen Approximationsfehlern aus Abschnitt 2.1.2 führen könnte und die Bedeutung des Begriffs dann nur aus dem Sinnzusammenhang erschlossen werden könnte, wird auf die Benutzung der Begriffe lokaler und globaler Fehler in diesem Zusammenhang verzichtet.

Um sehr schnelle Verfahren zu entwickeln, werden auch schnell zu berechnende Maße wie Kantenlänge, Winkel zwischen angrenzenden Flächennormalen und lokale Oberflächenkrümmung herangezogen [30, 41, 42]. Da auch diese Fehlermaße das Originalnetz bei fortschreitender Vereinfachung außer Acht lassen und der Fehler keinen direkten Bezug zu den Knotenpositionen des Netzes hat, kann das Vereinfachungsergebnis erheblich vom Original abweichen. Solch einfache Heuristiken sollten also lediglich für ganz bestimmte Netze oder Netzteile herangezogen werden, die zum Beispiel eine gleichmäßig glatte Oberflächenbeschaffenheit haben und ein äquidistantes Gitter besitzen.

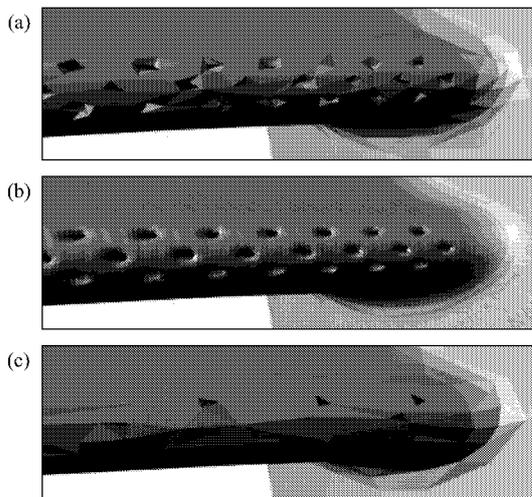


Abb. 2.24: Nahansicht eines Modells im Original in (b) vereinfacht mittels (a) eines rein geometrischen Fehlermaßes und (c) eines Bildfehler-basierten Ansatzes aus [43], Bild aus [43]

Die häufigste Anwendung der Vereinfachung ist die Beschleunigung der Visualisierung komplexer Modelle. Für das Ziel der Visualisierung ist aber nicht unbedingt die geometrische Genauigkeit am wichtigsten, die bei der überwiegenden Anzahl an Verfahren zur Fehlerbestimmung herangezogen wird und alle hier vorgestellten Ansätze prägt, sondern die visuelle Wahrnehmung. Es gibt aber erst wenige Verfahren, die das *Erscheinungsbild* als Maß für die Güte der Vereinfachung heranziehen, da die Bewertung der visuellen Unterschiede schwer ist: Lindstrom bewertet die Unterschiede erzeugter Bilder [43] und auch Cohen versucht durch die Vereinfachung das Erscheinungsbild so gering wie möglich zu verändern [44]. In Abbildung 2.24 ist gut zu erkennen, wie das *Bildfehler-basierte Fehlermaß* aus [43] die Dreiecke so positioniert, dass visuell wichtige Bereiche möglichst genau wiedergegeben werden. Eine weitere Schwierigkeit wird dabei sichtbar: das erzeugte Bild

hängt durch die Beleuchtung stark von der Betrachterposition ab, so dass entweder das Vereinfachungsverfahren mehrere Betrachterpositionen berücksichtigen oder sogar betrachterabhängige Datenstrukturen liefern muss.

Für Vereinfachungen, die nicht vordringlich die Darstellungsgeschwindigkeit erhöhen sollen und somit nicht im Bereich der Visualisierung eingesetzt werden, sondern zum Beispiel in der Finiten Elementen Analysis Berechnungen durch geringere Netzkomplexität beschleunigen sollen, können wiederum ganz andere Eigenschaften des Fehlermaßes wie die Erhaltung scharfer Kanten oder die Bewahrung der Topologie wichtig sein.

Die Vereinfachungsmethoden lassen sich aufgrund des Einflusses des Fehlermaßes auf das Abbruchkriterium einteilen. Hier kann zwischen Genauigkeit-basierten (fidelity based) und Anzahl-basierten (budget based) Methoden unterschieden werden. Die *Genauigkeit-basierten Verfahren* erlauben die Angabe von Fehlerschranken für den Simplifizierungsprozess und vereinfachen das Netz bis zu dieser festgelegten Schranke. *Anzahl-basierte Verfahren* hingegen werden durch die Angabe der Anzahl der gewünschten Knoten oder Dreiecke in der Simplifizierung begrenzt und versuchen dabei die Genauigkeit für die angegebene Zahl zu maximieren. Der Vorteil für die über einen maximalen Fehler zu beschränkenden Verfahren liegt in der vorhersagbaren Güte der Vereinfachung. Für die Möglichkeit der Angabe solch einer Fehlergrenze braucht das Verfahren einen aussagekräftigen Verfahrensfehler, mit dem die Methode arbeitet, welches das für die Implementation ausgewählte Quadrik-Fehlermaß nicht bietet. Dieses Fehlermaß, welches zu den Anzahl-basierten Verfahren gehört, hat hingegen durch die Möglichkeit der genauen Vorgabe der Anzahl der gewünschten Netzprimitive den Vorteil, die Netzkomplexität genau den Erfordernissen anzupassen. Dies ist für eine zeitkritische Visualisierung wichtig, die bei der Anwendung im Vordergrund steht und die Einstellung von Bildwiederholraten oder Ähnliches durch gezielte Erhöhung oder Reduzierung der Knoten- oder Dreieckszahl ermöglicht.

Eine letzte Klassifizierung der Fehlermaße soll aufgrund ihrer Art der Messung gemacht werden. Es wird zwischen Knoten-Knoten, Knoten-Ebene, Knoten-Oberfläche und Oberfläche-Oberfläche messenden Maßen unterschieden.

Die auf der Bestimmung des Fehlermaßes über den *Knoten-Knoten-Abstand* aufsetzenden Verfahren, zu denen das Maß von Rossignac und Borrel aus Abschnitt 2.4.1 gehört, arbeiten ohne topologische Beschränkungen in der Ein- und Ausgabe. Sie verhindern zudem, dass sich die Knotenposition stark ändert, besitzen aber eine hohe geometrische Oberflächenabweichung. Des Weiteren sind sie sehr robust, schnell und leicht zu implementieren. Es existiert für diese Verfahren eine ungenaue globale Fehlergrenze.

Die Fehlermaße, die den *Knoten-Ebenen-Abstand* zur Bewertung bemühen, wie es das implementierte Quadrik-Fehlermaß tut, führen in gewisser Weise für jeden Knoten seine mit ihm korrespondierenden Ebenen mit, die sich aus der Vereinigung der Ebenen vorhergehender Detailstufen ergeben. Dabei versuchen sie den Abstand der Knoten zu diesen Ebenen so gering wie möglich zu halten. Ihre Qualität ist gut, aber sie bieten keine globale Fehlergrenze für die Genauigkeit. Die Geschwindigkeit der auf den Maßen aufsetzenden Verfahren ist hoch, liegt aber unter der der Knoten-Knoten-Maße.

Verfahren, die auf dem *Knoten-Oberflächen-Abstand* aufsetzen, messen den Abstand einer Knotenmenge zur Oberfläche einer Simplifizierungsstufe wie es Hoppe in den Methoden aus 2.3.1 und 2.6.2 macht. Diese Fehlermaße approximieren den Abstand zweier Oberflächen sehr gut, bedürfen aber einer zeitintensiven Berechnung.

Die den *Oberflächen-Oberflächen-Abstand* bewertenden Fehlermaße und ihre Verfahren sind langsam, bieten dafür aber einen garantierten globalen Fehler über den maximalen Abstand beider Oberflächen. Die Klasse der Verfahren hat gemein, dass sie als Dreiecksnetz eine Mannigfaltigkeit voraussetzt und die Topologie während der Vereinfachung beibehält. Das auf der Einhüllung beruhende Verfahren von Cohen et al. aus Abschnitt 2.5.2 gehört zu dieser Gruppe von Verfahren.

2.7.4 Qualität und Geschwindigkeit

Zur abschließenden Bewertung der vorgestellten Verfahren soll die Qualität und Geschwindigkeit von konkreten Implementationen der vorgestellten Methoden verglichen werden. Die Implementationen stammen meist von den Autoren der Verfahren.

Um die *Qualität* der vereinfachten Netze zu bewerten, werden strengere Fehlermaße als die heuristischen gebraucht. Sie wurden in Abschnitt 2.1.2 eingeführt und messen globale geometrische Fehler unter anderem als durchschnittlichen oder maximalen Abstand.

Die Diagramme aus Abbildung 2.25 bieten für fünf vorgestellte Verfahren einen Qualitätsvergleich. Sie basieren auf Abbildungen von Cignoni et al. [45], deren Werte mittels des Programms Metro [10] des gleichen Autors bestimmt wurden.

Metro ist ein Programm zur objektiven Qualitätskontrolle von Simplifizierungsalgorithmen. Es bietet die Möglichkeit den Abstand zweier Netze in verschiedenen Varianten zu bestimmen, also insbesondere den Approximationsfehler einer Detailstufe zum Originalnetz auszurechnen. Darüber hinaus ist auch eine grafische Ausgabe möglich, die auf dem Netz anzeigt, wo und in welche Richtung Fehler aufgetreten sind.

Im Diagramm 2.25 (a) ist der globale maximale Fehler E_{max} und in (b) der globale durchschnittliche Fehler E_{sum} der einzelnen Verfahren aufgetragen. Die Fehlerskala ist logarithmisch und gibt den Fehler in Prozent zu der Länge der Raumdiagonalen des das Modell minimal umschließenden, achsenparallelen Quaders an. Dieser Quader wird *Umgebungsrechteck* (bounding box) genannt. Ebenso

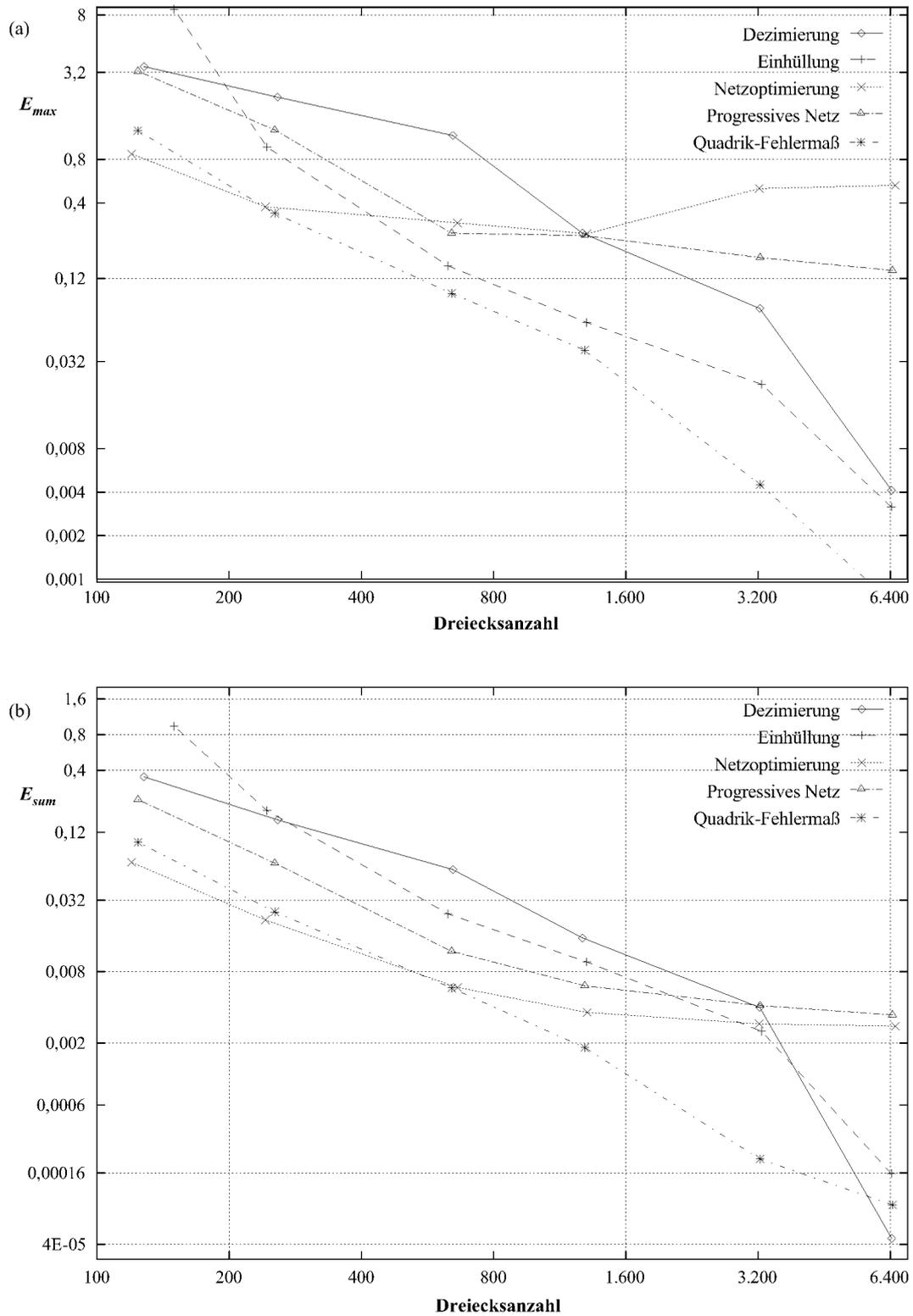


Abb. 2.25: Diagramme zum Qualitätsvergleich fünf vorgestellter Verfahren, die den globalen (a) maximalen und (b) durchschnittlichen Fehler in Abhängigkeit von der Dreiecksanzahl zeigen, basiert auf Diagrammen aus [45]

ist die Dreieckszahl auf einer logarithmischen Skala aufgetragen, in der sich an jeder Markierung die Netzkomplexität verdoppelt. Das Verhältnis der Verfahrensfehler zueinander hängt in gewissem Maße vom Modell ab, so dass die Diagramme nur eine Tendenz aufzeigen sollen.

Bei dem Modell handelt es sich um eine aus im Original 12.946 Dreiecken bestehende Ventilatorscheibe, die eher durch die konstruktive Herkunft für das Verfahren auf Basis des Quadrik-Fehlermaßes schlechter zu simplifizieren ist. Trotzdem zeigt es hier sehr gute Ergebnisse. Es ist zu sehen, dass die Netzoptimierung ihre rechenaufwendige Stärke erst bei hohen Vereinfachungsstufen ausspielen kann, hingegen nach nur wenigen Vereinfachungsschritten eher schlecht abschneidet, da bei der Energiebestimmung auch andere Netzparameter wie gleichmäßige Dreiecke mitbewertet werden. Das einhüllende Verfahren liegt etwas besser bei dem maximalen Fehler, da dieser im Vereinfachungsverfahren einfließt, wohingegen die anderen Verfahren durch die Berücksichtigung des durchschnittlichen Fehlers im Vereinfachungsverfahren dort besser abschneiden. Die Dezimierung liefert für wenig Vereinfachungsschritte gute Ergebnisse, kann diese aber nicht halten, da das Verfahren nur einen inkrementellen Fehler verfolgt und damit eine ungünstige Aufsummierung des totalen Fehlers nicht mitbekommt.

Die Verfahrensgeschwindigkeit kann der Tabelle 2.1 des nächsten Abschnitts entnommen werden, in deren letzter Spalte wenn möglich eine Zahl für die grobe Richtung der Geschwindigkeit in tausend reduzierte Dreiecke pro Sekunde steht. Die Werte sind den Angaben aus den ursprünglichen oder den darauf aufsetzenden Arbeiten entnommen und wegen der Verwendung unterschiedlicher Rechner und Daten in den Arbeiten nur ein grober Richtwert. Zudem geht die Dreieckszahl nicht bei allen Verfahren linear in die Berechnungszeit ein. Die Werte sind an den im Anhang A.1.5 spezifizierten Rechner angepasst. Für die Häufung ist kein Wert ermittelbar. Dieser dürfte aber noch über dem des Quadrik-Fehlermaßes liegen.

Das für die Implementation ausgewählte Verfahren auf Basis des Quadrik-Fehlermaßes zeichnet sich durch eine sehr hohe Geschwindigkeit aus. Mit dem Fehlerverlauf aus Abbildung 2.25 ist der sehr gute Kompromiss zwischen Geschwindigkeit und Netzgüte exemplarisch gezeigt. Um das gute Verhältnis von Qualität und Geschwindigkeit zu unterstreichen, ist in Diagramm 2.26 der globale durchschnittliche Fehler für eine Netzsimplifizierung der Ventilatorscheibe auf zehn Prozent auf die Verfahrensgeschwindigkeit in tausend reduzierte Dreiecke pro Sekunde aufgetragen. Die Achse, auf die die Geschwindigkeit aufgetragen ist, besitzt eine logarithmische Skala. Da beide zugrundeliegenden Werte

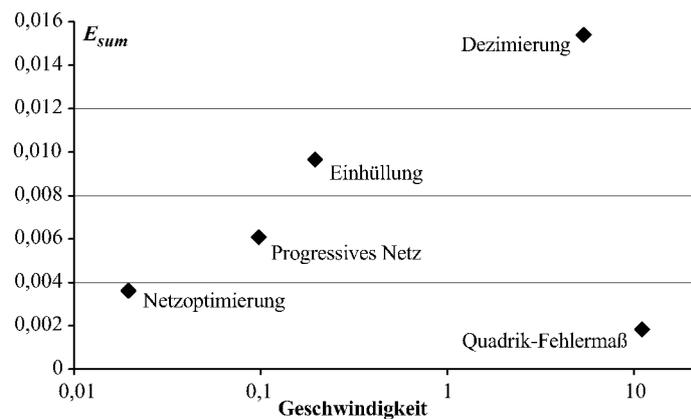


Abb. 2.26: Zusammenhang von Geschwindigkeit und Qualität für fünf vorgestellte Verfahren durch Auftragung des globalen durchschnittlichen Fehlers eines auf zehn Prozent reduzierten Netzes auf die Verfahrensgeschwindigkeit in tausend reduzierte Dreiecke pro Sekunde

für dieses Diagramm nur eine grobe Richtung angeben, hat dieses Diagramm ebenso nur beispielhaften Charakter. Es zeigt aber, dass nur das Quadrik-Fehlermaß der Aussage widerspricht, dass Qualität und Geschwindigkeit konkurrierende Gegensätze sind.

2.7.5 Zusammenfassung

Zum Abschluss des Verfahrenvergleichs sind hier die wesentlichen Unterschiede der Verfahren in einer Tabelle zusammengefasst.

Verfahren	Operation			kont. LOD	Topologie			Kno. fest	Eig. ber.	Fehlermaß				Gesch.
	Verf.	Dezi.	Vers.		erh.	mod.	tol.			Anz.	Gen.	total	begr.	
Verfein.	X			X	X						X	X	X	0,1
Häufung			X ¹			X	X				X ²	X ²	X ²	
Dezim.		X			X		X	X	X	X ³	X			5,5
Hülle		X			X			X	X		X	X	X	0,2
Optim.			X ⁴		X					X		X ⁵		0,02
PM			X	X	X				X	X		X ⁵		0,1
Quadrik			X	X		X ⁶	X			X		X		11,3

Tab. 2.1: Eigenschaften der Vereinfachungsverfahren

¹ im weiteren Sinne; die Anzahl der verschmelzenden Knoten schwankt

² der eigentliche Algorithmus hat keinen direkten Einfluss auf die Genauigkeit, die Gitterbreite ließe sich aber für einen global zu begrenzenden Approximationsfehler heranziehen

³ auch wenn in der eigentlichen Arbeit nicht darauf eingegangen wird, ist der Ansatz leicht dahingehend erweiterbar

⁴ andere Operationen werden auch durchgeführt, dienen aber nicht vordringlich der Netzvereinfachung

⁵ hierbei handelt es sich nicht um einen totalen geometrischen Fehler, sondern um eine Energie, die den Unterschied der Simplifizierung zum Originalnetz angibt

⁶ ist über einen Schwellwert einzuschränken

Die Verfahren sind in der Tabelle 2.1 in der Reihenfolge angegeben, wie sie auch in dieser Arbeit vorgestellt wurden. Die Tabellenspalten sollen nun kurz erläutert werden. Die dem Verfahrensnamen folgenden drei Spalten geben die Klasse der elementaren Vereinfachungsoperation an: Verfeinerung, Dezimierung oder Knotenverschmelzung. Die Knotenhäufung ist auch in die Klasse der knotenverschmelzenden Methoden aufgenommen worden, da, wenn auch unkontrollierter, das Zusammenziehen der Knoten die Vereinfachung bewirkt. Die vierte Spalte gibt Auskunft über die Möglichkeit einer einfachen Erzeugung kontinuierlicher Detailstufen zum Beispiel über die PM-Datenstruktur aus Abschnitt 2.6.2. Wenn die Möglichkeit in der Arbeit nicht angegeben ist, so sollte diese aber leicht dahingehend erweitert werden können.

Die folgenden drei Spalten teilen die Verfahren in topologieerhaltend, -modifizierend und -tolerant ein. Es schließt sich eine Spalte an, die Auskunft darüber gibt, ob das Verfahren die Knoten der Eingabe während des Vereinfachungsverfahrens an ihrer Position belässt und damit rein topologische Operationen einsetzt oder möglicherweise verschiebt. Eine Beibehaltung hat Vor- und Nachteile: als Vorteil ist zum Beispiel der geringere Speicherplatz insbesondere für kontinuierliche Datenstrukturen und als Nachteil die ungenauere Nachbildung des Originalnetzes und damit in dem größeren Approximationsfehler zu sehen. Die nächste Spalte enthält einen Eintrag, falls die Methode Eigenschaften des Netzes bei der Vereinfachung berücksichtigt.

Die Verfahren werden in den folgenden vier Spalten aufgrund von Eigenschaften des verwendeten Fehlermaßes klassifiziert. Es wird angegeben, ob das Fehlermaß benutzt wird, um direkt die Anzahl

der Netzprimitive der Simplifizierung zu steuern oder der Fehler als Grenze für die Genauigkeit zu verstehen ist. Außer dem Verfahren mittels Dezimierung, das einen inkrementellen Fehler zur Bewertung heranzieht, arbeiten alle mit einem totalen Fehler, bewerten also den Fehler des Originalnetzes zur aktuellen Vereinfachungsstufe und nicht nur zwei aufeinanderfolgende Detailstufen. Die Begrenzung des Fehlermaßes aus der vorletzten Spalte besagt, dass ein globaler Fehler vor der Simplifizierung vorgegeben werden kann, der vom Verfahren eingehalten wird.

Als letzte Spalte ist wenn möglich eine Zahl für die grobe Richtung der Geschwindigkeit des Vereinfachungsverfahrens in tausend reduzierte Dreiecke pro Sekunde eingetragen. Eine Kommentierung der Geschwindigkeitswerte ist im vorhergehenden Abschnitt zu finden.

2.8 Weitere Techniken

2.8.1 Geomorphs

Eine von Hoppe in [5] zuerst vorgeschlagene Technik, die das als Popping bezeichnete und oft als störend empfundene Umspringen von Detailstufen durch fließende Übergänge derselben vermeiden soll, wird Geomorph genannt. Insbesondere beim Detailwechsel der diskreten Stufen statischer Vereinfachungen ist der Übergang in eine feinere oder gröbere Modellrepräsentation deutlich zu erkennen. Als Bedingung für die Anwendbarkeit seiner Technik sind Detailstufen der Netze, bei denen das feinere Netz alle Knoten und Kanten des gröberen Netzes beinhaltet. Dies ist zum Beispiel bei den Progressiven Netzen, ebenfalls von Hoppe eingeführt (siehe Abschnitt 2.6.2), der Fall, so dass zwischen zwei verschiedenen Feinheitsstufen ein flüssiger Übergang definiert werden kann.

Sei $M_i^j(\alpha)$ der Übergang des Netzes M_i in das feinere Netz M_j , $0 \leq i < j \leq n$ mit dem Überblendparameter α , $0 \leq \alpha \leq 1$. $M_i^j(0)$ sieht dabei aus wie M_i und $M_i^j(1)$ wie M_j . M_i^j besitzt bereits alle Knoten von M_j , jedoch gilt für die von M_i zu M_j hinzugekommenen $j - i$ Knoten v_{t_l} , $l \in \{i + 1, \dots, j\}$, welche jeweils aus den Knoten v_{s_l} durch Knotenspaltung hervorgehen:

$$v_{t_l}^* = (1 - \alpha)v_{s_l} + \alpha v_{t_l}.$$

$v_{t_l}^*$ ist dabei der v_{t_l} ersetzende “geomorphe” Knoten, der sich bei stetiger Zunahme von α zwischen 0 und 1 von v_{s_l} bis v_{t_l} bewegt. Ist v_{s_l} auch noch nicht in M_i enthalten, so wird entsprechend $v_{s_l}^*$ eingesetzt usw.

Ist die Elementaroperation zur Netzvereinfachung nicht rein topologisch, würde zum Beispiel eine Kantenkontraktion einen beliebigen Verschmelzungspunkt \bar{v}_{s_l} erlauben, müssten auch die Knoten v_{s_l} angepasst werden:

$$\begin{aligned} v_{t_l}^* &= (1 - \alpha)\bar{v}_{s_l} + \alpha v_{t_l} \\ v_{s_l}^* &= (1 - \alpha)\bar{v}_{s_l} + \alpha v_{s_l}. \end{aligned}$$

Falls das Netz Eigenschaften besitzt, sollten auch diese sich fließend verändern, jedoch nicht zwingend linear wie die Punktkoordinaten. Normalen sollten auf der Einheitskugel bewegt werden und Farben möglicherweise in einem anderen als dem RGB-Farbraum.

2.8.2 Progressive Übertragung

Erzeugt der Vereinfachungsalgorithmus je Detailstufe nur geringe Änderungen und ist reversibel, so besteht die Möglichkeit der progressiven Übertragung (progressive transmission) der Netzdaten über ein Kommunikationsnetzwerk. Dies ist, wie es der Name schon vermuten lässt, für die Datenstruktur Progressive Mesh der Fall, die in Abschnitt 2.6.2 vorgestellt wurde. An dieser Datenstruktur soll auch das Verfahren erläutert werden.

Wenn aus den Original-Netzdaten das PM $(M_0, \{vsplit_0, \dots, vsplit_{n-1}\})$ erstellt wurde, sind die Originaldaten nicht mehr nötig, da sie komplett in PM enthalten sind.

Über das Kommunikationsnetz wird erst das kleine grobe Anfangsnetz M_0 übertragen, welches bereits darstellbar ist. Anschließend werden nacheinander $vsplit_0$ bis $vsplit_{n-1}$ übertragen und jeweils für die Anzeige genutzt, sobald sie eingetroffen sind. So wird während der Übertragung das Netz immer feiner. Bei schnellen Kommunikationsnetzen kann es sinnvoll sein, erst bei exponentiell angestiegener Knotenzahl ein Neuzeichnen zu veranlassen. Am Ende der Übertragung liegt auf der Empfängerseite das komplette Netz verlustfrei vor.

Auch ohne das Ziel der Netzvereinfachung zur Rechenzeit- und Speicherplatzeinsparung wäre diese progressive Darstellung eines Netzes wie zum Beispiel das progressive JPEG für Pixelbilder [46, 47] eine sinnvolle Anwendung der Simplifizierungsalgorithmen.

2.8.3 Betrachterabhängigkeit

Die Vereinfachung bzw. die Datenaufbereitung zur einfachen Simplifizierung wird in einem Vorverarbeitungsschritt gemacht und auf das gesamte Objekt angewandt. Bei der *betrachterabhängigen* (view dependent) Vereinfachung steht hingegen erst zur Laufzeit die Position und Blickrichtung fest und kann damit auch erst zur Laufzeit (online) durchgeführt werden.

Es wird zwischen *Hardware-* und *Software-basierten Verfahren* unterschieden.

Das so genannte *Auswählen* (culling) und *Abschneiden* (clipping) wird meist von der Grafikhardware durchgeführt. Hierbei unterscheidet man “view frustum culling”, “back face culling”, “occlusion culling” und “contribution culling”. Auch die Bezeichnung “hidden surface removal” ist für die Beseitigung nicht sichtbarer Objekte gängig.

Der für den Betrachter sichtbare Bereich wird in einer 3D-Szene durch ein so genanntes *Sichtvolumen* (view volume) begrenzt. Im Fall der Parallelprojektion ist das Sichtvolumen ein in Projektionsrichtung unendlich ausgedehnter Quader, im Fall der perspektivischen Projektion eine Pyramide. Beide Volumina werden aus praktischen Gesichtspunkten durch eine vordere und hintere Ebene (clipping plane) begrenzt und es entsteht ein endlicher Quader oder ein *Kegelstumpf* (view frustrum bzw. view frustum). Beim “view frustum culling” werden nun die Grafikprimitive auf ihre Lage zu diesem Quader bzw. Kegelstumpf hin untersucht. Liegt das Objekt außerhalb, wird es nicht weiter berücksichtigt, liegt es komplett innerhalb, so muß es weiter betrachtet werden. Ein Abschneiden an den Begrenzungsebenen ist bei Objekten die sowohl außerhalb wie innerhalb liegen durchzuführen.

Das Beseitigen der Rückseiten wird “back face culling” genannt. Rückseiten von undurchsichtigen, geschlossenen Körpern sind definitionsgemäß unsichtbar. Da sie etwa die Hälfte der in der Szene insgesamt vorkommenden Flächen ausmachen, werden sie von der weiteren Betrachtung ausgeschlossen. Eine Fläche ist mit der Rückseite dem Betrachter zugewandt, wenn der Winkel zwischen der Flächennormalen und der Betrachtungsrichtung größer als 90° ist [48, 49, 50, 51].

Gerade im Spielbereich wird zudem das “*occlusion culling*” eingesetzt, welches den Raum in Bereiche aufteilt, deren Sichtbarkeit entweder ein- oder ausgeschaltet ist. Es sollen also die wirklich sichtbaren Objekte gefunden werden, also die Objekte ausschließen, die zwar im Sichtbarkeitsbereich liegen und die Oberfläche zu einem gewandt haben, aber trotzdem durch ein anderes Objekt verdeckt werden. Ein Beispiel wäre ein Haus mit mehreren Zimmern, von denen man lediglich das Zimmer sehen kann, in dem man sich befindet oder durch eine offene Tür hineinschauen kann. Alle anderen Zimmer könnten bei der weiteren Betrachtung unberücksichtigt bleiben.

Die letzte Methode des “*contribution culling*” eliminiert die Objekte, deren Darstellung bei der Bildschirmprojektion irrelevant ist, weil die Objektausdehnung zu klein ist [52].

Neben diesen meist Hardware-basierten Lösungen gibt es auch Ansätze, die per Software den Betrachterstandpunkt in die simplifizierte Darstellung einfließen lassen (continuous multiresolution). Es kann trotz der bereits umfangreichen Hardware-basierten Verfahren sinnvoll sein, eine gewissen Sichtbarkeitsprüfung schon vor der Übergabe an die Grafikkarte durchzuführen, um die vom Hauptspeicher an den Speicher der Grafikkarte zu übertragende und dann darzustellende Datenmenge zu reduzieren.

Es ist immer dann sinnvoll, wenn das Programm-basierte Verfahren zeitlich weniger an Berechnung kostet als es bei der Darstellungsgeschwindigkeit einbringt. Um diesen Aspekt der softwareseitigen betrachterabhängigen Vereinfachung schnell umzusetzen, sind intelligente Datenstrukturen nötig [53].

Einige Aspekte der betrachterabhängigen Vereinfachung sind die genauere Darstellung der Silhouette oder der Bereiche, deren Flächennormalen einen großen Winkel zum Betrachterwinkel bilden. Auch ist eine genauere Darstellung von Bereichen nahe dem Betrachter oder in Blickrichtung des Betrachters sinnvoll. Weiter entfernte Gegenstände oder Objektteile werden durch die projektive Darstellung kleiner angezeigt und erzeugen somit durch eine Vereinfachung geringere Darstellungsfehler. Netzteile am Rand des Sichtfeldes werden weniger stark wahrgenommen als die Flächen, die genau in Blickrichtung des Betrachters liegen. [54, 55, 56, 57, 58].

Für die betrachterabhängige Vereinfachung werden oft andere oder zusätzliche Fehlermaße berücksichtigt, die versuchen die Objekttriangulierung dahingehend zu optimieren, dass ein festgelegter maximaler Darstellungsfehler mit möglichst wenig Dreiecken erzielt wird. Diese Maße werden auch *Bildfehlermaße* genannt.

Als erstes Beispiel ist in der nebenstehenden Abbildung 2.27 eine Aufsicht auf die Triangulierung eines Höhenfeldes zu sehen. Höhenfelder eignen sich deshalb gut für die betrachterabhängige Vereinfachung, da neben ihrer einfachen Struktur der Daten beim Überflug Teile des Objekts besonders nah und viele andere besonders weit vom Betrachter weg liegen. In der Implementation von Delonge [59] werden Höhenfelder betrachterabhängig vereinfacht, indem zum einen über eine Quadtree^G-Datenstruktur nur die Dreiecke an die Grafikkarte übergeben werden, die im Sichtfeld des Betrachters liegen. Das Sichtfeld ist durch zwei Geraden in der Zeichnung angedeutet. Zum anderen wird mit Hilfe einer Wavelet^G-basierten Methode ein Bildfehlermaß eingeführt, das die abgebildete feine Triangulierung nahe des Betrachters bewirkt und in der Ferne gröber wird. Eine wieder feiner werdende Unterteilung zur rechten oberen Ecke hin wird durch starke Höhenunterschiede hervorgerufen, die den Bildfehler negativ beeinflussen. Diese sichtabhängige Vereinfachung von Höhenfeldern,

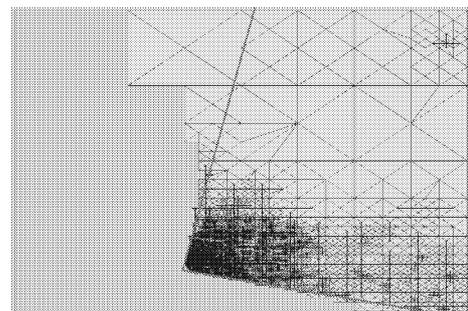


Abb. 2.27: Aufsicht auf die betrachterabhängige Triangulierung eines Höhenfeldes, Bild aus [59]

die zum Beispiel für Flugsimulatoren wichtig ist (terrain rendering), stellt den Großteil dieser Art von Simplifizierung dar, beschränkt sich aber auf eben diese Höhenfelder.

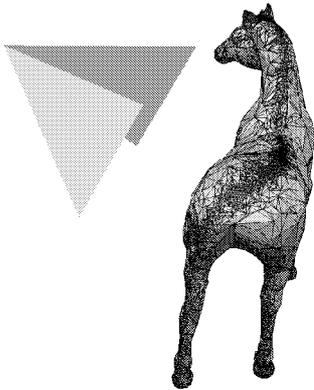


Abb. 2.28: Betrachterabhängige Verfeinerung der Triangulierung eines Pferdmodells im Bereich der Silhouette, Bild aus [60]

Es gibt auch Ansätze für beliebige Mannigfaltigkeiten wie im zweiten Beispiel in Abbildung 2.28 zu sehen. Hier wird von Pajarola [60] unter anderem eine schnelle Methode vorgestellt, um die für qualitativ hochwertige Bilder wichtige Silhouette betrachterabhängig zu verfeinern. Die Sichtpyramide wird durch die zwei Dreiecke neben dem Pferd angedeutet.

Die Schwierigkeit bei den Software-basierten Ansätzen ist die Geschwindigkeit. Oft scheitert die gute Idee daran, dass die Darstellungsgeschwindigkeit des Modells aufsetzend auf einer aufwendigen Datenstruktur durch die Betrachterabhängigkeit bei gleicher Dreiecksanzahl geringer wird. Dies ist insbesondere dann der Fall, wenn sich wie bei einem Virtual Reality-System^G die Betrachterposition ständig, wenn auch meist nur marginal, ändert. Die ständige Änderung der Betrachterposition rührt daher, dass durch positionsverfolgende (tracking) Geräte in einer Virtual Reality-Umgebung dem Rechner immer der aktuelle Standort des Betrachters mitgeteilt wird, um das Bild entsprechend der Bewegung anpassen zu lassen.

Ein Ansatz für eine betrachterabhängige Datenstruktur ist in Abschnitt 2.7.2 beschrieben.

2.8.4 Modellierung an Vereinfachungen

Werden die simplifizierten Netze in einer Umgebung eingesetzt, in der auch das Ändern der Netze möglich sein soll, ist es sinnvoll, dies auch für die verschiedenen Vereinfachungsstufen zuzulassen. Dadurch ist es elegant möglich, statt mühsam einzelne Knoten im feinen Originalnetz zu verschieben, ganze Bereiche seinen Wünschen anzupassen.

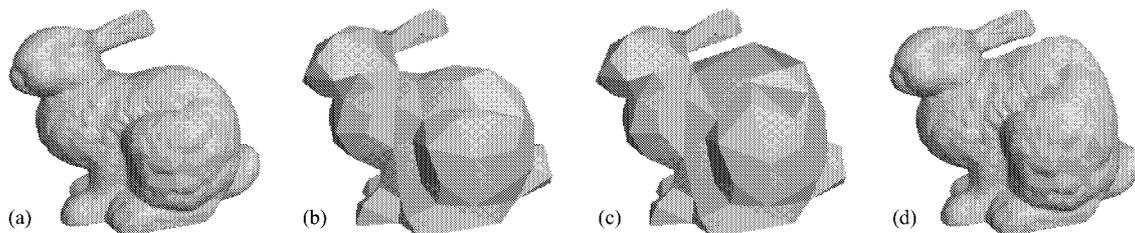


Abb. 2.29: Stufen der Modellierung an Vereinfachungen: (a) Originalnetz, (b) vereinfachtes Netz, (c) Netz an wenigen Knoten des Rückens modifiziert und (d) Auswirkung auf die originale Netzaufteilung, basiert auf Bildern aus [61]

Eine Änderung in einem groben Netz soll also an feinere Detailstufen weitergegeben werden. In Abbildung 2.29 (c) ist das in (b) vereinfachte Netz des Originalnetzes aus (a) an wenigen Knoten des Rückens modifiziert worden. In (d) ist die Auswirkung auf die originale Netzaufteilung zu erkennen.

Hierzu ist neben einer relativen Verarbeitung der Knotenpositionen auch eine dahinterliegende Datenstruktur nötig, die eine Abbildung der Gebiete einer jeden Detailstufe zueinander beinhaltet. Sehr gut eignet sich für die Modellierung an Vereinfachungen (multiresolution editing) das in Abschnitt 2.3.1 vorgestellte Vereinfachungsverfahren mit der dahinterliegenden Datenstruktur, da es topologieerhaltend ist und eine Zuordnung der Knoten in den einzelnen Vereinfachungsstufen über eine harmonische

Abbildung herstellt (subdivision connectivity). Aber auch die in Abschnitt 2.7.2 genannte Datenstruktur HDS bietet diese Abbildung der Knoten einzelner Detailstufen zueinander. Einige Ansätze sind in den Arbeiten [62, 54, 63] zu finden.

2.8.5 Kompression

Die in dieser Arbeit nicht berücksichtigte Kompression (compression) stellt ebenfalls eine Möglichkeit der Verringerung des Speicherplatzbedarfes eines Netzes dar. Bei der speichereffizienten Repräsentation wird jedoch nicht die Zahl der Komponenten wie Dreiecke, Knoten oder Kanten verringert, sondern unter anderem nach günstigen Datenstrukturen zur Speicherung der Struktur gesucht wie sie zum Beispiel der Dreiecksstreifen (triangle strip), der für jedes neue Dreieck nur einen neuen Knoten bedarf, oder eine Deltaspeicherung der Knoten darstellt, die nicht die absoluten Koordinaten der Punkte, sondern die besser zu komprimierenden Abstände zu Nachbarknoten als Knotenwerte verwendet [64, 65]. Von Kompression spricht man zudem, wenn zum Beispiel durch Quantisierung der Speicherplatz der eigentlichen Werte verringert wird, was meist aber nicht verlustfrei (lossless) sondern verlustbehaftet (lossy) geschieht [7].

Nicht unerwähnt soll auch die Geometrienachbildung durch *Texturen* bleiben, die in vielen Computerspielen eingesetzt wird, und vorab berechnete statische Bildausschnitte auf ein einfaches Netz wie eine Tapete aufklebt und somit den Eindruck einer detailreichen Szene erweckt. Auch *Beleuchtungsmethoden^G* unter zu Hilfenahme der *Gouraud- und Phong-Schattierung^G* gaukeln feinere Strukturen vor [48, 49, 50, 4].

2.9 Aktuelle Forschungsgebiete

Gerade in den letzten sieben Jahren ist viel Arbeit in die Untersuchung, Entwicklung und Implementation von Verfahren geflossen, die sich mit der Vereinfachung von Polygonnetzen befassen. Trotzdem gibt es noch eine ganze Reihe von offenen Fragen in diesem Bereich. Aktuelle Forschungsgebiete sind zum Beispiel die Entwicklung von Verfahren zur Datenorganisation im Hinblick auf die Cachelokalität, also die Erfolgchancen, die benötigten Daten im Cache aufzufinden [66].

Für eine schnelle Verarbeitung der Dreiecke in der Grafikkarte, ist die Verwendung von Dreiecksstreifen (triangle strips) oder -fächern (triangle fans) [49, 48] wünschenswert. Diese Datenstrukturen während der fortwährenden Netzveränderung zu pflegen, ist ebenfalls Inhalt laufender Forschungsarbeiten [67, 68, 69].

Aber die meiste zukünftige Arbeit wird nicht in die generelle Arbeit zur Netzvereinfachung fließen, sondern werden Entwürfe für spezielle Anwendungen wie Computerspiele, Vegetationsvisualisierung oder auch der Ausarbeitung, für die die hier vorgestellte Vereinfachung implementiert wurde, darstellen [70].

3

Vereinfachung mittels des Quadrik-Fehlermaßes

3.1 Verfahrensbeschreibung

Im vorherigen Kapitel sind einige der wichtigsten Simplifizierungsalgorithmen mit ihren Vor- und Nachteilen vorgestellt worden. Dazu gehörte auch das Verfahren von Garland und Heckbert auf Basis der Quadrik-Fehlermetrik [31] (quadric error metric, QEM), die in dieser Arbeit mit Quadrik-Fehlermaß bezeichnet wird und auf welches sich diese Arbeit im Kern bei der Simplifizierung stützt. Es zeichnet sich durch einen guten Kompromiss zwischen Geschwindigkeit und Qualität aus. In der dieser Arbeit zugrundeliegenden Anwendung vorkommende Netzgrößen von 100.000 Dreiecken sind auf einem gewöhnlichen 1800 MHz Laptop (Rechnerspezifikation siehe Anhang A.1.5) in 5 Sekunden vollständig simplifiziert. Auch Luebke schlägt in seinem anwendungsorientierten Vergleich von Simplifizierungsverfahren [71] für die dieser Arbeit zugrundeliegenden Daten die Vereinfachung auf Basis des Quadrik-Fehlermaßes vor. Er schreibt, dass die Methode sich besonders gut für die schnelle Reduzierung der Netzkomplexität von überabgetasteten Regionen, die leicht beim Marching-Cube-Algorithmus entstehen, eignet. Zudem verarbeitet das Verfahren für gewöhnlich organische Formen wissenschaftlicher Visualisierung, zu denen die Elektronendichtewolke der zugrundeliegenden Anwendung zählt, besser als konstruierte Objekte. Eine ausführliche Auswertung der Ergebnisse der Implementierung ist in Kapitel 4 zu finden. Der Algorithmus soll im Folgenden näher vorgestellt werden.

Die Simplifizierung geschieht über das Verschmelzen von zwei Knoten, die für gewöhnlich auf einer gemeinsamen dann verschwindenden Kante liegen, aber nicht liegen müssen (siehe Abschnitt 3.1.1). Die Topologie des Ausgangsnetzes kann dabei beliebig sein und wird auch nicht zwingend erhalten. Durch die beliebige Topologie ist dieses Verfahren nicht wie viele andere auf reine Mannigfaltigkeiten beschränkt. Das Verfahren geht sogar noch weiter und kann das Verschmelzen von voneinander getrennten Netzen zulassen. Diese Vorgehensweise ist zwar zum Beispiel bei einem Loch im Herzen kritisch, das für die Diagnoseerhalten bleiben sollte, bei diesem Verfahren aber bei fortschreitender Vereinfachung möglicherweise verschwindet. Für gewöhnlich ist der Ansatz aber sinnvoll, um bessere Approximationen eines Netzes mit möglichst wenig Dreiecken zu erzielen. Als extremes Beispiel

wäre dafür ein Schwamm mit vielen kleinen Poren zu nennen, dessen feine Struktur mit Topologieerhaltung kaum vereinfacht werden könnte. Für die visuelle Güte ist nicht die Topologie, sondern die geringe Differenz zum Original wichtig.

Topologiebewahrende Einschränkungen für den Algorithmus sind denkbar und könnten auch Mannigfaltigkeiten erhalten, sind aber außer den in Abschnitt 3.1.5 aufgeführten Begrenzungen der Vereinfachung durch Bestrafung von Kontraktionspaaren nicht implementiert worden.

Welche Knoten sich verbinden können (siehe Abschnitt 3.1.2) und wo der Verschmelzungspunkt liegt (siehe Abschnitt 3.1.4) wird über ein Quadrik-Fehlermaß bestimmt. Dieses gibt auch die Reihenfolge der Verschmelzung vor (siehe Abschnitt 3.1.3). Um die gewünschte Netzdichte schnell mit Hilfe der berechneten Simplifizierung einstellen zu können (siehe Abschnitt 3.1.7), wird sie in einer speziellen Datenstruktur gespeichert (siehe Abschnitt 3.1.6).

Netzattribute sind auch mit Hilfe des Quadrik-Fehlermaßes behandelbar, aber in dieser Arbeit nicht erwähnt, weil die zugrundeliegende Anwendung keine Attribute enthält. Wie diese elegant in den Entscheidungsprozess über die Quadriken mit einfließen können, wird in [39, 40] beschrieben.

3.1.1 Fortschreitende Knotenverschmelzung

Der Vereinfachungsalgorithmus beruht ausgehend vom vollständigen Originalnetz auf der iterativen Verschmelzung von Knotenpaaren (vertex pair contraction). Hierbei werden zwei Knoten zu einem zusammengeführt (Geometrieänderung). Die zuvor zwischen diesen Knoten möglicherweise bestandene Kante verschwindet mit ihren meist zwei Dreiecken. Die Knotenzugehörigkeit zu den Dreiecken und Kanten wird entsprechend angepasst (Topologieänderung). An Rändern kann eine Kante auch nur einem Dreieck angehören, in entarteten Bereichen ist eine Kante auch manchmal in mehr als zwei Dreiecken enthalten. Diese Kontraktion wird soweit durchgeführt bis die gewünschte Vereinfachungsstufe erreicht ist oder keine Knotenpaare zur Verschmelzung übrig sind.

Bei der Verschmelzung bleibt unter Umständen die Topologie nicht erhalten. Die Netzveränderung wirkt sich nur sehr lokal aus und lässt den Rest des Netzes unberührt, was für die Berechnungsgeschwindigkeit von großem Vorteil ist. Zudem hat die Lokalität der Operation auch den Vorteil, dass jeder Vereinfachungsschritt nur geringe visuelle Änderungen bewirkt und eine kontinuierliche Speicherung der Detailstufen erlaubt (siehe Abschnitt 3.1.6).

Die Knotenpaare müssen für eine Kontraktion nicht auf einer gemeinsamen Kante liegen. Die zwei generellen Arten der Verschmelzung sind in Abbildung 3.1 zu sehen. Variante (a) ist die auch in vielen

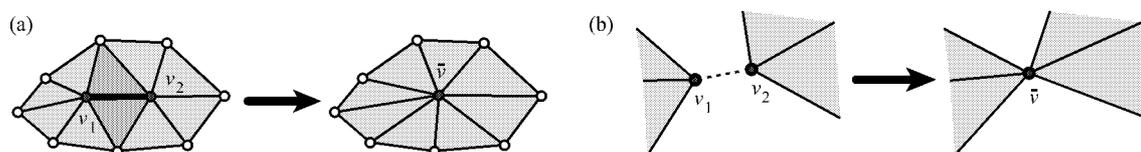


Abb. 3.1: (a) Kantenkontraktion: die dicke Kante kollabiert auf einen Knoten und die dunkelgrau hinterlegten Dreiecke verschwinden aus dem Netz, (b) kantenungebundene Knotenverschmelzung: die gestrichelte Linie zeigt die sich verbindenden Knoten an, die zwei bisher getrennten Netzteile verschmelzen; basiert auf Bild aus [31]

anderen Vereinfachungsalgorithmen verwandte Kantenkontraktion (edge collapse, edge contraction), bei der die dick markierte Kante zusammenfällt und ein Aufeinanderfallen der Knoten v_1 und v_2 auf den Knoten \bar{v} bewirkt. Dabei verschwinden die dunkel gefüllten Dreiecke.

In (b) deutet die gestrichelte Linie an, dass die Knoten v_1 und v_2 sich im Knoten \bar{v} vereinigen, obwohl sie keine gemeinsame Kante besitzen. Das Knotenpaar (v_1, v_2) wird als imaginäre Kante bezeichnet. Der Vorgang wird kantenungebundene Knotenzusammenführung (non-edge contraction) genannt und führt dazu, dass zuvor unverbundene Netzteile verschmelzen (aggregation). Hierbei können sich sogar komplett getrennte Netze zu einem Netz verbinden. Ist eines der Teilnetze, denen v_1 oder v_2 angehört, eine Mannigfaltigkeit, dann wird sie durch diese Zusammenführung aufgehoben, da im kritischen Punkt \bar{v} keine zweidimensionale Mannigfaltigkeit mehr vorliegt.

Die in beiden Kontraktionsarten hell hinterlegten Dreiecke sind die Dreiecke, die sich lokal verändern; andere Dreiecke im Gesamtnetz bleiben unberührt. Zusammen bilden sie die in diesem Algorithmus verwandten Knotenverschmelzungsarten. Welche Knoten als Kandidaten zur Verschmelzung herangezogen werden, wird im folgendem Abschnitt beschrieben.

Die Vorgehensweise der Simplifizierung funktioniert wie die meisten Vereinfachungsalgorithmen wie ein simpler Greedy-Algorithmus (Beschreibung siehe Abschnitt 2.2). Die Menge besteht hier aus den Knotenverschmelzungskandidaten. Die Bewertung erfolgt über das Quadrik-Fehlermaß und bestimmt, welche Verschmelzung zuerst durchgeführt wird. Dieses Knotenpaar wird abgearbeitet und verschwindet danach aus der Menge der Kandidaten. Die erste Verfahrensskizze sieht also wie folgt aus:

1. Bestimme die Menge der Knotenverschmelzungskandidaten (siehe Abschnitt 3.1.2)
2. Weise jedem Kandidaten seine Kosten zu, die seine Verschmelzung verursacht (siehe Abschnitt 3.1.3)
3. Wiederhole folgende Punkte, bis keine Kandidaten mehr vorliegen oder die gewünschte Approximation erreicht ist:
 - (a) Eliminiere das Knotenpaar mit den geringsten Kosten
 - (b) Führe die Verschmelzung durch
 - (c) Aktualisiere die Kosten aller noch bestehenden Verschmelzungskandidaten, die einen Knoten des aktuellen Knotenpaares beinhalten

Die vielen Simplifizierungsverfahren diesen Typs unterscheiden sich hauptsächlich in der Berechnung der Kosten. Es kann nicht davon ausgegangen werden, dass dieser simple Greedy-Algorithmus die optimale Lösung findet. Das Auffinden einer optimalen Approximation ist NP-hart^G, weshalb bis heute kein optimales Simplifizierungsverfahren existiert [72, 73]. Eine gründlichere Optimumsuche könnte zwar bessere Vereinfachungen finden, aber das Ergebnis bliebe suboptimal.

Aus der Knotenverschmelzung resultiert eine natürliche Abhängigkeit der Knoten aufgrund der Simplifizierungsschritte des Originalnetzes zu den Netzen der fortschreitenden Vereinfachung. Die Abhängigkeit kann zum Beispiel für Geomorphs genutzt werden (siehe Abschnitt 2.8.1) oder zur Erzeugung einer natürlichen hierarchischen Repräsentation der Vereinfachungsschritte herangezogen werden (siehe Abschnitt 2.7.2).

Die einzige elementare Operation ist die Verschmelzung von zwei Knoten. Es sind aber noch andere Operationen denkbar, die den sonstigen Algorithmus weitestgehend unberührt lassen. Statt der zu kontrahierenden Kante könnte als Kontraktionsprimitiv ein Dreieck herangezogen werden, welches

auf einen Knoten zusammenfällt und dabei für gewöhnlich zwei Knoten und vier Dreiecke eliminiert. Es ist zu erwarten, dass diese Art der Verschmelzung schneller, aber auch ungenauer ist, da pro Iterationsschritt doppelt so viele Elemente verschwinden, aber die Operation größer ist.

Sogar eine beliebige Anzahl an Knoten kann mittels des Quadrik-Fehlermaßes bewertet werden. Es ließe sich zum Beispiel die Knotenhäufung aus Abschnitt 2.4 dahingehend anpassen, dass die Bewertung und sogar die im Folgenden beschriebene Wahl des Knotenverschmelzungskandidaten für die einzelne Zelle über das Quadrik-Fehlermaß abgewickelt wird, da die Additivität der Quadriken (siehe Abschnitt 3.1.3) ausgenutzt werden kann: $Q = \sum_i Q_i$ [33]. Die Knotenhäufung kann somit als die generellste Art der Knotenverschmelzung mit schwankenden Knotenzahlen aufgefasst werden.

Bei der Implementation wird nur die Kollabierung von Kanten angeboten, da sie die qualitativ besten Ergebnisse liefert und die in der Anwendung interessanten Problemgrößen schnell genug simplifiziert.

3.1.2 Wahl der Knotenverschmelzungskandidaten

Der initiale Schritt des Algorithmus ist die Bestimmung der Knotenverschmelzungskandidaten. Diese Paarbestimmung wird nur einmal zu Beginn durchgeführt. Die Entscheidung, während der Vereinfachung keine weiteren Kandidaten neu aufzunehmen, basiert auf der Annahme, dass in einer guten Approximation ein Knoten sich nicht weit von seiner ursprünglichen Position wegbewegt. Eine Vererbung von Kontraktionspartnern findet aber während der iterativen Simplifizierung statt.

Die natürlichen Kandidaten sind die Knoten an gemeinsamen Kanten des Ausgangsnetzes. Für ein zusammenhängendes Netz mag das auch meist ausreichen, aber insbesondere für Netze, die aus mehreren voneinander getrennten Teilnetzen bestehen, erhält man eine bessere Approximation, wenn auch kantenfremde Knotenpaare zur Verschmelzung herangezogen werden.

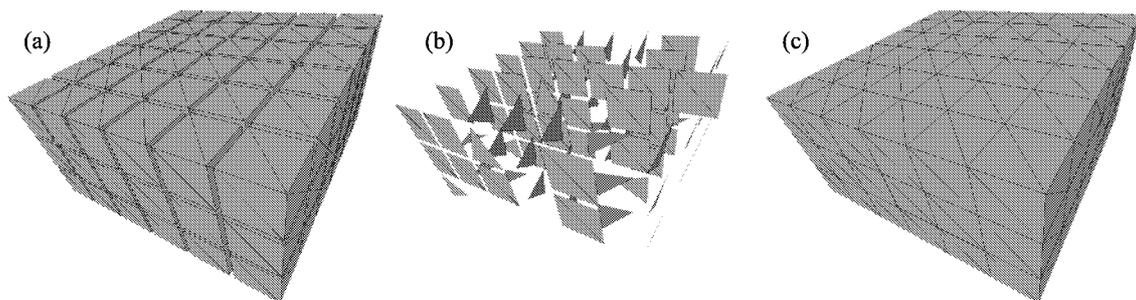


Abb. 3.2: (a) Originalmodell dicht beieinanderliegender Würfel mit 864 Knoten, 1944 Kanten und einer Ausdehnung von 2, (b) Vereinfachung auf 350 Knoten ohne Schwellwert, (c) Vereinfachung auf 200 Knoten mit Schwellwert 0,1 und daraus resultierenden 1988 zusätzlichen Knotenpaaren

Abbildung 3.2 zeigt in (a) ein Originalmodell 108 dicht beieinanderliegender Würfel. Wird auf die Möglichkeit verzichtet, nicht auf einer Kante liegende Knoten zu verbinden, dann sähe die Vereinfachung von 864 auf 350 entsprechend (b) aus. Jedes Teilnetz musste für sich vereinfacht werden. Netze mit wenig Dreiecken wie hier degenerieren dann schnell, bis sie ganz verschwinden.

Das Modell wird in (c) viel besser wiedergegeben, obwohl sogar nur 200 Knoten zur Darstellung ausreichen. Hierfür sind die knapp 2000 zusätzlichen Verschmelzungspaare¹ verantwortlich, die die dicht beieinanderliegenden Würfecken verschmelzen lassen.

¹ Genauer sind es 48 Knotenhäufungspunkte an den Außenkanten ($\cdot 1$), 90 Knotenhäufungspunkte auf den Außenflächen ($\cdot 6$) und 50 im Inneren des Modells ($\cdot 28$), was zusammen 1988 zusätzliche Knotenpaare ergibt.

Es kann also sinnvoll sein, kantenungebundene Knotenpaare zur Vereinfachung heranzuziehen. Statt komplexe Auswahlkriterien über Klasseneinteilungen, Octrees^G oder k -nächsten Nachbarn [27] wurde in dieser Arbeit wie bei Garland und Heckbert [31] folgende einfache Bedingung an ein Knotenpaar (v_i, v_j) gestellt, damit es in die Verschmelzungsmenge aufgenommen wird:

1. (v_i, v_j) ist eine Kante des Netzes oder
2. $\|v_i - v_j\|_2 < r$ und (v_i, v_j) ist keine Kante des Netzes.

r ist dabei ein Schwellwert (threshold), der die maximale Entfernung der Knotenpaarkandidaten angibt. Knoten, die jeweils innerhalb einer Kugel mit Radius r um einen anderen Knoten liegen und zusammen mit diesem keine Kante des Netzes bilden, werden somit durch Bedingung (2) zusätzlich zu den Kanten des Netzes als Kandidaten in die Verschmelzungsmenge aufgenommen. Die weitere Behandlung insbesondere der Kostenberechnung ist für beide Arten von Knotenpaaren identisch.

Die doppelte Anzahl an Knotenpaaren wie in Abbildung 3.2 (c) bedarf natürlich einer längeren Berechnung, zum einen zur Bestimmung und zum anderen zur Abarbeitung während der Simplifizierung. Die Bestimmung aller Knoten in einer durch den Schwellwert vorgegebenen Umgebung für jeden Knoten des Netzes (all ϵ -closeness) hat eine Zeitkomplexität von $O(n \log n + kn)$, wobei k die durchschnittliche Anzahl von Knoten in der untersuchten Umgebung um einen jeden Knoten ist. Daraus folgt für kleine Schwellwerte eine Zeitkomplexität von $O(n \log n)$, für große von $O(n^2)$ [6].

In der dieser Arbeit zugrundeliegenden Implementation wurde für große r ein direkter Knotenvergleich implementiert. Für kleine r wird eine Datenstruktur aufgebaut, um Nachbarknoten zu finden, in der alle Knoten nach ihren Koordinaten sortiert aufgenommen werden. Eine adaptive Wahl von r wird in [74] vorgestellt.

Die Verschmelzungskandidaten müssen während der iterativen Simplifizierung wie auch die Knoten und Dreiecke des Netzes auf dem Laufenden gehalten werden. Zieht sich das Paar (v_i, v_j) auf \bar{v} zusammen, so werden in allen verbleibenden Kandidaten v_i und v_j durch \bar{v} ersetzt, also Knotenpaarkandidaten, die v_i oder v_j enthalten, dem neuen Knoten \bar{v} gutgeschrieben. Daraus resultierende doppelte Einträge bei Knotenpaaren auf Kanten müssen eliminiert werden.

Für $r = 0$ werden über die obige Position (2) keine Kandidaten hinzugewonnen und das Verschmelzen von unverbundenen Netzteilen wird damit unterbunden. Das Verfahren verhält sich dann, als ob nur echte Netzkannten zur Simplifizierung herangezogen werden.

Andere Auswahlverfahren für die kantenungebundenen Kandidaten passen sich zwar besser an die Netzkomplexität an, was hier von Hand geschehen muss, lassen sich aber nicht so gut regeln, wenn es um die Überwindung bestimmter Lücken geht.

Für das Beispiel in Abbildung 3.3 sind in Abschnitt 4.4.3 Fehlerbetrachtungen in Abhängigkeit des Schwellwertes durchgeführt worden. Hierbei zeigt sich, dass die Hinzunahme von kantenfremden Paaren zwar eine Qualitätssteigerung der Simplifizierung bewirken

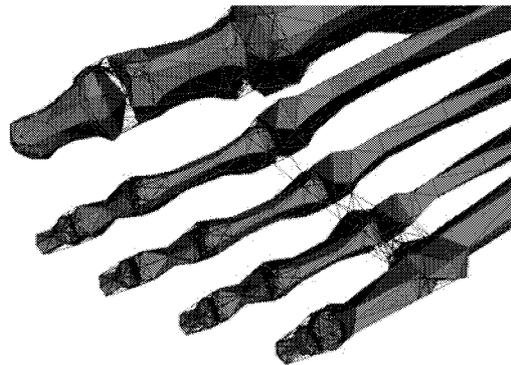


Abb. 3.3: Knotenpaarkandidaten bei echt positivem Schwellwert, kantengebundene und kantenungebundene Knotenpaare sind verschiedenfarbig gekennzeichnet, Bild aus [31]

kann, der Schwellwert aber mit Bedacht gewählt werden muss, da diese Art von Verschmelzungen nicht so robust gegen Artefakte ist. Zudem steigt die Paaranzahl schnell und bedarf somit viel Rechenzeit.

Die Anzahl der Knotenpaare zu einem Knoten wächst bei Erhöhung des Schwellwertes wie der Flächeninhalt einer Kreisscheibe quadratisch, da die Dreiecksflächen vornehmlich Oberflächen darstellen. Ein Beispiel in Abschnitt C.4 soll dies verdeutlichen. Bei stark zerklüfteten Oberflächen oder Netzen wie in Abbildung 3.2 kann die Anzahl auch kubisch entsprechend eines Kugelvolumens ansteigen.

Die maximale Gesamtzahl aller Knotenpaare liegt aber immer nur in $O(n^2)$, da n Knoten nur genau

$$(n-1) + (n-2) + \dots + 1 = \frac{n^2 - n}{2}$$

Verbindungen eingehen können.

3.1.3 Kostenbestimmung der Kontraktionen

Nun stellt sich noch die Frage, welche Kante im Greedy-Verfahren in jedem Iterationsschritt als bestbewertetes Element angesehen wird. Es müssen dafür die Kosten einer jeden erlaubten Kontraktion berechnet werden. Die Kosten sollen widerspiegeln, wie stark sich die jeweilige Knotenpaarverschmelzung auf die Oberfläche auswirkt. Wie in Abschnitt 2.7.3 und 2.7.4 beschrieben, ist bei Außerachtlassung von Netzeigenschaften und betrachterunabhängigen Vorgehensweise der geometrische Fehler ein gutes Maß zur Qualitätskontrolle.

In der dieser Arbeit zugrundeliegenden Anwendung hat man sich wie schon zuvor erwähnt für das Fehlermaß von Garland und Heckbert entschieden. Dieses Maß vereint die Eigenschaften schnell in der Berechnung zu sein, qualitative hochwertige Vereinfachungen zu erzeugen und jede auf Dreiecken aufgebaute Netztopologie zuzulassen. Obwohl dieses Fehlermaß schnell zu berechnen ist, liefert es ein gutes Maß für den geometrischen Fehler einer jeden Simplifizierungsstufe zum Originalnetz und nicht nur zum Netz aus der vorangegangenen Iterationsstufe.

Ebenenabstand als Fehlermaß

Die Idee hinter dem favorisierten Fehlermaß ist, den geometrischen Fehler durch den Abstand der vereinfachten Netzknoten zu den korrespondierenden Dreiecksebenen des Originalnetzes auszudrücken. Eine Dreiecksebene ist dabei die Ebene, die durch alle drei Punkte des Dreiecks verläuft. Korrespondierende Ebenen meint all diejenigen, die vom Originalnetz ausgehend während der Vereinfachung an einem der Verschmelzungskandidaten angelegen haben.

Dieses Konzept geht auf Ronfard und Rossignac [16] zurück, die aber den Maximalabstand betrachten. In dieser Arbeit wird hingegen die Summe der Quadratabstände als Maß genommen. Der Quadratabstand wird statt des einfachen absoluten Abstandes genommen, da dies bei der weiteren Berechnung eine kompakte Zusammenfassung der Summe erlaubt und dadurch erst die Effizienz des Algorithmus ermöglicht. Die bei der verwandten Euklidischen Metrik angewandte Wurzel über die Gesamtsumme wird bei der Bewertung weggelassen, da sie eine kompakte Zusammenfassung verhindert, Rechenzeit kostet und durch ihre streng wachsende Monotonie die Ordnung der Bewertungsfunktion nicht verändert. Insbesondere ist die Stelle, an der das Maß das Minimum annimmt, mit und ohne

Wurzel identisch. Der Fehler hat damit nicht so sehr eine weiter zu verwendende Aussagekraft als absolute Zahl, als vielmehr den Zweck einer relativen Einordnung möglicher Vereinfachungsschritte.

Das Verhalten gebräuchlicher Maße ist in Abschnitt 2.1.2 beschrieben, wo auch bereits das lokale Quadratsummen-Maß $E_{avg}^l(v, E)$ eingeführt wurde, welches dem hier nochmals vorgestellten Quadrik-Fehlermaß $Q(v)$ entspricht:

$$Q(v) = E_{avg}^l(v, E).$$

Die Umbenennung erfolgte, um das Fehlermaß kompakter schreiben zu können. Ein Vergleich mit anderen Maßen findet nicht mehr statt und die Menge der Ebenen E ist immer die Menge der korrespondierenden Ebenen.

Sei $v = (x, y, z)^t$ ein Knoten und die korrespondierenden k Ebenen gegeben durch die Hessesche Normalform $E_i : n_i^t w + d_i = 0$, $w \in \mathbb{R}^3$, $i \in \mathbb{N}_k$ mit $n_i = (a_i, b_i, c_i)^t$ als normierte Normalenvektoren und $|d_i|$ als Abstände der Ebenen zum Nullpunkt, dann ist der quadratische Abstand $D_i^2(v)$ des Punktes v von den Ebenen E_i

$$D_i^2(v) = (n_i^t v + d_i)^2 = (a_i x + b_i y + c_i z + d_i)^2$$

und die Summe der Quadratabstände zu allen korrespondierenden Ebenen

$$Q(v) = \sum_{i=1}^k D_i^2(v) = \sum_{i=1}^k (n_i^t v + d_i)^2.$$

Abbildung 3.4 zeigt im Zweidimensionalen exemplarisch das vorhin beschriebene Fehlermaß. Alle Geraden des abgebildeten dunklen Originalnetzes sind die korrespondierenden Geraden zu \bar{v} .

\bar{v} liegt an der Stelle, an der $Q(v)$ minimal ist. Solange die Geraden sich nicht in einem Punkt schneiden ist $Q(v) > 0$, ansonsten wäre der Schnittpunkt die Stelle, an der $Q(v)$ minimal 0 ist. Die gestrichelten elliptischen Linien sind Isolinien von $Q(v) = \epsilon$ für verschiedene Werte von ϵ . \bar{v} liegt dabei im Zentrum dieser konzentrischen Ellipsen. Im Dreidimensionalen werden aus den elliptischen Isolinien Isoflächen in Form von Ellipsoiden. Ganz allgemein gehören die Isoflächen, die $Q(v) = \epsilon$ erzeugen, in die Klasse der Quadriken. Dies ist auch der Grund, warum das Fehlermaß als Quadrik-Fehlermaß bezeichnet wird. In Abschnitt 3.2.5 wird gezeigt, dass es sich bei den Quadriken normalerweise um Ellipsoide handelt, die in Ausnahmefällen zu Zylindern oder Ebenenpaaren degenerieren können. Diese Degeneration existiert auch im Zweidimensionalen, wenn alle Geraden zueinander parallel sind. Dann werden aus den Ellipsen Geradenpaare und das Minimum liegt auf einer Geraden in der Mitte dieser Geradenpaare.

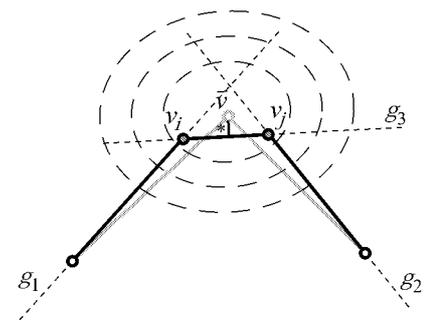


Abb. 3.4: Kostenbestimmung über den Quadratabstand zu den Geraden

Die Isolinien bzw. Isoflächen bleiben auch bei mehreren nicht parallelen Geraden Ellipsen bzw. bei Ebenen Ellipsoide. Warum das so ist, sieht man an der Möglichkeit der Zusammenfassung der Qua-

dratsummen.

$$\begin{aligned}
D_i^2(v) &= (n_i^t v + d_i)^2 \\
&= (n_i^t v + d_i)(n_i^t v + d_i) \\
&= (n_i^t v)(n_i^t v) + 2(n_i^t v)d_i + d_i^2 \\
&= (v^t n_i)(n_i^t v) + 2d_i(n_i^t v) + d_i^2 && \text{(Kommutativität des Skalarproduktes)} \\
&= v^t(n_i n_i^t)v + 2(d_i n_i)^t v + d_i^2 && \text{(Assoziativität der Matrixmultiplikation)} \\
&= v^t \underbrace{\begin{pmatrix} a_i^2 & a_i b_i & a_i c_i \\ a_i b_i & b_i^2 & b_i c_i \\ a_i c_i & b_i c_i & c_i^2 \end{pmatrix}}_{=: A_i} v + 2 \underbrace{\begin{pmatrix} a_i d_i \\ b_i d_i \\ c_i d_i \end{pmatrix}}_{=: B_i^t} v + \underbrace{d_i^2}_{=: C_i} \\
&= v^t A_i v + 2B_i^t v + C_i.
\end{aligned} \tag{3.1}$$

Die Quadratsumme lässt sich damit folgendermaßen schreiben:

$$\begin{aligned}
Q(v) &= \sum_{i=1}^k (v^t A_i v + 2B_i^t v + C_i) \\
&= v^t \left(\sum_{i=1}^k A_i \right) v + 2 \left(\sum_{i=1}^k B_i \right)^t v + \sum_{i=1}^k C_i && \text{(Distributivität der Matrixmultiplikation)} \\
&= v^t \underbrace{\left(\sum_{i=1}^k \begin{pmatrix} a_i^2 & a_i b_i & a_i c_i \\ a_i b_i & b_i^2 & b_i c_i \\ a_i c_i & b_i c_i & c_i^2 \end{pmatrix} \right)}_{=: A} v + 2 \underbrace{\left(\sum_{i=1}^k \begin{pmatrix} a_i d_i \\ b_i d_i \\ c_i d_i \end{pmatrix} \right)^t}_{=: B^t} v + \underbrace{\sum_{i=1}^k d_i^2}_{=: C} \\
&= v^t A v + 2B^t v + C.
\end{aligned} \tag{3.2}$$

Die Abbildung hat damit für eine beliebige Anzahl von Geraden im Zweidimensionalen bzw. Ebenen im Dreidimensionalen die gleiche Form, also für gewöhnlich als Isolinien Ellipsen bzw. als Isoflächen Ellipsoide. Da die Matrix A symmetrisch ist, genügen zur Beschreibung der Quadratsumme 10 Parameter, nämlich jeweils die Summe aller a_i^2 , b_i^2 , c_i^2 , d_i^2 , $a_i b_i$, $a_i c_i$, $a_i d_i$, $b_i c_i$, $b_i d_i$ und $c_i d_i$. Der Vorteil dieser Zusammenfassung der Summen ist offensichtlich: nur 10 Parameter beschreiben beliebig viele Ebenenabstände und bieten somit eine sehr effiziente Speicherung des Fehlers.

Eine kompaktere Schreibweise ist über homogene Koordinaten möglich. Dabei wird der zu v korrespondierende homogene Knoten durch $\tilde{v} = (x, y, z, 1)^t$ und die Ebenen mit $\tilde{n}_i = (a_i, b_i, c_i, d_i)^t$ durch $\tilde{n}_i^t \tilde{v} = 0$ beschrieben. Diese Schreibweise ähnelt den erweiterten Vektoren aus [75]. Die Qua-

dratabstände lauten dann folgendermaßen:

$$\begin{aligned}
 D_i^2(\tilde{v}) &= (\tilde{n}_i^t \tilde{v})^2 \\
 &= \tilde{v}^t \tilde{n}_i \tilde{n}_i^t \tilde{v} \\
 &= \tilde{v}^t \underbrace{\begin{pmatrix} a_i^2 & a_i b_i & a_i c_i & a_i d_i \\ a_i b_i & b_i^2 & b_i c_i & b_i d_i \\ a_i c_i & b_i c_i & c_i^2 & c_i d_i \\ a_i d_i & b_i d_i & c_i d_i & d_i^2 \end{pmatrix}}_{=: Q^f} \tilde{v} \\
 &= \tilde{v}^t Q^f \tilde{v}
 \end{aligned} \tag{3.3}$$

und die Quadratsumme

$$\begin{aligned}
 Q(\tilde{v}) &= \tilde{v}^t \left(\sum_{i=1}^k \underbrace{\begin{pmatrix} a_i^2 & a_i b_i & a_i c_i & a_i d_i \\ a_i b_i & b_i^2 & b_i c_i & b_i d_i \\ a_i c_i & b_i c_i & c_i^2 & c_i d_i \\ a_i d_i & b_i d_i & c_i d_i & d_i^2 \end{pmatrix}}_{=: Q} \right) \tilde{v} \\
 &= \tilde{v}^t Q \tilde{v}.
 \end{aligned} \tag{3.4}$$

Q wird aufgrund der Isoflächen von $Q(v) = \epsilon$ im Folgenden wie die Flächen selbst als Quadrik bezeichnet. Die Quadrik Q^f nur einer Ebene wird fundamentale Quadrik genannt. Man beachte, dass $Q(\cdot)$ und $D(\cdot)$ für drei- und vier-komponentige Vektoren entsprechend definiert wurden.

Quadrik-Fehlermaß

Das Fehlermaß eines Knotens lässt sich für beliebig viele Ebenen durch nur 10 Werte beschreiben. Um aber die geometrische Abweichung jeder Simplifizierungsstufe mit dem Originalmodell und nicht mit dem Modell aus der vorherigen Stufe zu bestimmen, müsste zu jedem Knoten die Menge der Flächen aus dem Originalmodell vorgehalten werden, mit der er korrespondiert. Nicht nur, dass dies Speicherplatz bedarf, auch müsste für jede Bewertung die Summe aller homogenen Matrizen bzw. der 10 Parameter neu bestimmt werden.

Zur Beschleunigung des Verfahrens zur Bewertung einer Knotenverschmelzung wird wieder die einfache Additivität der Quadratabstände bzw. Quadriken ausgenutzt. Für das Quadrik-Fehlermaß wird wie folgt vorgegangen. Initial wird für jeden Knoten des Originalnetzes die Abstandskadratsumme zu seinen anliegenden Ebenen bestimmt, die durch die Dreiecke gegeben sind, die v_i als Eckpunkt beinhalten. Für den Knoten v_i wird die Quadratsumme mit $Q_i(v)$ und die zugehörige Quadrik mit Q_i bezeichnet, wobei $Q_i(v_i) = 0$, da gemäß Konstruktion alle anliegenden Ebenen genau durch v_i gehen. Zur Bewertung eines Knotenpaares (v_i, v_j) und zur Bestimmung des Kontraktionspunktes \bar{v} (siehe Abschnitt 3.1.4) wird nun nicht die Menge der zu v_i und v_j korrespondierenden Ebenen bestimmt und die Fehlermaße addiert, sondern lediglich die Quadriken bzw. 10 Parameter des Knotenpaares addiert. Die Kosten für eine Kontraktion $(v_i, v_j) \rightarrow \bar{v}$ sind also

$$Q(\bar{v}) = Q_i(\bar{v}) + Q_j(\bar{v}) = \bar{v}^t (Q_i + Q_j) \bar{v}.$$

Die Summe der Quadriken $Q_i + Q_j$ wird bei erfolgter Kontraktion dem Verschmelzungspunkt \bar{v} als Quadrik zugewiesen und für folgende Kostenbestimmungen von Knotenpaaren, die \bar{v} enthalten, herangezogen. Es muss sich also für den gesamten Simplifizierungsprozess nur eine Quadrik pro Knoten gemerkt werden und nicht mehr die Menge aller korrespondierenden Flächen.

Das Fehlermaß ist also sehr speicherplatzsparend und zudem sehr schnell, da eine Bewertung nur einer Multiplikation mit den 10 Parametern der Quadrik bedarf und die Aktualisierung der Quadrik für einen neuen Knoten durch eine einfache Addition dieser Parameter möglich ist. Bei dem Maximalabstand als Fehlermaß aus der Arbeit von Ronfard und Rossignac [16], auf die dieser Ansatz zurückgeht, war es hingegen noch nötig, sich alle Ebenen zu merken und bei der Bewertung zu berücksichtigen.

Die einfache Addition der Quadriken hat aber zur Folge, dass Flächen und damit Dreiecke, die v_i und auch v_j beinhalten, mehrfach gewertet werden. Beim ersten Kontraktionsschritt sind dies meist zwei Dreiecke, die an der gemeinsamen Kante liegen, die doppelt bewertet werden. In weiteren Iterationsstufen steht eine Quadrik für den Quadratabstand zu einer Vielzahl von Dreiecken. Jedes Dreieck kommt in allen Quadriken des Gesamtnetzes zu jedem Zeitpunkt der Simplifizierung aber nur genau drei Mal vor, da es genau über seine drei Eckpunkte einfließt. Es kann also kein Dreieck mehr als drei Mal in die Bewertung eingehen. Zudem gehen gerade die zentralen Dreiecke, denen beide Knoten angehören, mehr ein, was den Verschmelzungspunkt stärker zwischen den Verschmelzungskandidaten hält, Randbereiche werden hingegen schwächer bewertet.

In Abbildung 3.4 trifft die Doppelbewertung für den Abstand von \bar{v} zu $g_3(*)$ zu, da $Q_i(v)$ und $Q_j(v)$ diese Gerade beinhalten. Für die Abbildung lautet das Maß also

$$Q(v) = D_1^2(v) + 2D_2^2(v) + D_3^2(v).$$

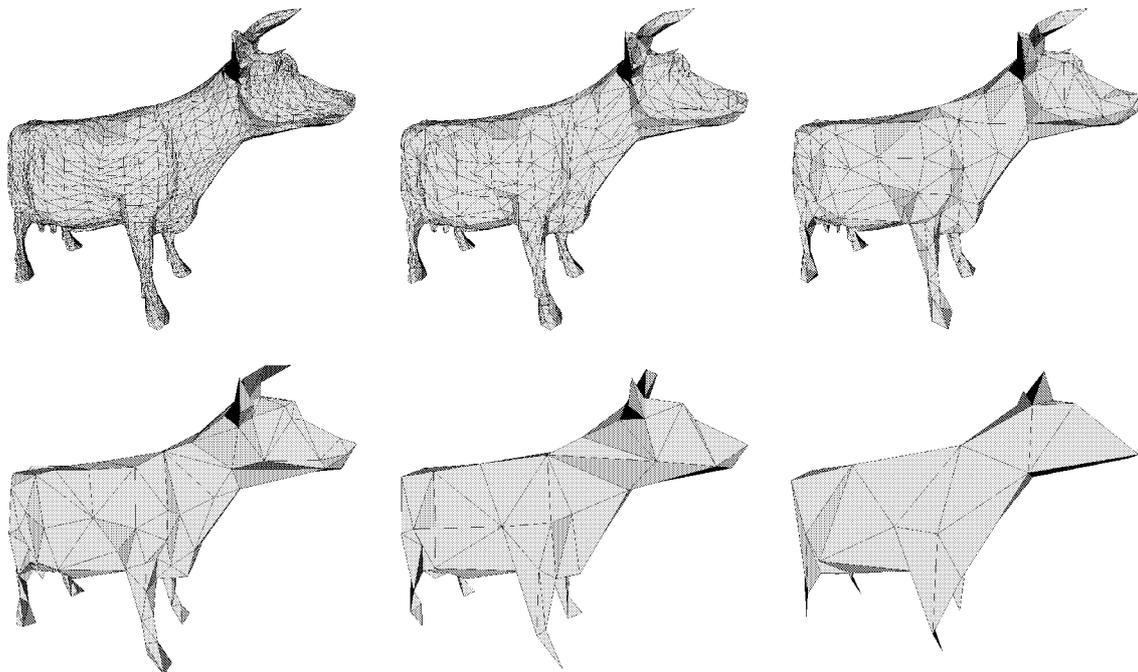


Abb. 3.5: Detailstufen einer mit dem Quadrik-Fehlermaß simplifizierten Kuh mit 2.903, 1.000, 400, 200, 100 und 50 Knoten

Da aber auch die Summe der einfachen Abstandsquadrate als Fehlermaß rein heuristisch ist, weil der Abstand zur Ebene nicht gleich dem Abstand zum Dreieck sein muss, ist die Ungenauigkeit der Mehrfachbewertung als unkritisch anzusehen. Zudem ist die Abstandsbetrachtung am ehesten für zentrale Dreiecke exakt.

Ein Beispiel zum Quadrik-Fehlermaß im Zweidimensionalen findet der Leser in Anhang C.3, in dem auch der Unterschied zur einfachen Bewertung einer jeden Geraden beschrieben ist.

Neben der Sparsamkeit im Speicherplatzverbrauch und Effizienz erzeugt das Quadrik-Fehlermaß auch sehr gute optische Ergebnisse. Als Beispiel hierzu dient in Abbildung 3.5 das Modell einer mit dem Quadrik-Fehlermaß simplifizierten Kuh, das mit verschiedenen Knotenzahlen dargestellt wird. Um die resultierenden Flächen besser zu sehen, ist hier wie auch in den meisten anderen Abbildungen dieser Arbeit eine konstante Schattierung^G (flat shading) gewählt worden. Schattierung durch Interpolation (Gouraud- oder Phong-Shading) würde den optischen Unterschied der Stufen noch weiter reduzieren. Das Drahtmodell soll die einzelnen Dreiecke besser hervorheben.

Das Originalmodell besitzt 5.804 Dreiecke, 2.903 Knoten und 8.706 echte Netzkanten. Die vollständige Simplifizierung auf einen Knoten dauert knapp unter einer viertel Sekunde auf einem gewöhnlichen 1.800 MHz Laptop (Rechnerspezifikation siehe Anhang A.1.5). Obwohl das Originalmodell nicht sehr fein ist, ist doch gut zu erkennen, dass Strukturen wie die Hörner lange erhalten bleiben. Ausführliche Qualitätsuntersuchungen sind in Kapitel 4 zu finden.

3.1.4 Bestimmung des Verschmelzungspunktes

In dem vorherigen Abschnitt ist die sukzessive Knotenverschmelzung und deren Kostenberechnung besprochen worden. Dabei wurde die Kontraktion immer mit $(v_i, v_j) \rightarrow \bar{v}$ angegeben. Was noch fehlt, ist die Bestimmung der Position von \bar{v} . Diese ist nicht nur zur eigentlichen Kontraktion, sondern auch zur Kostenberechnung nötig, da das Knotenpaar (v_i, v_j) über $Q(\bar{v})$ bewertet wird. Der optimale Punkt \bar{v} für das Quadrik-Fehlermaß wäre der Ort, an dem $Q(v)$ minimal ist. Da $Q(v)$ eine quadratische Gleichung in drei Unbekannten ist, ist die Optimumfindung ein lineares Problem der Form

$$\text{grad}(Q(v)) = 0.$$

In der homogenen Form mit $Q = \begin{pmatrix} q_{11} & \cdots & q_{14} \\ \vdots & & \vdots \\ q_{14} & \cdots & q_{44} \end{pmatrix}$ lauten für

$$\begin{aligned} Q(\tilde{v}) &= \tilde{v}^t Q \tilde{v} \\ &= q_{11}x^2 + 2q_{12}xy + 2q_{13}xz + 2q_{14}x + q_{22}y^2 + 2q_{23}yz + 2q_{24}y + q_{33}z^2 + 2q_{34}z + q_{44} \end{aligned}$$

die Ableitungen nach den drei Unbekannten x , y und z liefern das Gleichungssystem

$$\begin{aligned} 2q_{11}x + 2q_{12}y + 2q_{13}z + 2q_{14} &= 0 \\ 2q_{12}x + 2q_{22}y + 2q_{23}z + 2q_{24} &= 0 \\ 2q_{13}x + 2q_{23}y + 2q_{33}z + 2q_{34} &= 0 \end{aligned} .$$

Den Faktor 2 gekürzt, heißt die zu lösende Gleichung

$$\underbrace{\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{=: \bar{Q}} \bar{v} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.5)$$

$$\bar{v} = \bar{Q}^{-1} \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^t.$$

Die letzte Zeile in der Matrix \bar{Q} ist für eine geschlossene Darstellung hinzugefügt worden. Die Invertierung ist natürlich nur zulässig, wenn \bar{Q} invertierbar ist. Hierzu ist aber lediglich zu zeigen, dass die obere linke 3×3 -Untermatrix A (Bezeichnung gemäß Gleichung (3.2)) von \bar{Q} eine Inverse besitzt, da genau dann auch ganz \bar{Q} invertierbar ist. Dies ist dann der Fall, wenn die Determinante $\det(A) = \det(\bar{Q}) = \bar{Q}_{44} = Q_{44}$ von Null verschieden ist. Die Adjunkten

$$Q_{ij} = (-1)^{i+j} \det \begin{pmatrix} q_{11} & \cdots & q_{1j-1} & q_{1j+1} & \cdots & q_{14} \\ \vdots & & \vdots & \vdots & & \vdots \\ q_{i-1} & \cdots & q_{i-1j-1} & q_{i-1j+1} & \cdots & q_{i-14} \\ q_{i+1} & \cdots & q_{i+1j-1} & q_{i+1j+1} & \cdots & q_{i+14} \\ \vdots & & \vdots & \vdots & & \vdots \\ q_{14} & \cdots & q_{j-14} & q_{j+14} & \cdots & q_{44} \end{pmatrix}$$

der letzten Zeile von \bar{Q} entsprechen denen von Q . Von der Inversen wiederum sind zur Bestimmung von \bar{v} nur die oberen drei Werte der letzten Spalte interessant. Die Lösung für den Verschmelzungspunkt lautet

$$x = \frac{Q_{41}}{Q_{44}}, \quad y = \frac{Q_{42}}{Q_{44}}, \quad z = \frac{Q_{43}}{Q_{44}}. \quad (3.6)$$

Zur Berechnung von \bar{v} sind somit bei dieser direkten Lösungsmethode 48 Multiplikationen, 24 Additionen bzw. Subtraktionen und 3 Divisionen nötig. Wendet man zur Lösung von Gleichung (3.5) die Gauß-Elimination mit Spaltenpivotisierung an, sind nur noch 13 Multiplikationen, 8 Subtraktionen und 3 Divisionen nötig.

Für reguläre \bar{Q} ist \bar{Q} positiv definit und es existieren damit eine Vielzahl weiterer effizienter Lösungsverfahren wie das Cholesky-Verfahren oder das numerisch stabilere Householder-Verfahren bzw. die QR-Zerlegung (QR decomposition) mittels Householder-Transformation [8, 76], insbesondere wenn man die Gleichung (3.5) in den \mathbb{R}^3 überträgt. Sie lautet dann

$$\begin{aligned} Av + B &= 0 \\ v &= -A^{-1}B. \end{aligned} \quad (3.7)$$

A ist regulär, wenn Q regulär ist.

Mehrdeutige Lösungen

Es gibt Fälle, für die keine eindeutige Lösung existiert. Das ist genau dann der Fall, wenn A bzw. \bar{Q} keine Inverse besitzt bzw. singulär ist. Die Lösung ist dann entweder eine Gerade oder Ebene, entsprechend, ob die Normalen der Dreiecksebenen in einer Ebene liegen oder die Dreiecksebenen selbst

alle zueinander parallel sind. Für diese Fälle, bei denen A bzw. \bar{Q} nur positiv semidefinit ist, gibt es abgewandelte Verfahren wie die Cholesky-Methode mit Spaltenpermutation [77] oder die erweiterte Cholesky-Methode [78] zur Bestimmung der Lösungsmenge. Da aber nicht eine Menge gesucht wird, sondern ein eindeutiger Verschmelzungspunkt, bieten sich diese Verfahren hier nicht an.

Ein Beispiel im Zweidimensionalen für eine mehrdeutige Lösung wären zwei zueinander parallele Geraden. Die elliptischen Isolinien mit ihrem Zentrum als eindeutige Lösung entarten in Geradenpaare, die als Lösung im Zentrum eine Gerade besitzen. Des Weiteren gilt, wenn alle Geraden identisch sind oder nur eine in die Quadrik einfließt, ist diese Gerade selbst Lösung.

Für solche Fälle, in denen keine eindeutige Lösung existiert, wird eine andere Strategie zur Findung des Verschmelzungspunktes benötigt. Garland und Heckbert [31] schlagen vor, auf dem Geradenstück zwischen v_i und v_j des aktuell zu untersuchenden Knotenverschmelzungspaares nach einer möglichst guten Lösung zu suchen. Hierzu ist in dem Fehlermaß v durch das Geradenstück

$$v_i + \lambda(v_j - v_i) \quad \text{mit } \lambda \in [0, 1]$$

zu ersetzen und die daraus resultierende quadratische Gleichung in einer Unbekannten λ nach dieser abzuleiten. Wenn das Geradenstück nicht parallel zur Lösungsgeraden bzw. Lösungsebenen ist, hat die quadratische Gleichung ein eindeutiges Minimum, das aber nicht unbedingt zwischen 0 und 1 liegen muss.

Für diesen Fall oder der Parallelität des Geradenstückes zu den Lösungsmannigfaltigkeiten muss eine letzte Stufe in der Lösungsfindung her, um ein eindeutiges \bar{v} zu finden. Garland und Heckbert untersuchen nun die Endpunkte v_i und v_j und wählen den mit dem geringsten Fehler aus oder einen beliebigen von beiden, falls auch diese den gleichen Fehler besitzen.

Eine andere Strategie für nicht eindeutige Lösungen losgelöst von dem Quadrik-Fehlermaß könnte die Position für den Verschmelzungspunkt nehmen, dessen Abstandsquadratsumme von allen zu v_i und v_j benachbarten Knoten am geringsten ist. Dieses Minimum existiert immer und ist eindeutig. Es entspricht dem Zentrum bzw. Schwerpunkt des Polygons durch die k adjazenten Knoten v_l^* , $l \in \mathbb{N}_k$:

$$\bar{v} = \frac{1}{k} \sum_{l=1}^k v_l^* .$$

Der Schwerpunkt muss aber nicht innerhalb des Polygons liegen, was zu einem unschönen Umsprung von Dreiecksorientierungen führen kann.

Ein letzter Vorschlag der Lösungsfindung bei Quadriken mit singulärer Matrix A bzw. \bar{Q} greift die Idee des Abstandsgewichtes eines Punktes zur Erhaltung wichtiger Netzknoten vom Ende des Abschnitts 3.1.5 auf. Die Gewichtung erfolgt durch eine spezielle dort vorgestellte Quadrik, dessen Untermatrix A per Konstruktion regulär ist. Die Idee ist nun folgende: durch einfache Addition solch einer Spezialmatrix auf die Quadrik mit nicht eindeutiger Lösung wird das Resultat regulär (Begründung am Ende des Abschnitts 3.2.2) und ermöglicht die Ermittlung eines eindeutigen Verschmelzungspunktes. Als Knoten für die spezielle Quadrik, die das Abstandsquadrat zu einem Knoten gewichtet, könnte der Mittelpunkt $v_m = (v_i + v_j)/2$ der Verschmelzungskandidaten genommen werden. Der damit berechnete Verschmelzungspunkt liegt auf dem Geradenstück zwischen v_m und der senkrechten Projektion von v_m auf die Lösungsmannigfaltigkeit der ursprünglichen Quadrik. Die Lösung wird somit nahe dem Mittelpunkt v_m gesucht.

Die letzten beiden Stufen der von Garland und Heckbert vorgeschlagenen Optimumsuche von \bar{v} oder der alternativen Strategien wurde bei der dieser Arbeit zugrundeliegenden Implementation durch einen

einfachen Vergleich der Fehler der Endpunkte v_i und v_j und dem Punkt $(v_i + v_j)/2$ zwischen diesen ersetzt, da der Rechenaufwand für diese Spezialfälle nicht im Verhältnis zum Nutzen steht. Zudem tritt die Uneindeutigkeit des Minimums des Fehlermaßes in der dieser Arbeit zugrundeliegenden Anwendung vorkommenden gemessenen Isoflächen über die Elektronendichte im Gegensatz zu konstruierten Flächen nur aus Gründen der Ungenauigkeit der Zahlendarstellung im Rechner oder der Feinheit der Messapparatur und somit selten auf. Eine Zählung der singulären Quadriken im Abschnitt 4.5 des Ergebnis-Kapitels bestätigt dies, wobei die vermeidlich singulären Quadriken eher von der im Folgenden beschriebenen Beinahe-Singularität herrühren.

Kommen wir nochmal zurück zu der Frage, ob eine Singularität der zu invertierenden Matrix A bzw. \bar{Q} vorliegt. Theoretisch ist dies wie schon gesagt für $\det(A) = 0$ der Fall. Da aber im Rechner nur eine endliche Genauigkeit vorzufinden ist, sollte der Betrag der Determinante nicht auf Null sondern auf ein $\epsilon > 0$ getestet werden. Dieses ϵ muss zudem verhältnismäßig hoch gewählt werden (in der Implementation wurde $\epsilon = 0,0001$ gewählt), da zum einen die Annäherung der Determinante an Null nicht gut mit der Singularität korreliert – andere Ansätze sind dort robuster – und zum anderen der Verschmelzungspunkt sich nahe der Singularität ohne einen wesentlichen Fehler vom Optimum entfernen kann. Durch die große Wahl von ϵ insbesondere wegen des instabilen Verschmelzungspunktes, stellen die singulären Quadriken kein Problem für die Anwendung dar. Ein Problemfall für eine zu kleine Wahl von ϵ ist in Abschnitt 4.8.2 beschrieben.

Alternativ kann die Singularität der Matrizen auch während der Lösung mittels der Gauß-, Cholesky- oder Householder-Verfahren bestimmt werden. Diese Methoden ließen während der Berechnung solche Analysen zu. Auch wäre die Bestimmung der Eigenwerte oder eine Singulärwertzerlegung eine wenn auch aufwendige Möglichkeit zur Bestimmung von singulären Matrizen, die Lindstrom in [33] einsetzt. Insbesondere für große Matrizen existieren in Programmbibliotheken spezielle Routinen zur Analyse solcher kritischen Matrizen. Sie bieten zwar durch Konditionsschätzer bessere zur Singularität korrelierte Werte als die Determinante, spielen ihre Stärke aber erst bei großen Matrizen aus. Lindstrom schlägt in [28] vor, die Winkel zwischen den Ebenennormalen paarweise zu vergleichen, und eine Sonderbehandlung für nahezu parallele Normalen vorzunehmen. Dies kann nur bei der initialen Bestimmung der Quadriken an jedem Punkt durchgeführt werden, da nur dann ohne Zwischenspeicherung die Ebenen für jeden Knoten vorliegen. Dies stellt aber keine Einschränkung dar, wenn dafür gesorgt wird, dass singuläre Quadriken zum Beispiel durch ein Punktgewicht regularisiert werden, da einmal reguläre Quadriken nicht wieder singulär werden können (siehe Abschnitt 3.2.2).

Optimaler Verschmelzungspunkt oder Originalknoten

Die Möglichkeit des verwendeten Fehlermaßes, einen optimalen Verschmelzungspunkt zu finden, ist ein Grund für die gute Qualität der Vereinfachung. Es gibt aber Anwendungen, wo diese Vorteile – gute Annäherung an das Originalnetz, wohlgeformte Oberfläche, möglichst gleichseitige und größtmäßig ähnliche Dreiecke – anderen Wünschen nachstehen. Wegen der schnellen Berechenbarkeit mit konstanter Geschwindigkeit pro Knoten wird dies weniger der Rechenaufwand für die Bestimmung des Verschmelzungspunktes sein, als vielmehr der zusätzlich benötigte Speicherplatz für die Koordinaten des neuen Punktes \bar{v} . Die neuen Punktkoordinaten können zwar als Punktdifferenz gespeichert werden, um mit weniger Platz eine genauere Position anzugeben², aber sie machen immer

²Da für gewöhnlich der Verschmelzungspunkt nahe der Kontraktionskandidaten liegt, ist die Punktdifferenz klein im Verhältnis zu den absoluten Knotenkoordinaten. Kleine Werte können absolut betrachtet in der normalisierten Gleitkommazahl- (Floatingpoint-) Struktur nach der IEEE-Norm heutiger Rechner mit mehr Nachkommastellen angegeben

noch den Großteil des Speicherplatzes für die Angabe der Simplifizierung aus. Kommen Netzattribute neben der Position hinzu, steigt der benötigte Speicherplatz weiter. Würde die Knotenwahl auf bereits existierende Knoten des Originalnetzes fallen, würden keine zusätzlichen Daten für die Verschmelzungspunkte benötigt. Es muss also Approximationsqualität gegen Speicherplatzeffizienz abgewogen werden. Wie stark sich die entsprechende Knotenwahl auf die Netzqualität auswirkt, wird im Abschnitt 4.4.2 besprochen.

Werden also Punktepaare immer nur auf einen existierenden Punkt zusammengezogen und wird sich bei der Paarangabe noch darauf geeinigt, dass der Verschmelzungspunkt der erstgenannte Punkt des Punktepaares ist, kann eine sukzessive Simplifizierung lediglich anhand einer Reihe von Punktepaaren beschrieben werden. Zur Bewertung ist nur $Q(v_i)$ und $Q(v_j)$ zu berechnen und jede Approximationsstufe besitzt eine Untermenge der ursprünglichen Netzknoten an ihrer anfänglichen Position. Auch bei der Aktualisierung der Verschmelzungskosten der Kandidaten sind bloß noch halb so viele Neuberechnungen nötig, da nur ein Knoten seine Position ändert oder besser gesagt durch einen anderen Knoten ersetzt wird.

Obwohl in der Implementation die fortschreitende Vereinfachung mittels eines Progressive Meshes (siehe Abschnitt 2.6.2) gespeichert wird, um ohne erneute Berechnung jede Simplifizierungsstufe mit linearem Aufwand zu den Kanten zu erreichen, ist die Entscheidung auf die exakte Berechnung gefallen, da der Speicherplatzbedarf für die Netze mit bis zu 200.000 Knoten, die für die dieser Arbeit zugrundeliegende Anwendung von Interesse sind, bei heute aktuellen Rechnern kein Speicherplatzproblem darstellt.

3.1.5 Einschränkungen

Der eigentliche Kernalgorithmus ist nun besprochen, aber es gibt Fälle, die einer Sonderbehandlung bedürfen. Kanten, Ecken oder sonstige unstetige Bereiche werden schon durch das Fehlermaß entsprechend ihrer Ausprägung ausreichend berücksichtigt und damit möglichst erhalten.

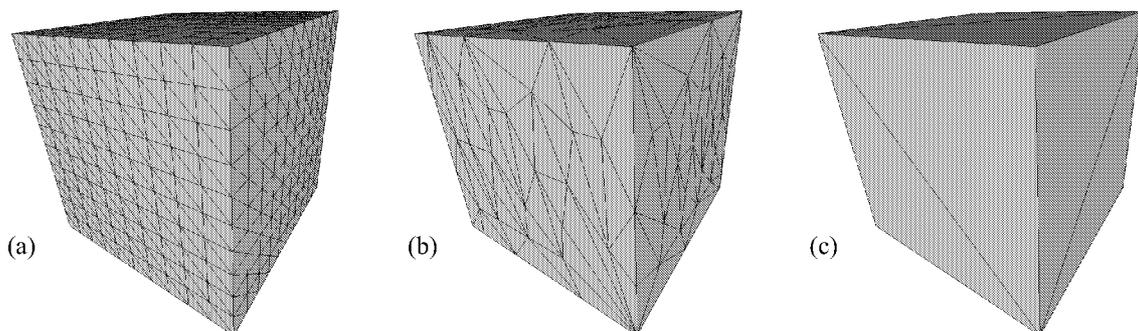


Abb. 3.6: Würfel mit (a) original 610 Knoten, (b) 150 Knoten und (c) 8 Knoten; ohne Drahtmodell würden alle Abbildungen identisch aussehen, da während der Vereinfachung das Quadrik-Fehlermaß Minimalkosten von Null aufweist

In Abbildung 3.6 sind die Kontraktionskosten für alle Netzkanten bei optimaler Findung des Verschmelzungspunktes Null. Die optimalen Verschmelzungspunkte liegen möglichst in den Ebenen angrenzender Dreiecke, bleiben also auf der Würfeloberfläche, -kante bzw. -ecke. Erst nach der letzten

werden oder andersherum kann ein platzsparender Datentyp bei gleicher Genauigkeit für die Verschmelzungspunktkoordinaten herangezogen werden (zum Beispiel in `C float` (typischerweise 4 Byte) statt `double` (typischerweise 8 Byte)).

Abbildung würden nach und nach die Ecken einfallen und echt positive Kosten entstehen, da die angrenzenden Flächen keine nicht leere Schnittmenge mehr besitzen.

Materialeigenschaften oder Farbgebungen werden bei dem in dieser Arbeit beschriebenen Ansatz nicht berücksichtigt, könnten aber über ein erweitertes Fehlermaß höherer Dimension in die Entscheidungsfindung einfließen. Dies ist durchaus sinnvoll, da zum Beispiel Farbsprünge eine hohe visuelle Relevanz haben. In der dieser Arbeit zugrundeliegenden Anwendung besitzt das Netz keinerlei über die Netzgeometrie und -topologie hinausgehenden Eigenschaften. Eine spezielle Behandlung der Materialeigenschaften ist somit nicht nötig.

Zu Anfang der Verfahrensbeschreibung wurde hervorgehoben, dass beliebige Mannigfaltigkeiten verarbeitet werden können. Bei der Wahl von großen Schwellwerten für die Kantenkandidaten ist theoretisch auch jeder Knoten mit jedem verschmelzbar. Letzteres ist schon aus Gründen der daraus resultierenden langen Laufzeit uninteressant, aber auch die sonstige Generalität sollte durch gewisse Einschränkungen gedämpft werden. Die folgenden drei Punkte fügen Bestrafungen für Knotenverschmelzungspaare ein, die unschöne Effekte der Simplifizierung mildern helfen sollen. Der erste Punkt kann zur Initialisierungszeit durchgeführt werden, die letzten beiden können erst während der Vereinfachung Problemfälle entdecken und verhindern helfen.

Offene Ränder

Was bisher unberücksichtigt blieb, sind offene Ränder. Diese würden sogar wegen ihrer oft geringen Zahl angrenzender Flächen besonders früh zur Kontraktion kommen, dabei aber die Lage des Randes unberücksichtigt bleiben. Abhilfe schaffen hier elegant zusätzliche "durchsichtige" Dreiecke an diesen offenen Rändern, die nicht dem Netz hinzugefügt aber zur Bewertung von Randpunkten herangezogen werden.

Hierzu werden alle echten Kanten im Netz aufgesucht, an die nur genau ein Dreieck angrenzt. Die Knoten v_i und v_j dieser Kanten erhalten bei der einmaligen anfänglichen Quadrik-Zuweisung als zusätzliche Ebene eine Ebene senkrecht zu dem einzigen die Kante angrenzenden Dreieck t zugewiesen. Da solche offenen Ränder für die visuelle Güte sehr wichtig sind, wird die imaginäre Ebene beim Hinzufügen zur Quadrik eines Knotens noch mit einem hohen Faktor f versehen, der zwar die Kante nicht unüberwindbar bestraft, sie aber lange bewahrt oder zumindest sich nur nah dieser Ebene verschieben lässt (in der Implementation wurde $f = 1000$ gewählt). Sei n der Normalenvektor des Dreiecks t , dann lautet der normierte Normalenvektor der senkrechten Ebene

$$\overset{\star}{n} = \frac{n \times (v_j - v_i)}{\|n \times (v_j - v_i)\|_2}$$

und damit die fundamentale Bestrafungsquadrik

$$f \begin{pmatrix} \overset{\star}{n} \\ \overset{\star}{n} v_i \end{pmatrix} (\overset{\star}{n}, -\overset{\star}{n} v_i).$$

Das Schöne bei dieser Vorgehensweise ist, dass am weiteren Algorithmus keine Änderungen durchgeführt werden müssen, also während der eigentlichen Simplifizierung keine aufwendige Fallunterscheidungen nötig sind. Auch ist der Kantenzusammenfall entlang gerader offener Ränder durch die Bestrafungsebene nicht beeinträchtigt, so dass die Dreiecksdichte in solchen Randbereichen nicht

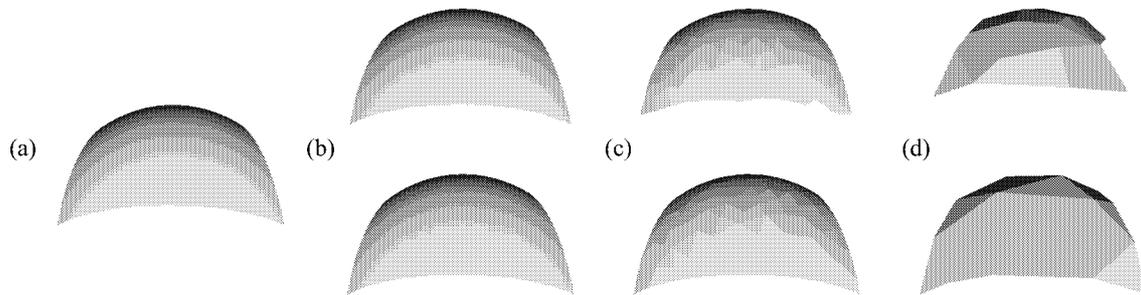


Abb. 3.7: Eine gleichmäßig gekrümmte Fläche mit (a) originalen 9216 Knoten, (b) 2000, (c) 200 und (d) 20 Knoten; in der oberen Reihe ist die Kantenbestrafung ausgeschaltet, in der unteren Reihe hält diese Bestrafung den offenen Rand des Modells bei

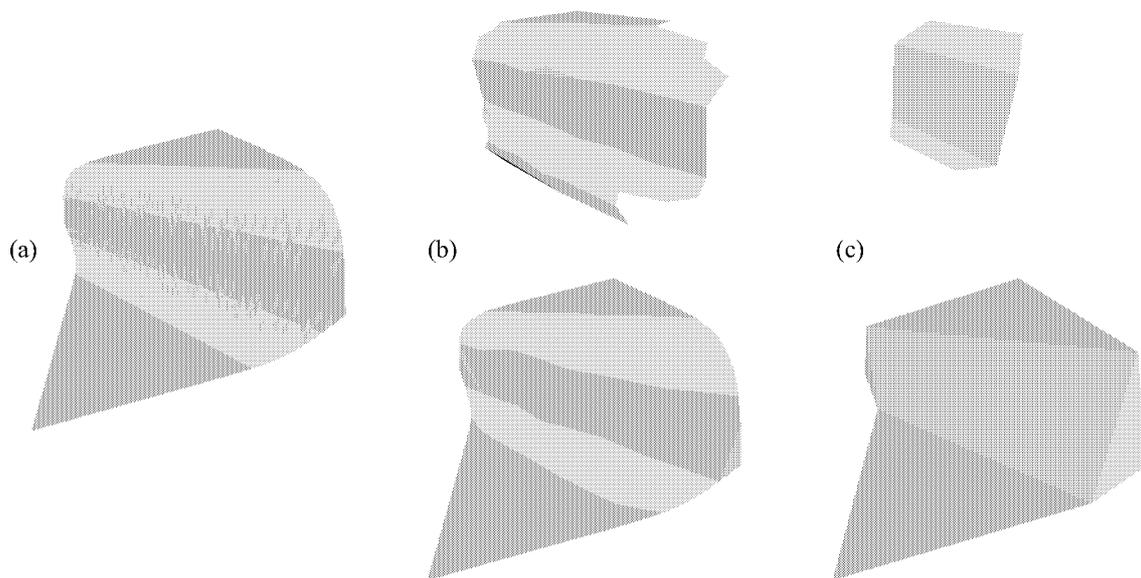


Abb. 3.8: Ein mittig einmal geschwungenes Blatt Papier mit (a) original 1252 Knoten, (b) 100 und (c) 8 Knoten; in der oberen Reihe ist die Kantenbestrafung ausgeschaltet und lässt das Bild durch große Ränddreiecke stark vom Original abweichen, in der unteren Reihe hält diese Bestrafung den offenen Rand des Modells bei

höher ist als anderswo im Netz. Kritischer sind unruhige zackige offene Ränder, die durch diese Vorgehensweise schwer vereinfacht werden können. Ein Beispiel dazu findet der Leser in Abschnitt 4.8.1.

In Abbildung 3.7 und 3.8 ist in der unteren Reihe die Bewahrung der Kante durch die Randbestrafung und die daraus resultierenden guten Annäherungen an das Originalmodell zu sehen. Für die Simplifizierung der oberen Reihe wurde von der Bestrafung der Randkanten abgesehen. Ein weiteres Beispiel für die gute Arbeit der Randbestrafung ist die Geländeoberfläche aus Abbildung 2.20, deren Rand dadurch bis zum Ende aufrecht erhalten wird.

Die gekrümmte Fläche in Abbildung 3.7 besitzt eine sehr feine und äquidistante Aufteilung der Oberfläche, so dass im Ausdruck im Gegensatz zum Bildschirm trotz einer konstanten Schattierung^G bei 2000 Knoten kaum ein Unterschied zum Original zu erkennen ist. Würde man eine interpolierte Schattierung für die Darstellung wählen, würde man sogar mit 200 Knoten am Bildschirm kaum einen Unterschied zum Original feststellen können. Das Ausfransen der Ränder an der gekrümmten Oberfläche schreitet in der oberen Reihe nur langsam voran, da die Ränddreiecke sehr klein sind.

Dies ist bei dem geschwungenen Blatt aus Abbildung 3.8 anders, bei dem die flachen Ecken durch nur wenige große Dreiecke modelliert wurden und bei einer Simplifizierung ohne Randbestrafung diese durch ihr Verschwinden schnell eine unansehnliche Darstellung bewirken. Zudem kommt hier hinzu, dass diese großen Randdreiecke gerade in flachen Bereichen liegen, die durch das Quadrik-Fehlermaß früh zur Simplifizierung vorgeschlagen werden.

Diese Vorgehensweise durch Hinzufügen von nicht real existierenden Dreiecken bzw. Ebenen zur starken Gewichtung von offenen Rändern bzw. zur Bestrafung der Kontraktion derselben kann auch auf andere Ränder erweitert werden und zum Beispiel Ränder diskreter Materialeigenschaften erhalten. Als Beispiel ist eine Landkarte vorstellbar, die für jedes Land eine separate Farbe enthält. Die Landesgrenzen ließen sich über diesen Mechanismus möglichst lange erhalten.

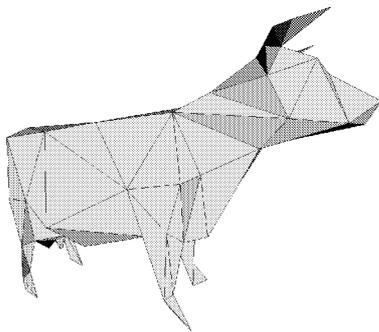


Abb. 3.9: Ein auf 100 Knoten simplifiziertes Kuhmodell mit gewichteten Hörnerspitzen; vergleiche hierzu die vorletzte Simplifizierungsstufe in Abbildung 3.5

Ein ähnlicher Ansatz könnte auch helfen, bei einem Modell sehr wichtige Punkte gesondert zu gewichten. Zum Beispiel ließen sich so die Hörnerspitzen der Kuh aus Abbildung 3.5 länger erhalten. Hierzu wird den Knoten bei der initialen Quadrik-Zuweisung keine weitere Ebene zugewiesen, sondern es fließt über eine spezielle Quadrik der quadratische Abstand zu diesen Knoten ein. Diese Quadrik hätte für den stärker zu gewichtenden Knoten $\tilde{v} = (x^*, y^*, z^*)^t$ die spezielle Form

$$Q = \begin{pmatrix} 1 & 0 & 0 & -x^* \\ 0 & 1 & 0 & -y^* \\ 0 & 0 & 1 & -z^* \\ -x^* & -y^* & -z^* & v^{*t}v^* \end{pmatrix},$$

was dem Fehlermaß $Q(v)$ lediglich

$$\tilde{v}^t Q \tilde{v} = v^t v - 2(x^*x + y^*y + z^*z) + v^{*t}v^* = \|v - \tilde{v}\|_2^2$$

als Kosten hinzufügt. Die Quadrik entspricht der Summe dreier fundamentaler Quadriken, die durch Ebenen erzeugt werden, die jeweils senkrecht zu den Koordinatenachsen stehen und durch den zu gewichtenden Knoten gehen. Das Ergebnis für das Kuhmodell mit gewichteten Hörnerspitzen ist in der nebenstehenden Abbildung 3.9 zu sehen. Sie zeigt das Modell mit 100 Knoten entsprechend der vorletzten Simplifizierungsstufe aus Abbildung 3.5.

Dreiecksidentität

Liegen zwei Dreiecke aufeinander, dann hat die Grafikhardware es schwer, sich bei der Darstellung zu entscheiden, welches Dreieck zu oberst gezeichnet werden soll. Haben diese aufeinanderliegenden Dreiecke auch noch die entgegengesetzte Normalenrichtungen, was bei der Netzvereinfachung den Normalfall darstellt, dann ist der Farbunterschied der Dreiecke für gewöhnlich gravierend. Aus diesem Grund kommt es in solchen Fällen oft zu Artefakten, die durch wechselnde Darstellung beider Dreiecke verursacht werden. Dies drückt sich dann durch Flackern oder durch eine gestreifte Darstellung der Dreiecke aus.

Dieses Aufeinanderfallen der Dreiecke kommt zum Beispiel schon bei der Eliminierung einer Kante aus einem Tetraeder mit vier Dreiecken vor, bei dem dann die letzten beiden Dreiecke genau übereinander liegen. Ein Tetraeder ist für gewöhnlich bei Modellen mit einer einfachen geschlossenen

Oberfläche wie bei der Kuh aus Abbildung 3.5 das Objekt mit vier Knoten, auf das sich das Modell am Ende einer Vereinfachungsfolge zurückzieht. Das Zusammenfallen zweier Dreiecke kann aber häufiger während der Simplifizierung vorkommen. Die Unterbindung einer Knotenverschmelzung, die dieses Flackern zur Folge hat, ist nur sinnvoll, wenn noch andere Verschmelzungen möglich sind, die ähnlich hohe Kosten haben.

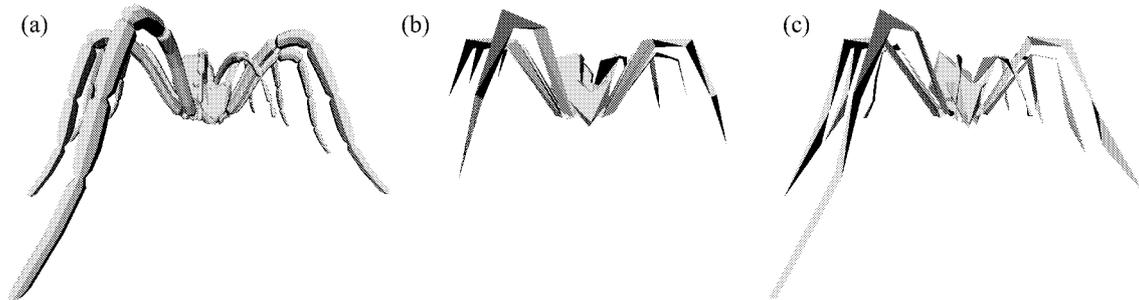


Abb. 3.10: (a) Originalmodell einer Spinne mit 4670 Knoten, (b) Vereinfachung auf 160 Knoten ohne Bestrafung der Dreiecksidentität und (c) mit Bestrafung

Neben der Verbesserung der Darstellung wird bei filigranen Objekten zudem die Gesamtstruktur besser erhalten, da feine Details nicht so schnell komplett in sich zusammenfallen. Ein Beispiel zeigt Abbildung 3.10, in der die Spinnenbeine und Fühler aus (a) in (c) durch die Bestrafung des Zusammenfalls identischer Dreiecke länger aufrecht erhalten werden als in (b), wo die Bestrafung ausgeschaltet wurde. Die oben angesprochenen Artefakte liegen ebenso in (b) am Ende des vorderen Beines vor. Sie sind auf dem Bildschirm störend zu sehen, kommen im Druck jedoch nicht so zur Geltung.

In der Implementation kann optional das Aufeinanderfallen von zwei Dreiecken zurückgehalten werden, wenn der bestrafte Kontraktion bald eine Kollabierung folgt, die keine identischen Dreiecke erzeugt. Irgendwann muss aber das Zurückhalten aufgehoben werden, da sonst eine Vereinfachung bis zum Ende mit der Eliminierung aller Netzkanten nur bei Modellen mit offenen Rändern zu realisieren wäre und die Vereinfachung von Netzen mit vielen nicht verschmelzenden Zusammenhangskomponenten wie in Abbildung 3.2 (a) früh abbrechen würde. Bei solchen Netzen wie dem letztgenannten sollte man diese Bestrafung besser ausschalten, da die Abarbeitung der Kontraktionen visuell ungünstig sein kann.

Das Zurückhalten der Kontraktion wird dadurch bewirkt, dass die Quadrik mit einem Bestrafungsfaktor multipliziert wird (in der Implementation wurde 1000 gewählt) und in die Hierarchie der Abarbeitungsliste der Kantenkontraktionen neu einsortiert wird. Die Quadrik kann nicht Null sein und damit die bestrafende Multiplikation nichts bewirken, da vor der Kontraktion immer Dreiecke aus nicht identischen Ebenen an der Kantenbewertung beteiligt waren.

Trotz dieser Bestrafung kommt es insbesondere zum Ende der Vereinfachungsfolge zu aufeinanderfallenden Dreiecken. Zudem werden auch nur genau identische Dreiecke bestraft. Dreiecke, die zwar aufeinander liegen, aber mindestens einen unterschiedlichen Knoten besitzen, werden durch diese Vorgehensweise nicht erfasst. Dieser letzte und seltener vorkommende Fall kann aber nicht sinnvoll behandelt werden, da ein rechenintensiver Vergleich aller veränderter Dreiecke mit allen noch im Netz befindlichen Dreiecken stattfinden müsste. Dies würde den sonst schnellen Algorithmus unverhältnismäßig langsam machen.

Dreiecksdrehung

Ein weiterer störender Punkt ist die so genannte Netzumkehrung (mesh inversion), die auch Dreiecksdrehung genannt wird. Dieses Phänomen zeigt Abbildung 3.11.

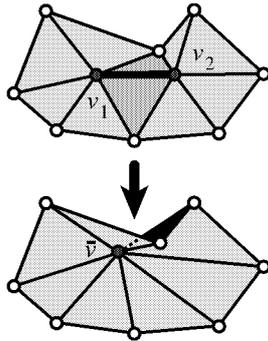


Abb. 3.11: Eine Kantenkollabierung, die eine Oberflächenfaltung verursacht

Bei der Knotenverschmelzung kann es trotz optimaler Wahl des Verschmelzungspunktes dazu kommen, dass die Oberfläche sich faltet. Die Rückseite eines Dreiecks würde sichtbar werden und die Beleuchtungsberechnung über die Normalen zu einer inversen Helligkeit kommen. Dieser Helligkeitsunterschied ist in der Abbildung 3.11 für das schwarze Dreieck gegeben. Wahrscheinlicher ist die Faltung für ein von den zu v_1 und v_2 adjazenten Knoten aufgespanntes Polygon, das nicht konvex ist. Gleichzeitig kommt es bei diesen Umkehrungen leicht zur Selbstdurchdringung des Netzes, was ebenso Artefakte verursacht.

Um solch eine Faltung ausfindig zu machen, gibt es verschiedene Ansätze. In der dieser Arbeit zugrundeliegenden Implementation werden die Dreiecksnormalen, deren Dreiecke an den zu verschmelzenden Knoten angrenzen - ausgenommen der in Abbildung 3.11 dunkel eingefärbten verschwindenden

Dreiecke -, vor und nach einer Verschmelzung untersucht. Kommt es bei mindestens einer der Normalen zu einer starken Änderung (normal flip), ist wahrscheinlich eine Faltung aufgetreten.

Wie bei der obigen Identitätsbetrachtung wird solch eine Faltung dadurch zurückgehalten, indem die angedachte Knotenverschmelzung bestraft und in die Liste der Verschmelzungskandidaten neu einsortiert wird. Hierbei ist für gewöhnlich auch ein Bestrafungsfaktor zuständig (in der Implementation wurde 1000 gewählt), der der Fehlerquadrik hinzumultipliziert wird. Es kann aber vorkommen, dass auch bei Kontraktionen mit keinem Fehler die Netzinversion auftritt. Dann muss statt der Multiplikation eine Addition einer Strafe stattfinden (in der Implementation wurde 0,001 gewählt). Kommt nach Neueinsortierung des Knotenpaares in die Abarbeitungsliste das gleiche Paar wieder zum Vorschlag und würde wieder ein Normalsprung stattfinden, würde im zweiten Schritt die Multiplikation durchgeführt werden.

Wie auch bei der Identitätsbetrachtung aus dem vorherigen Abschnitt, darf die Zurückstellung der Kandidaten nicht dazu führen, dass das Netz sich nicht komplett simplifizieren lassen kann. In der Implementation wird mit der Bestrafung für gewöhnlich bei einstellbaren 135 Grad Unterschied einer Normalenrichtung nach einer Kontraktion angefangen. Lässt sich durch die Kontraktionsverhinderung das Netz nicht weiter vereinfachen, wird die Gradzahl langsam erhöht. Ist die endgültige Vereinfachung immer noch nicht durchzuführen, wird letztendlich die Gradzahl leicht über die maximalen 180 Grad heraufgesetzt und damit die Bestrafung ausgesetzt.

Solch eine Dreiecksdrehung verursachende Kontraktion muss noch nicht einmal hohe Kosten verursachen, da der Verschmelzungspunkt nur nahe der richtigen Ebenen liegen muss. Dabei kann ein Kontraktionspunkt auf der anderen Seite der Kante landen, die einem Verschmelzungskandidaten in einem Dreieck gegenüber gelegen hat.

Als Minimalbeispiel ist in Abbildung 3.12 ein Netz mit zwei Dreiecken angegeben, die in einer Ebene liegen. Das Originalnetz aus (a) mit vier Knoten wird ohne Randbestrafung auf ein Netz mit drei Knoten reduziert. Jede Kantenkollabierung besitzt bei ausgeschalteter Randbestrafung keine Kosten, da jeder Punkt trivialerweise zu Dreiecken gehört, die in der gleichen Ebene liegen. Unter (b) bleibt die Dreiecksdrehung unberücksichtigt und die unterste Kante liegt trotz gleicher Kosten aller Kontraktionen zufällig als erstes zur Bearbeitung vor und wird auf einen Punkt reduziert. Da die Quadrik

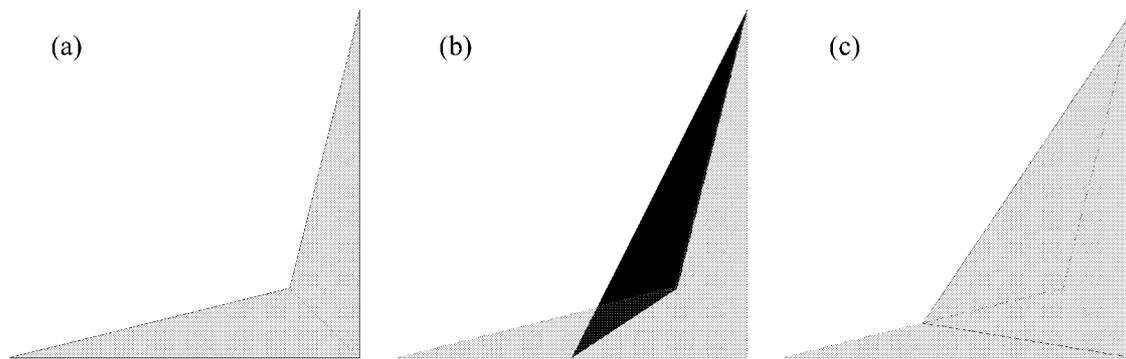


Abb. 3.12: Minimalbeispiel (a), das in (b) nicht und bei (c) doch vor einer Normalendrehung geschützt wurde

nicht invertierbar ist, werden die Eckpunkte der Kante und die Mitte untersucht. Aber auch hierbei sind die Kosten gleich Null für alle drei Möglichkeiten. Für diesen Fall wird die Mitte als Lösung genommen. Hierbei kommt es zur Drehung des Dreiecks um genau 180 Grad, und das Dreieck ist von der Rückseite zu sehen. Da die Normalenrichtung des Dreiecks die Beleuchtung bestimmt, befindet sich der Betrachter nun auf der unbeleuchteten Rückseite, von der die Normale wegzeigt. In (c) hat eine Bestrafung der Dreiecksdrehung stattgefunden und die unter (b) durchgeführte Kollabierung mit einem Straffehler versehen. Die Kollabierung der leicht schrägen langen Kanten des unteren Dreiecks erzeugt keine Netzinversion und wird zugelassen. Das Originalnetz wurde leicht transparent vor die vereinfachten Netze gelegt, um die Knotenlage nach der Kantenkontraktion einfacher zu erfassen.

Weitere Netzkonistenzen könnten überprüft und gegebenenfalls unterdrückt werden. Hierzu gehören zum Beispiel die topologische Bewahrung von Löchern oder die Verhinderung von allzu spitzen Dreiecken, die zum Beispiel bei Finiten Elementen Berechnungen störend sind. Diese letzten Fälle sind in der Implementation nicht berücksichtigt worden, da sie für die dieser Arbeit zugrundeliegende Anwendung irrelevant sind.

3.1.6 Speicherung der Simplifizierung

Bisher wurde immer davon gesprochen, dass die iterative Knotenkontraktion soweit durchgeführt wird, bis die gewünschte Vereinfachungsstufe erreicht wird oder keine Kante mehr zum Verschmelzen vorliegt. Der erste Fall braucht bei der dieser Arbeit zugrundeliegenden Implementation nicht berücksichtigt werden, da alle Kanten kollabieren, so dass nur noch Knoten und keine Dreiecke und damit Kanten mehr im Netz vorhanden sind. Die restliche Knotenanzahl hängt mit der Anzahl an Zusammenhangskomponenten des Netzes und der Verschmelzung solcher Komponenten durch den Schwellwert zusammen.

Diese rigorose Eliminierung kann deshalb durchgeführt werden, da jeder Simplifizierungsschritt, und wie er später wieder rückgängig gemacht werden kann, gespeichert wird. Die Simplifizierung wird dadurch reversibel. Diese von Hoppe eingeführte Idee basierend auf der Datenstruktur Progressive Mesh (PM, siehe Abschnitt 2.6.2) und wird genutzt, um jeden Simplifizierungsschritt zu protokollieren. Ziel der Implementation ist es, diese Datenstruktur so aufzubauen, dass sie für die Repräsentation des gesamten Netzes ausreicht und im Gegensatz zu [30] das Originalnetz überflüssig macht. In zu den Knoten linearer Zeit kann somit jede Detailstufe des Modells ohne Zugriff auf das Originalnetz direkt aus dem PM erstellt werden.

Daher ist es auch möglich, diese Datenstruktur als separates das gesamte Netz wiedergebendes Datenformat zu betrachten und ein Herausschreiben und Einlesen zu ermöglichen. Der Plattenspeicherplatz ohne Komprimierung und nur mit den nötigsten Angaben ist gegenüber der originalen Netzangabe über Knotenkoordinaten und Dreiecke aus Knotennummern nur unwesentlich größer. Werden wie bei der dieser Arbeit zugrundeliegenden Implementation weitere die Darstellung beschleunigende Angaben hinzugefügt, steigt der Platz auf das Dreifache des ursprünglichen Speicherplatzes.

Sei M_n das Originalnetz und M_0 das nicht mehr zu vereinfachende Netz, dann lässt sich dies formal so darstellen:

$$M_n \begin{array}{c} \xrightarrow{\psi_n} \\ \xleftarrow{\psi_n^{-1}} \end{array} M_{n-1} \begin{array}{c} \xrightarrow{\psi_{n-1}} \\ \xleftarrow{\psi_{n-1}^{-1}} \end{array} \cdots \begin{array}{c} \xrightarrow{\psi_2} \\ \xleftarrow{\psi_2^{-1}} \end{array} M_1 \begin{array}{c} \xrightarrow{\psi_1} \\ \xleftarrow{\psi_1^{-1}} \end{array} M_0$$

$$\psi_k(M_k) = M_{k-1} \quad \psi_k^{-1}(M_{k-1}) = M_k \quad k \in \{1, \dots, n\}$$

$$(\psi_1 \circ \dots \circ \psi_n)(M_n) = M_0 \quad (\psi_n^{-1} \circ \dots \circ \psi_1^{-1})(M_0) = M_n$$

ψ_k ist dabei die Knotenverschmelzung (vertex pair contraction) und ψ_k^{-1} die Knotenspaltung (vertex split). M_0 ist im Gegensatz zu Abschnitt 2.6.2 kein beliebiges grobes Anfangsnetz, sondern das Netz, das durch Knotenverschmelzung nicht mehr zu vereinfachen ist.

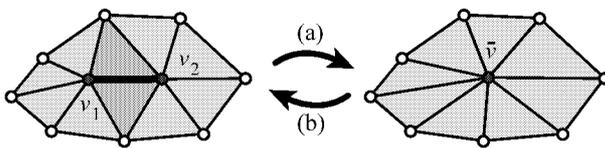


Abb. 3.13: (a) Knotenverschmelzung und (b) Knotenspaltung als inverser Vorgang

Abbildung 3.13 ist fast identisch mit Abbildung 3.1 (a). Nur der mit (b) gekennzeichnete rückwärtige Pfeil ist neu und deutet an, dass mit der PM-Datenstruktur auch eine Verfeinerung möglich ist. (a) ist die Knotenverschmelzung und (b) die Knotenspaltung. Für Abbildung 3.1 (b) wäre ein ähnliches Schaubild denkbar.

Für die Datenstruktur werden für den Schritt (a)

- Knotenindizes des Knotenpaares (v_i, v_j) (2i) und
- Verschmelzungspunktkoordinaten von \bar{v} (3f)

und für den Schritt (b)

- Punktkoordinaten der beiden Endpunkte v_i und v_j der realen oder imaginären Kante ($2 \times 3f$),
- Liste von Dreiecken, die neue Punkte enthalten (helle Dreiecke) ($\sim 4i$) und
- Liste von Dreiecken, die aufgespannt durch die Endpunkte und einen dritten Knoten neu entstehen (dunkle Dreiecke) ($\sim 2i$)

gebraucht. Die in Klammern angegebene Zahl mit folgendem “i” oder “f” gibt an, wie viele ganzzahlige oder Gleitkomma-Werte zur Speicherung der Daten nötig sind. Für n Knoten im Originalnetz sind also $8n$ ganzzahlige und $9n$ Gleitkomma-Werte für die PM-Datenstruktur nötig.

Für die Knotenverschmelzung sind die Knotenindizes zur Angabe der realen oder imaginären Kante nötig. Da bei der Verschmelzung nur ein Knoten verschwindet, der andere nur seine Position verändert und der erstangegebene Knoten des Paares immer als der sich verändernde Knoten angenommen wird, braucht für die Spaltung kein weiterer Index vorgehalten werden. Das gleiche Argument gilt auch für die Spaltung, für die die Koordinaten der Endpunkte der jeweiligen Kante vorgehalten werden müssen.

Da bei der Implementation der für das Quadrik-Fehlermaß optimale Verschmelzungspunkt berechnet wird, ist die Angabe der Koordinaten dieses Punktes für die Zusammenführung nötig. Bei der Speicherung als Punktdifferenzen genügt in C die Genauigkeit des Datentyps `float`. Durch die Speicherung als Verschiebungsvektor sind die Zahlen klein und der Zahlenraum von `float` groß genug, um die nötige Genauigkeit zu gewährleisten.

Für die Spaltung sind ferner zwei Dreieckslisten nötig. Von den hellen Dreiecken müssen sich nur die des neu hinzukommenden Knotens gemerkt werden, da sich für die anderen Dreiecke durch die Verschiebung nichts ändert. Die gleichen Dreiecke werden auch schon bei der Verschmelzung herangezogen, damit in diese der Index des anderen Knotens eingetragen werden kann. Es handelt sich dabei durchschnittlich um vier Dreiecke, da laut [6] bei einem Voronoi-Diagramm der Rand einer Voronoi-Region im Mittel sechs Kanten besitzt. Ein Knoten hat also im Mittel sechs anliegende Dreiecke. Da auch die verschwindenden Dreiecke aus der nächsten Liste dazugehören und diese durchschnittlich zwei sind, ergibt sich die ungefähre Listenlänge von vier.

Die zweite Liste ist die Liste der Dreiecke, die neu entstehen und durch den neben den Kantenknoten zusätzlichen Knoten angegeben werden können. Die Anzahl der Dreiecke in dieser Liste sind ungefähr zwei, da an einer Kante meist zwei Dreiecke angrenzen. Mehr als zwei Dreiecke können es nur in entarteten Bereichen sein, ein Dreieck an Rändern und kein Dreieck, wenn es sich nicht um eine reale Kante, sondern um die in Abbildung 3.1 (b) beschriebene kantenungebundene Knotenverschmelzung handelt.

Um die Darstellung zu beschleunigen, werden in der Implementation zu den neuen Knoten im feineren Netz die Normalen mit abgelegt, da sie dann für die Darstellung nicht mehr berechnet werden müssen. Zudem werden noch Netzeigenschaften an den Knoten abgespeichert, die in dieser Arbeit keine Rolle spielen. Für die in unserer Anwendung zu betrachtenden Modellgrößen ist der zusätzliche Speicherplatzbedarf kein Problem. Um effektiv Speicherplatz durch die Simplifizierung zu sparen, könnte ein vereinfachtes Netz nur durch die benötigte Anzahl an PM-Stufen gespeichert werden.

Bei der Speicherung als PM-Struktur braucht nicht einmal ein Grundnetz vorliegen, von dem aus verfeinert wird, da die ersten PM-Schritte die ersten Knoten enthalten und nur Kanten erzeugen ohne dabei anfänglich Dreiecke aufzuspannen. Die Anzahl dieser anfänglichen Schritte, die kein Dreieck erzeugen, kann so groß sein, wie es separate Zusammenhangskomponenten im Netz zum Ende hin gibt.

Um einen direkten Zugriff in konstanter Zeit auf die Knoten über Indizes zu erreichen, wird zur Laufzeit im Hauptspeicher parallel zum PM wie auch bei den anderen Netzstrukturen jeweils Dreiecks- und Knotenvektoren entsprechend Anhang A.2.2 angelegt, die das aktuelle Netz im Ganzen beschreiben. Diese Vektoren werden durch die Daten im PM entsprechend aufgebaut und modifiziert.

Diese PM-Datenstruktur ermöglicht es nun wie schon oben erwähnt sehr schnell zu jeder Detailstufe des Netzes durch die Liste zu laufen. Außerdem besteht durch die autarke Nutzung des PM ohne eine zusätzliche Netzdatenstruktur im Hintergrund die Möglichkeit der progressiven Übertragung gemäß Abschnitt 2.8.2.

3.1.7 Adaptive Netzanpassung

Für die dieser Arbeit zugrundeliegende Anwendung war aber statt der autarken Nutzung der Datenstruktur PM die adaptive Anpassung der Dreiecksanzahl des Netzes aufgrund von Randbedingungen viel wichtiger. Die Anwendung läuft wie in der Einleitung und im Anhang A.1 beschrieben in einem Virtual Reality-System^G unter Nutzung der Kraftrückkopplung^G. Es besteht nun durch das PM die Möglichkeit, Angaben über die gewünschte Bildwiederholrate (frame rate) in Bildern pro Sekunde (frames per second, fps) oder Kraftwiederholrate (force update rate) in Kraftberechnungen pro Sekunde (force updates per second, fus) zu machen und gemäß der erzielten gemessenen Werte das Netz feiner oder gröber einzustellen, je nachdem ob die Bild- oder Kraftberechnung schnell genug oder zu langsam war. Dies geschieht voll automatisch und wird durch ein entsprechendes Wandern innerhalb des PM realisiert.

Bei der automatischen Netzanpassung aufgrund von Randbedingungen, wertet jedes Objekt kontinuierlich in gewissem Abstand (in der Implementation wurde ein Zeitraum von 1,5 Sekunden gewählt) die Werte zu den zu berücksichtigenden Randbedingungen aus und passt sich den vom Benutzer gemachten Einstellungen an. Damit der Wert, für die eine Randbedingung angegeben ist, nicht zu stark von temporären Einflüssen abhängig ist, wird sein Durchschnitt über die seit der letzten Abfrage ermittelten Werte für die Bewertung herangezogen. Die schwankende Systemlast wird durch die Konstruktion der heutigen Rechner und Betriebssysteme als Multitasking- und Multiuser-System und der eigenen Interaktion verursacht.

Die Netzanpassung geschieht nicht in der durch die Randbedingungen vorgegebenen vollen Höhe, sondern wird schrittweise (in der Implementation wurden 30% gewählt) durchgeführt. Dies geschieht wiederum, um temporäre Lastschwankungen des Systems besser abfangen zu können und um die visuelle Veränderung kontinuierlicher aussehender zu lassen. Zwar ist die für den glatten Übergang zweier Detailstufen bestimmte Geomorph-Technik aus Abschnitt 2.8.1 implementiert worden, aber der Einsatz verlangsamt die Visualisierung so stark, dass sie nicht praktikabel ist. Diese Verlangsamung liegt daran, dass die Display-Liste^G nicht mehr sinnvoll genutzt werden kann, weil sich die Netzgeometrie fortlaufend ändert.

Die vorgegebenen Randbedingungen müssen nicht genau erzielt werden (in der Implementation wurde eine Abweichung von $\pm 25\%$ zugelassen), da sonst schon leichte Schwankungen der Systemlast oder des Betrachterstandpunktes eine Netzanpassung veranlassen würde, die einen Neuaufbau der Display-Liste nötig macht.

Sei Δt der zeitliche Abstand zweier Aktualisierungsläufe in Sekunden, r_{act} der aktuelle durchschnittliche Wert für die anzupassende Randbedingung, r_{goal} der gewünschte Wert der Randbedingung und v_{anz} die aktuelle Anzahl an Netzknoten. Des Weiteren gebe s die Geschwindigkeit der Wertanpassung an, wobei $s = 1$ eine komplette Anpassung in einem Schritt bewirken würde und in der Implementation 0,3 gewählt wurde, und c einen Toleranzfaktor für die Abweichung von Wunsch und Wirklichkeit der Randbedingung vor, der identisch Null ist, wenn keine Abweichung erlaubt ist, und in der Implementation 0,25 ist. Die Netzanpassung würde wie folgt ablaufen:

alle Δt Sekunden:

wenn $r_{act} < (1 - c) \cdot r_{goal}$:

$$\text{vereinfache: } v_{anz} := v_{anz} - s \cdot v_{anz} \cdot \frac{r_{goal} - r_{act}}{r_{goal}} \quad (3.8)$$

wenn $r_{act} > (1 + c) \cdot r_{goal}$:

$$\text{verfeinere: } v_{anz} := v_{anz} + s \cdot v_{anz} \cdot \frac{r_{act} - r_{goal}}{r_{goal}} \quad (3.9)$$

Ein Verlauf der adaptiven Netzanpassung ist in Abschnitt 4.7 angegeben.

Für konkurrierende Modelle muss sich für die gesamte Szene ein Prozentsatz, der das Verhältnis der Anzahl von Kanten im vereinfachten Netz zum Originalnetz widerspiegelt, gemerkt werden, gegen den alle Objektauflösungen streben. Ansonsten besteht die Möglichkeit, dass der Grad der Vereinfachung konkurrierender Modelle sich auf unterschiedliche Werte einpendelt. Da jedes Modell autark seine Vereinfachungsstufe anpasst, kann es dazu kommen, dass ein Modell eine Vereinfachung durchführt, das die Werte zu den Randbedingungen so stark beeinflusst, dass das nächste Modell bei der Abfrage der Werte eine Verfeinerung veranlasst. Dies kann wiederum dazu führen, dass das nächste Modell wieder entgegengesetzt handelt. Es können dadurch ungünstigerweise Modelle auf Kosten anderer wachsen. Aus diesem Grund ist die Verwaltung eines zentralen Prozentsatzes sinnvoll, der die Richtung für alle Modellvereinfachungen vorgibt. Dieser Prozentsatz wird entsprechend der Anfragen der autarken Modelle angepasst.

Nicht abgefangen werden kann damit aber die Konkurrenz, die durch unterschiedliche Randbedingungen der Modelle hervorgerufen werden. Soll zum Beispiel ein Modell so eingestellt werden, dass 20 fps erreicht werden, und ein anderes hat als Randbedingung 50 fps vorgegeben, dann wird das Modell mit der geringeren vorgeschriebenen Bildwiederholrate auf Kosten des anderen wachsen und dieses eliminieren, wenn es vorher nicht die gewünschte Rate erreicht.

Statt der Nutzung des Prozentsatzes, wäre es für Modelle, die in gewisser Abhängigkeit stehen, besser, zu einem Objekt zu verschmelzen, um gemeinsam vereinfacht zu werden, so dass das Quadrik-Fehlermaß entscheidet, was vereinfacht wird und nicht der Prozentsatz. Aufgrund des Prozentsatzes kann es nämlich sein, dass zum Beispiel auf ein sehr feines aber einfaches und damit kostengünstig zu vereinfachendes Modell Operationen durchgeführt werden, die wesentlich geringere Kosten verursachen als die Operationen auf ein anderes Modell der gleichen Szene. Mit denen im Ausblick 5.2.2 beschriebenen gewichteten Quadriken wäre es bei verschmolzenen Modellen auch gegeben, dass kleinere Objektteile mit feinerer Dreiecksaufteilung geringer gewichtet werden als große mit einer größeren Aufteilung.

Ein nicht implementierter Ansatz könnte die Modellkonkurrenz auf ganz andere Weise lösen. Hierzu ist es auch wieder wichtig, dass entsprechend Abschnitt 5.2.2 mit gewichteten Quadriken bei der Vereinfachung gearbeitet wurde, damit die Oberflächengröße in die Fehlerquadrik entsprechend einfließt. Wird sich nun für jeden Schritt des PM zusätzlich der verursachte Fehler gemerkt, könnte die Anwendung anhand des Fehlers aller PM-Listen der Szene entsprechend ihrer Fehlereinträge durchlaufen. Die Vereinfachung geschieht dann nicht mehr Modell- sondern Szenenbezogen. Dieser Ansatz ist nicht implementiert worden, da er zusätzlichen Speicherplatz kostet und mehrere gleichzeitig zu visualisierende Modelle in der Anwendung selten auftreten.

3.1.8 Zusammenfassung des Algorithmus

Die Verfahrensskizze der Vereinfachung aus Abschnitt 3.1.1 kann nun nach ausführlicher Behandlung der einzelnen Verfahrensschritte auf folgende Punkte erweitert bzw. präzisiert werden:

1. Bestimme die Menge der Knotenverschmelzungskandidaten (v_i, v_j) , für die gilt:
 - (a) (v_i, v_j) ist eine reale Kante des Netzes oder
 - (b) $\|v_i - v_j\|_2 < r$ mit r als Schwellwert und (v_i, v_j) ist keine Kante des Netzes
2. Berechne für jeden Knoten v_i seine Quadrik Q_i
 - (a) Über die angrenzenden Dreiecke und deren aufgespannte Ebene
 - (b) Gegebenenfalls über zu bestrafende offene Ränder durch gewichtete zusätzliche Quadriken senkrechter Ebenen
3. Für jeden Verschmelzungskandidaten führe folgende Punkte durch:
 - (a) Berechne die Quadrik des Punktepaars (v_i, v_j) durch $Q = Q_i + Q_j$
 - (b) Bestimme den optimalen Verschmelzungspunkt \bar{v} mit dem geringsten Fehler auf Basis des Quadrik-Fehlermaßes
4. Sortiere die Kandidatenpaare aufsteigend nach dem verursachenden Fehler $Q(\bar{v})$
5. Wiederhole folgende Punkte, bis keine Kandidaten mehr vorliegen:
 - (a) Untersuche gegebenenfalls Dreiecksidentitäten und Dreiecksdrehungen, die die aktuelle Kontraktion mit den geringsten Kosten verursacht, und stelle falls nötig das Punktepaar durch Bestrafung der Fehlerquadrik in der Abarbeitungsliste der Kandidatenpaare zurück bis Kandidaten vorliegen, die keine Netzinkonsistenz erzeugen
 - (b) Eliminiere das Knotenpaar (v_i, v_j) mit den geringsten Kosten $Q(\bar{v})$
 - (c) Führe die Verschmelzung $(v_i, v_j) \rightarrow \bar{v}$ durch und weise dem verbleibendem Punkt die Quadrik $Q_i + Q_j$ zu
 - (d) Merken des Simplifikationsschrittes durch Eintrag in die kontinuierliche Netzrepräsentation PM
 - (e) Aktualisiere den Verschmelzungspunkt und die Kosten aller noch bestehenden Verschmelzungskandidaten gemäß Punkt (3), die einen Knoten des aktuellen Knotenpaares beinhalten; diese Paare müssen neu in die Abarbeitungsliste einsortiert werden
6. Stelle aufgrund von Randbedingungen über die Netzrepräsentation PM die gewünschte Detailstufe des Netzes adaptiv ein

Nachdem nun das Verfahren beschrieben wurde, soll in den folgenden Abschnitten die Qualitätsbewertung der resultierenden Netze besprochen und das Wesen des Quadrik-Fehlermaßes genauer untersucht werden. In den daran anschließenden Kapiteln werden die Ergebnisse der Implementation erörtert und ein Ausblick auf noch zu verbessernde oder zu erweiternde Aspekte der Vereinfachung gegeben.

3.2 Eigenschaften des Quadrik-Fehlermaßes

In diesem Abschnitt wird das Quadrik-Fehlermaß genauer untersucht. Andere Fehlermaße wurden bereits kurz in Abschnitt 2.7.3 vorgestellt. Zur Einführung und Motivation sei hier empfohlen, die Abschnitte 3.1.3 und 3.1.4 vor der mathematischen Betrachtung des Fehlermaßes zu lesen. Eingeführt wurde das Quadrik-Fehlermaß über die Summe der Quadratabstände eines Punktes zu angrenzenden Ebenen. Es soll hier unter anderem der Zusammenhang der Quadrik mit der lokalen Oberflächenkrümmung gezeigt werden.

Der zweidimensionale Fall, der in dieser Arbeit häufiger zur Anschauung bemüht wurde, wird in [75, 79] ausführlich behandelt. Die in den Beispielen vorkommenden Geradenpaare und Ellipsen kommen aus der Klasse der Kegelschnitte, die auch die Parabeln und Hyperbeln beinhalten. In [75] werden auch Quadriken behandelt, die eine höhere Dimension als der Raum \mathbb{R}^3 besitzen und damit Hyperflächen zweiten Grades bzw. Ordnung darstellen. In diesem Abschnitt beschränken wir uns auf die für diese Arbeit wichtigen dreidimensionalen Quadriken.

Implizite polynomiale Funktionen beschreiben algebraische Flächen. Ist die Summe der Exponenten in den Unbekannten der polynomialen Funktion maximal zwei, so spricht man von einer algebraischen Fläche zweiter Ordnung, einem impliziten quadratischen Polynom, einer Hyperfläche zweiter Ordnung oder einer Quadrik.

Seien x_i , $i \in \mathbb{N}_n$ die Unbestimmten, dann lautet das allgemeine quadratische Polynom

$$f(x_1, \dots, x_n) = \sum_{1 \leq i, j \leq n} a_{ij} x_i x_j + \sum_{i=1}^n a_i x_i + a.$$

Im dreidimensionalen Fall, der dieser Arbeit als Fehlermaß zugrunde liegt, werden Quadriken ganz allgemein durch die implizite Flächengleichung der Form

$$f(x, y, z) = q_{11}x^2 + 2q_{12}xy + 2q_{13}xz + 2q_{14}x + q_{22}y^2 + 2q_{23}yz + 2q_{24}y + q_{33}z^2 + 2q_{34}z + q_{44} = 0$$

beschrieben [48]. Dies entspricht der Gleichung (Bezeichnungen gemäß Gleichung (3.2))

$$\begin{aligned} f(x, y, z) &= (x, y, z) \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + 2(q_{14}, q_{24}, q_{34}) \begin{pmatrix} x \\ y \\ z \end{pmatrix} + q_{44} \\ &= f(v) = v^t A v + 2B^t v + C = 0. \end{aligned}$$

In der homogenen Darstellung lässt sich dies auch schreiben als

$$\begin{aligned} f(x, y, z) &= (x, y, z, 1) \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \\ &= \tilde{v}^t Q \tilde{v} = 0. \end{aligned} \tag{3.10}$$

Q und A sind dabei per Konstruktion symmetrisch. Wäre in Gleichung (3.10) eine Matrix Q vorgegeben, die nicht symmetrisch ist, ließe sich die Gleichung so umschreiben, dass Q symmetrisch wird:

$$\tilde{v}^t Q \tilde{v} = 0 \quad \Leftrightarrow \quad \tilde{v}^t \frac{1}{2} (Q + Q^t) \tilde{v} = 0.$$

Es kann also im Folgenden o.B.d.A. davon ausgegangen werden, dass Q und damit A symmetrisch sind.

Die Gleichung $Q(\tilde{v}) = \tilde{v}^t Q \tilde{v}$, die aus der impliziten Flächengleichung eine Abbildung $Q : \mathbb{R}^4 \rightarrow \mathbb{R}$ macht, ist eine reelle quadratische Form, die zur reellen symmetrischen Bilinearform

$$\beta(\tilde{v}, \tilde{v}) = \tilde{v}^t Q \tilde{v} = Q(\tilde{v})$$

gehört. Die Bilinearform ist zwar in jeder Komponente linear, nicht aber die quadratische Form.

Die Gleichung (3.10) beinhaltet Kugeln, Ellipsoide, Hyperboloide, Paraboloid, Kegel und Zylinder, aber auch die leere Menge. Für singuläre A kommen noch Objekte niedrigerer Ordnung wie Ebenen, Geraden und Punkte hinzu. Gilt zum Beispiel $q_{11} = q_{22} = q_{33} = -q_{44} = 1$ und haben die restlichen Koeffizienten den Wert Null, so beschreibt die Gleichung eine Einheitskugel um den Ursprung. Ist die Matrix A identisch Null, so entsteht für B ungleich Null eine Ebene.

In der Computergrafik besitzen Quadriken eine Bedeutung in Spezialanwendungen wie der Molekül- oder Festkörpermodellierung. Quadriken haben für die Computergrafik den Vorzug einer einfachen Flächennormalenberechnung $((df/dx, df/dy, df/dz)^t)$, einem einfachen Test, ob ein Punkt innerhalb, außerhalb oder auf der Fläche liegt und einer einfachen Berechnung von z zu gegebenen x und y zum Beispiel für den Tiefenpuffer-Algorithmus^G. Des Weiteren lassen sich die Schnittmenge zweier Flächen leicht bestimmen und eine affine oder projektive Abbildung auf die Quadriken einfach durchführen [48].

In dieser Arbeit werden die Quadriken aber nicht zur Modellierung, sondern zur Definition eines Fehlermaßes herangezogen. Trotzdem kann die Möglichkeit der einfachen Transformation der Quadrik auch hier von Nutzen sein. Wird auf ein Modell eine lineare Transformation im homogenen Sinne (affine Abbildung der Skalierung, Scherung, Rotation und Translation oder projektive Abbildung) durchgeführt, so können die zuvor berechneten Quadriken ebenso transformiert werden. Für die affinen Abbildungen bleibt für ein einzelnes Modell die Simplifizierung dieselbe, da die Verhältnisse von Längen, Ebenen und Volumina unverändert bleiben [50, 51].

Stehen aber mehrere unterschiedlich transformierte Modelle in Konkurrenz, die gleichzeitig simplifiziert werden sollen, kann sich die Abarbeitungsreihenfolge bei der Vereinfachung ändern. Wird also zum Beispiel von zwei gleichen Modellen in einer Szene eines kleiner skaliert, wird sich auch die Quadrik entsprechend verkleinern und der Fehler verringern. Das kleinere Modell wird somit vorrangig vereinfacht, was auch wegen der geringeren visuellen Relevanz Sinn macht.

Wird das Modell durch die reguläre Abbildung T transformiert und dabei der Punkt v auf \hat{v} abgebildet ($\hat{v} = Tv$), so kann eine Bewertung durch Rücktransformation geschehen:

$$Q(v) = (T^{-1}\hat{v})^t Q(T^{-1}\hat{v}),$$

was wie folgt umgeschrieben werden kann:

$$Q(v) = \hat{v}^t ((T^{-1})^t Q(T^{-1})) \hat{v}.$$

Statt einer Rücktransformation der Knoten können also alle Quadriken durch die Abbildung $Q \rightarrow (T^{-1})^t Q T^{-1}$ angepasst werden [79].

3.2.1 Konstruktion

Das Quadrik-Fehlermaß ist über die Summe der Quadratabstände eines Punktes zu angrenzenden Ebenen eingeführt worden. Es gibt aber auch andere Möglichkeiten der Konstruktion des Quadrik-Fehlermaßes. Es lässt sich über eine lokale Volumendifferenz herleiten oder das Minimum über die Methode der “kleinsten Fehlerquadratsumme” bestimmen.

Volumendifferenz

Das Verfahren der lokalen Volumendifferenz, das von Lindstrom und Turk eingeführt wurde [28], lässt sich auch auf das Quadrik-Fehlermaß abbilden. Die Volumendifferenzmethode versucht das Volumen zwischen dem Originalnetz und der Netzapproximation möglichst gering zu halten.

Sei T ein Dreieck (u_1, u_2, v_i) , das über eine Knotenverschmelzung $(v_i, v_j) \rightarrow \bar{v}$ gemäß Abbildung 3.14 verändert wird. Bei der Verschiebung wird ein Tetraeder (u_1, u_2, v_i, \bar{v}) aufgespannt, dessen Volumen auch das Volumen zwischen den beiden Simplifizierungsstufen beschreibt, wenn alle sich verändernden Dreiecke hinzugezogen werden. Das Gesamtvolumen stellt somit ein Maß für die Änderung des Netzes dar.

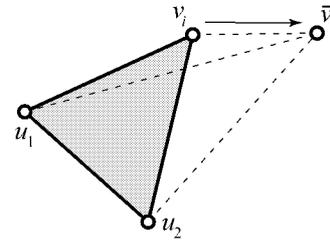


Abb. 3.14: v_i wird zu \bar{v} verschoben und beschreibt dabei ein Tetraeder (u_1, u_2, v_i, \bar{v})

Das Volumen des Tetraeders kann leicht über die nicht normalisierte Flächennormale m des Dreiecks angegeben werden. Die Normale lautet

$$m = (u_1 - v_i) \times (u_2 - v_i)$$

und damit die Fläche g des Dreiecks

$$g = \frac{1}{2} \|m\|_2.$$

Der minimale also senkrechte Abstand h von \bar{v} zur Dreiecksebene ist über die Projektion auf den Normalenvektor ausgedrückt

$$h = (\bar{v} - v_i)^t \frac{m}{\|m\|_2}$$

und somit das Volumen V des Tetraeders

$$V = \frac{1}{3} gh = \frac{1}{6} (\bar{v} - v_i)^t m.$$

Entsprechend dem Quadratabstand untersuchen wir nun das quadratische Volumen des Tetraeders

$$\begin{aligned} V^2 &= \frac{1}{36} ((\bar{v} - v_i)^t m) ((\bar{v} - v_i)^t m) \\ &= \frac{1}{36} (\bar{v} - v_i)^t (mm^t) (\bar{v} - v_i)^t \\ &= \frac{1}{36} (\bar{v}^t mm^t \bar{v} - v_i^t mm^t \bar{v} - \bar{v}^t mm^t v_i + v_i^t mm^t v_i) \\ &= \frac{\|m\|_2^2}{36} (\bar{v}^t nn^t \bar{v} - 2\bar{v}^t nn^t v_i + v_i^t nn^t v_i) \quad (\text{da } mm^t \text{ symmetrisch}) \\ &= \frac{\|m\|_2^2}{36} (\bar{v}^t A \bar{v} + 2B^t \bar{v} + C), \end{aligned} \tag{3.11}$$

wobei mit n als normierte Flächennormale

$$\begin{aligned} A &= \frac{mm^t}{\|m\|_2^2} = nn^t, \\ B &= -nn^t v_i = dn, \quad (\text{da Abstand vom Nullpunkt } d = -n^t v_i) \\ C &= v_i^t nn^t v_i = d^2. \end{aligned}$$

Das Ergebnis in Gleichung (3.11) entspricht der Gleichung (3.1), die durch einen Faktor

$$\frac{\|m\|_2^2}{36} = \frac{g^2}{9}$$

als Gewicht für die Dreiecksfläche erweitert wurde. Hiermit ist also die Ähnlichkeit der beiden Abstandsmaße für Netze gezeigt. Das gewichtete Quadrik-Fehlermaß entspricht der Summe der quadratischen Volumina zwischen zwei Netzapproximationen. Bei der fortschreitenden Vereinfachung entsprechen die Quadriken aber nicht mehr den Volumina, da die gewerteten Tetraeder sich überlappen.

Da eine Gewichtung der Quadriken durchaus Sinn macht, wird in Abschnitt 5.2.2 nochmals darauf eingegangen; denn für ungleichmäßige Dreiecksverteilungen kann eine Gewichtung auch über die Dreiecksgröße wie hier Sinn machen.

Kleinste Fehlerquadratsumme

Das Quadrik-Fehlermaß soll die Knotenkontraktion bewerten und zudem den für dieses Maß optimalen Verschmelzungspunkt bestimmen helfen. Die Bewertung findet gemäß der Konstruktion der Quadrik über die Summe von Quadratabstände statt. Dies legt die Vermutung nahe, dass die Optimierung oder besser Minimierung dieser Summe ein ähnliches Ergebnis liefert wie die Methode der "kleinsten Fehlerquadratsumme" (least squares method) überbestimmter linearer Gleichungssysteme [8, 76].

Soll der Quadratabstand der Ebenen zu einem Punkt entsprechend des linearen Quadratmittelproblems gemäß Gauß gelöst werden, so lautet das überbestimmte Gleichungssystem

$$\underbrace{\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \vdots & & \\ a_k & b_k & c_k \end{pmatrix}}_{=: N} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = - \underbrace{\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{pmatrix}}_{=: D}, \quad (3.12)$$

wenn die k Ebenen wie zuvor durch $E_i : n_i^t w + d_i = 0$, $w \in \mathbb{R}^3$, $i \in \mathbb{N}_k$ mit $n_i = (a_i, b_i, c_i)^t$, $\|n_i\|_2 = 1$ als normierte Normalenvektoren gegeben sind. Dieses inhomogene Gleichungssystem ist für sich nicht mindestens in einem gemeinsamen Punkt schneidende Ebenen unlösbar und damit überbestimmt. Dies ist für gewöhnlich bei mehr als drei Ebenen der Fall, wenn es sich um ein nicht künstlich konstruiertes Netz handelt.

Der entsprechende Fehler $r = Nv + D$ soll bei der Methode der "kleinsten Fehlerquadratsumme" im Quadrat minimiert werden, also $r^t r$ minimal sein. Wegen der Orthogonalität des Approximationsfehlers zum Bildraum [8] muss der Restvektor r orthogonal zu allen Spalten von N sein. Im Spalten-

bzw. Bildraum liegt Nv dann nahestmöglich an $-D$. Dies führt über die Gauß-Transformation zum System der Normalgleichungen

$$N^t r = N^t N v + N^t D = 0. \quad (3.13)$$

$N^t N$ ist symmetrisch und positiv semidefinit. Hat N den Maximalrang drei, liegen also mindestens drei Ebenen paarweise nicht parallel zueinander bzw. sind mindestens drei Normalenvektoren linear unabhängig, dann ist $N^t N$ auch positiv definit und hat damit eine eindeutige Lösung. Die Lösung lautet dann

$$v = - \underbrace{(N^t N)^{-1} N^t D}_{=: I}$$

$I \in \mathbb{R}^{3 \times k}$ ist in diesem Fall die Pseudo-Inverse. Diese entspricht bei genau drei linear unabhängigen Normalenvektoren der Inversen von N . Für weniger als drei linear unabhängige Normalenvektoren kann $N^t N$ nicht invertiert werden, und es ergeben sich aus der Gleichung (3.12) Lösungsmannigfaltigkeiten entsprechend der Lösung des Quadrik-Fehlermaßes mit singulären Quadriken (siehe Abschnitt 3.1.4).

Statt des Ansatzes über die Normalgleichungen würde man bei der Lösung dieses linearen Quadratmittelproblems aus Gründen der numerischen Stabilität zum Beispiel die QR-Zerlegung mittels Householder-Transformation anwenden [76]. Der hier gewählte Ansatz zeigt aber gut die Identität der Minimierung des Quadrik-Fehlermaßes mit der Bestimmung des Punktes mit der kleinsten Fehlerquadratsumme. $N^t N$ und $N^t D$ lassen sich nämlich auch wie folgt schreiben:

$$N^t N = \sum_{i=1}^k n_i n_i^t = A$$

$$N^t D = \sum_{i=1}^k d_i n_i = B.$$

Substituiert man dieses in die Normalgleichungen (3.13), ergibt sich

$$N^t N v + N^t D = 0$$

$$\Leftrightarrow Av + B = 0,$$

was gemäß Gleichung (3.7) exakt die Gleichung zur Bestimmung des optimalen Verschmelzungspunktes des Quadrik-Fehlermaßes ist.

Bei dem Quadrik-Fehlermaß werden bei der Knotenschmelzung durch Addition der Quadriken teilweise Ebenen mehrfach bewertet. Dies führt zu einer stärkeren Gewichtung dieser Ebenen und einem Verschmelzungspunkt, der näher an diese Ebenen heranrückt. Das gleiche Resultat erhält man auch für die Methode der "kleinsten Fehlerquadratsumme", wenn auch hier die Ebenen mehrfach in das überbestimmte Gleichungssystem aufgenommen werden und somit in die Normalgleichungen einfließen.

3.2.2 Definitheit

Symmetrie, Definition:

Eine Matrix $M \in \mathbb{R}^{n \times n}$, $n \in \mathbb{N}$ ist symmetrisch, wenn

$$M = M^t.$$

Symmetrie, Eigenschaften:

Ist M symmetrisch, so gilt:

- M ist normal:

$$MM^t = M^tM$$

- M besitzt reelle Eigenwerte bzw. das charakteristische Polynom von M zerfällt
- es existiert eine Orthonormalbasis aus Eigenvektoren von M
- M ist zu einer Diagonalmatrix ähnlich, es existiert also eine Orthogonalmatrix T ($TT^t = E$, Bezeichnung: $T \in O(n)$), mit der

$$TMT^{-1} = TMT^t$$

eine Diagonalmatrix ist

Positive Definitheit, Definition [78]:

Eine Matrix $M \in \mathbb{R}^{n \times n}$, $n \in \mathbb{N}$ ist positiv definit (positiv semidefinit), wenn

$$x^t M x \underset{(\geq)}{>} 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\},$$

und wird bezeichnet mit

$$M \underset{(\geq)}{>} 0.$$

Die Menge der symmetrisch und positiv definiten Matrizen wird mit *SPD* bezeichnet.

Positive Definitheit, Eigenschaften [78]:

Sei M positiv definit (positiv semidefinit), dann hat M folgende Eigenschaften:

- M hat echt positive (nicht negative) Diagonalelemente
- M hat echt positive (nicht negative) Eigenwerte λ_i , $i \in \mathbb{N}_n$
- $\det(M) \underset{(\geq)}{>} 0$, da $\det(M) = \prod_{i=1}^n \lambda_i$
- es existiert eine Zerlegung $M = CC^t$ mit $C \in Gl(n, \mathbb{R})$

Zudem ist eine positiv definite Matrix M regulär (Bezeichnung: $M \in Gl(n, \mathbb{R})$) bzw. hat vollen Rang n und ist damit invertierbar.

Ist M positiv semidefinit und nicht positiv definit, hat M mindestens einen Eigenwert Null, also die Determinante Null und ist damit singular bzw. besitzt einen Rang kleiner n .

Ist die positiv definite Matrix auch symmetrisch, dann liegt das betragsmäßig größte Element immer auf der Diagonalen.

Positive Semidefinitheit

Per Konstruktion über Quadratabstände ist die Abbildung $Q(\tilde{v})$ nie negativ, so dass schnell die positive Semidefinitheit der Matrix Q daraus geschlossen werden könnte. Dies geht aber nicht daraus hervor, da die Aussage

$$Q(\tilde{v}) \geq 0$$

nur für $\tilde{v} = (x, y, z, 1)^t$ mit $x, y, z \in \mathbb{R}$ gilt. Für die positive Semidefinitheit muss aber $Q(\dot{v}) \geq 0$ für alle $\dot{v} = (x, y, z, w)^t \in \mathbb{R}^4$ gelten.

Es soll nun gezeigt werden, dass Q dennoch positiv semidefinit ist.

Beweis:

$w \neq 0$:

Sei $w \neq 0$, so gilt

$$Q(\dot{v}) = (x, y, z, w)Q \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \underbrace{w^2}_{\geq 0} \underbrace{\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}, 1 \right) Q \begin{pmatrix} x/w \\ y/w \\ z/w \\ 1 \end{pmatrix}}_{\geq 0} \geq 0. \quad (3.14)$$

Damit ist gezeigt, dass für $w \neq 0$ die Abbildung $Q(\dot{v})$ nicht negativ ist.

$w = 0$:

Für $w = 0$ wird die letzte Zeile und Spalte von Q nicht berücksichtigt, und es gilt

$$\begin{aligned} Q(\dot{v}) &= \dot{v}^t Q \dot{v} = v^t A v \\ &= v^t \left(\sum_i n_i n_i^t \right) v = \sum_i (n_i^t v)^2 \geq 0. \end{aligned} \quad (3.15)$$

$i \in \mathbb{N}$ läuft über die Menge der Ebenen, die in die Matrix A eingeflossen sind. Ein anschaulicher Beweis für die Semidefinitheit von A ist im Anhang B.1 zu finden.

Aus den Gleichungen (3.14) für $w \neq 0$ und (3.15) für $w = 0$ folgt, dass

$$Q(\dot{v}) \geq 0 \quad \forall \dot{v} \in \mathbb{R}^4$$

und somit die positive Semidefinitheit von Q . ■

Positive Definitheit

Der oben geführte Beweis zur positiven Semidefinitheit gilt aber nicht entsprechend für ein echt positives Fehlermaß mit

$$Q(\tilde{v}) > 0.$$

Aus dieser Ungleichung folgt nicht, dass Q positiv definit und damit regulär ist. Gleichung (3.14) kann zwar auch hier entsprechend angewandt werden,

$$Q(\dot{v}) = (x, y, z, w)Q \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \underbrace{w^2}_{\geq 0} \underbrace{\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}, 1 \right)Q}_{> 0} \begin{pmatrix} x/w \\ y/w \\ z/w \\ 1 \end{pmatrix} > 0, \quad (3.16)$$

aber die Beweisführung für $w = 0$ nutzt nur die generelle Nichtnegativität der Abstandsquadrate aus und kann somit auch für echt positive Fehlermaße nur die positive Semidefinitheit beweisen.

Beispiel:

Ein Beispiel zeigt, dass ein echt positives Fehlermaß positiv semidefinite Quadriken besitzen kann:

Seien vier Ebenen

$$E_1 : x = 0, \quad E_2 : y = 0, \quad E_3 : x - 1 = 0, \quad E_4 : y - 1 = 0$$

gegeben. Die Summe ihrer Quadratabstände ist

$$Q(\tilde{v}) = x^2 + y^2 + (x - 1)^2 + (y - 1)^2 \quad (3.17)$$

$$\begin{aligned} &= \tilde{v}^t \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \tilde{v} \\ &= \tilde{v}^t \begin{pmatrix} 2 & 0 & 0 & -1 \\ 0 & 2 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix} \tilde{v}. \end{aligned} \quad (3.18)$$

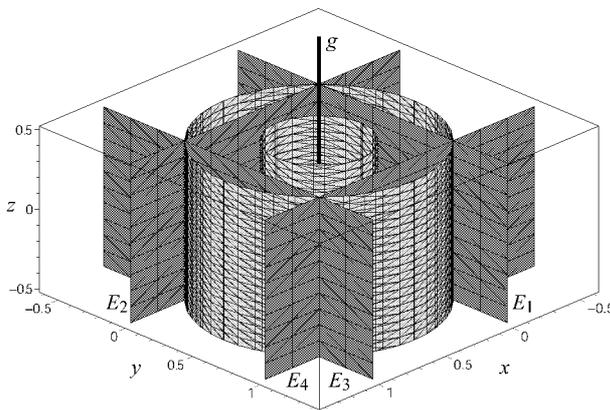


Abb. 3.15: Gerade g als Minimum des Quadrik-Fehlermaßes der Ebenen E_1 bis E_4 und deren entarteten zylindrischen Isoflächen

Aus der Grafik 3.15, die zeigt, dass die Ebenen sich nicht schneiden und somit kein Punkt des Raumes zu allen Ebenen die Entfernung Null besitzt, und aus der Tatsache, dass die ersten beiden Summanden der Gleichung (3.17) nur für $x = y = 0$ und die letzten beiden Summanden nur für $x = y = 1$ Null sind, folgt, dass $Q(\tilde{v}) > 0$ ist. Diese Tatsache suggeriert, dass Q positiv definit ist. Aber es gilt zum Beispiel $Q((0, 0, 1, 0)^t) = 0$, und somit ist Q nur positiv semidefinit. Dieses Beispiel zeigt, dass aus $Q(\tilde{v}) > 0$ nicht $Q > 0$ folgt. Selbstverständlich gilt aber die Umkehrung: Ist Q positiv definit, so ist $Q(\tilde{v}) > 0$. In unserem Beispiel zeigen auch die Eigenwerte $\{0, 2, 2 + \sqrt{2}, 2 - \sqrt{2}\}$,

dass Q nicht positiv definit, sondern semidefinit ist, da auch 0 Eigenwert ist.

Wegen des Eigenwertes 0 ist Q singulär. Anhand der Nullzeile und -spalte in Gleichung (3.18) ist zu erkennen, dass sogar die Untermatrix A singulär, also die Lösung nicht eindeutig ist. Dies ist auch ein Indiz dafür, dass Q nicht positiv definit ist, da sonst Q regulär und damit invertierbar wäre.

$$Q > 0 \Leftrightarrow Q \in Gl(4, \mathbb{R})$$

Ein letztes Indiz dafür, dass Q nicht positiv definit sein kann, ist ein nicht echt positives Diagonalelement, was eine notwendige Bedingung für positiv definite Matrizen ist.

Die Lösungsmannigfaltigkeit ist die Gerade

$$g : \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \lambda \in \mathbb{R} \quad (3.19)$$

im Zentrum der vier Ebenen und den Zylindern, die als Isofläche aus $Q(\tilde{v}) = \epsilon$ hervorgehen. Das auf der Geraden identische Minimum lautet zum Beispiel für den Aufpunkt der Geraden

$$Q\left(\frac{1}{2}, \frac{1}{2}, 0\right)^t = 1.$$

□

Für die positive Definitheit von Q muss allgemein $Q(\tilde{v}) > 0$ gelten und zusätzlich A und damit Q regulär sein:

$$Q(\tilde{v}) > 0 \quad \wedge \quad A \in Gl(3, \mathbb{R}) \quad \Leftrightarrow \quad Q > 0.$$

Beweis:

$w = 0$:

Die Regularität der Matrix A bewirkt, dass auch für $w = 0$ die Ungleichung

$$Q(\dot{v}) > 0 \quad \forall \dot{v} \in \mathbb{R}^4 \setminus \{0\}$$

gilt.

$w \neq 0$:

Für $w \neq 0$ ist der Beweis schon durch die Ungleichung (3.16) erfolgt. Ist die Komponente w Null, geht $Q(\dot{v})$ über in $v^t A v$. Da A gemäß des obigen Beweises zur Semidefinitheit eine Quadratsumme darstellt, ist $v^t A v \geq 0$. Die Regularität von A schließt Null als Eigenwert aus, so dass alle Eigenwerte echt positiv sind und damit $v^t A v > 0$ und A positiv definit ist. ■

Singularität und deren Fortpflanzung

Der Fall $Q(\tilde{v}) = 0$ liegt vor, wenn es mindestens einen gemeinsamen Schnittpunkt der Ebenen gibt. Dies ist zum Beispiel dann der Fall, wenn eine singuläre Quadrik Nullkosten, also den Fehler Null, an mehr als einem Punkt besitzt, zum Beispiel an der Schnittgeraden zweier sich schneidender Ebenen.

Drei oder weniger Ebenen erzeugen immer eine singuläre Quadrik und zwei oder weniger eine singuläre Matrix A . Präziser sind folgende Aussagen äquivalent:

- A ist singulär
- A ist nicht positiv definit bzw. mindestens ein Eigenwert ist Null
- die Ebenennormalen liegen alle in einer Ebene bzw. weniger als drei Ebenennormalen sind linear unabhängig:

$$\dim(\text{span}(\{n_i, i \in \mathbb{N}_k\})) < 3$$

- die Ebenen sind parallel zu einer Geraden
- der Lösungsraum des Minimums ist mindestens eindimensional

Bei Daten der Anwendung, die dieser Arbeit zugrunde liegt, ist Q für gewöhnlich positiv definit und damit $Q(\tilde{v}) > 0$. Auch andere gemessene Flächen, zum Beispiel Netze, die aus einem 3D-Scan hervorgegangen sind, besitzen meist positiv definite Fehlerquadriken. Dies liegt an folgenden zwei Eigenschaften der zu einem Knoten korrespondierenden Ebenen.

1. Ist zum einen die Bestrafung der Ränder aktiviert, liegen an jedem Knoten bei nicht entarteten Dreiecken oder Volumina mindestens drei Dreiecke an. Bei einer Kontraktion werden also zur Kostenbestimmung mindestens vier unterschiedliche Ebenen herangezogen. Zwei gemeinsame Ebenen entfallen bei einer Kantenkontraktion.
2. Zum anderen ist bei gemessenen Werten die Wahrscheinlichkeit sehr gering, dass diese vier Ebenen parallel zueinander sind, deren Normalen in einer Ebene liegen oder sich alle Ebenen in einem Punkt schneiden. Die zu bewertenden Ebenen liegen im Netz nicht an einem gemeinsamen Knoten.

Mindestens vier zufällig zueinander ausgerichtete Flächen bewirken die positive Definitheit.

Ist eine Quadrik an einem Punkt einmal positiv definit bzw. regulär, sind alle daraus resultierenden Quadriken ebenfalls positiv definit bzw. regulär, können also im weiteren Verlauf des Algorithmus nicht singulär werden, so dass bei der fortschreitenden Knotenverschmelzung die Wahrscheinlichkeit von singulären Quadriken sinkt. Die Summe $Q_i + Q_j$ zur Kostenbestimmung einer Kontraktion ist also nur singulär, wenn Q_i und Q_j singulär sind. Das Entsprechende gilt auch für die Untermatrix A . Ist sie einmal positiv definit und damit invertierbar, bleibt sie es auch:

$$Q_i > 0 \quad \Rightarrow \quad v^t(Q_i + Q_j)v = \underbrace{v^t Q_i v}_{>0} + \underbrace{v^t Q_j v}_{\geq 0} > 0 \quad \Rightarrow \quad Q_i + Q_j > 0.$$

Eher bei konstruierten Flächen kann es vorkommen, dass alle Flächen, die an einem Knotenverschmelzungspaar anliegen, sich mindestens in einem Punkt schneiden oder die Flächennormalen in einer Ebene liegen. Dies führt dann zu positiv semidefiniten Quadriken, bei denen $Q(\tilde{v})$ meist Null annimmt.

In sehr seltenen Fällen kann es aber zu einem Rückfall in die Beinahe-Singularität kommen. Wird zu einer regulären Quadrik eine singuläre Quadrik sehr häufig hinzugefügt, so dehnt sich die reguläre Quadrik in die Richtung immer weiter aus, in der die singuläre Quadrik entartet ist. Die Ausdehnung kann durch eine fortwährende Addition beliebig groß werden. In der Implementation wird zur

Bestimmung der Singularität die Determinante bemüht, so dass ein Rückfall während des Programm-
laufs nicht auftreten kann, da für eine Matrix A mit echt positiven Eigenwerten und eine Matrix B mit
nicht negativen Eigenwerten gilt:

$$\det(A + kB) \geq \det(A), \quad k \in \mathbb{N} \text{ endlich.}$$

Beweis:

Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch positiv definit und $B \in \mathbb{R}^{n \times n}$ symmetrisch positiv semidefi-
nit. Da es für $A \in SPD$ eine Orthogonalmatrix $U \in O(n)$ gibt, für die

$$A = U \text{diag}(\lambda_1, \dots, \lambda_n) U^t$$

ist, wobei $\lambda_i > 0$, $i \in \mathbb{N}_n$ die Eigenwerte von A sind, existiert auch die Wurzel von A
mit

$$A^{\frac{1}{2}} = U \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n}) U^t,$$

da

$$\begin{aligned} A^{\frac{1}{2}} A^{\frac{1}{2}} &= U \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n}) \underbrace{U^t U}_E \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n}) U^t \\ &= U \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n}) \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n}) U^t \\ &= U \text{diag}(\lambda_1, \dots, \lambda_n) U^t = A \end{aligned}$$

und die Inverse von A mit

$$A^{-1} = U \text{diag} \left(\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_n} \right) U^t,$$

da

$$\begin{aligned} AA^{-1} &= U \text{diag}(\lambda_1, \dots, \lambda_n) \underbrace{U^t U}_E \text{diag} \left(\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_n} \right) U^t \\ &= U \text{diag}(\lambda_1, \dots, \lambda_n) \text{diag} \left(\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_n} \right) U^t \\ &= U \text{diag}(1, \dots, 1) U^t = E. \end{aligned}$$

Zusammen ist die Wurzel der Inversen

$$A^{-\frac{1}{2}} = U \text{diag} \left(\frac{1}{\sqrt{\lambda_1}}, \dots, \frac{1}{\sqrt{\lambda_n}} \right) U^t$$

und $A^{-\frac{1}{2}} \in SPD$, und es gilt:

$$\begin{aligned}
& \det(A + kB) \geq \det(A) \\
& \Leftrightarrow \det(A^{-\frac{1}{2}}) \det(A + kB) \det(A^{-\frac{1}{2}}) \geq \det(A^{-\frac{1}{2}}) \det(A) \det(A^{-\frac{1}{2}}) \\
& \Leftrightarrow \det(A^{-\frac{1}{2}}AA^{-\frac{1}{2}} + A^{-\frac{1}{2}}(kB)A^{-\frac{1}{2}}) \geq \det(A^{-\frac{1}{2}}AA^{-\frac{1}{2}}) \\
& \Leftrightarrow \det(E + \underbrace{kA^{-\frac{1}{2}}BA^{-\frac{1}{2}}}_{=: C}) \geq \det(E) \\
& \stackrel{(*)}{\Leftrightarrow} \det(U_C U_C^t + U_C D_C U_C^t) \geq 1 \\
& \Leftrightarrow \det(U_C (E + D_C) U_C^t) \geq 1 \\
& \Leftrightarrow \det(U_C) \det(E + D_C) \det(U_C^t) \geq 1 \\
& \Leftrightarrow 1 \cdot \prod_{i=1}^n (1 + \lambda_i) \cdot 1 \geq 1.
\end{aligned}$$

Dies ist wahr, da C positiv semidefinit ist und damit die Eigenwerte $\mu_i \geq 0, i \in \mathbb{N}_n$ sind:

$$x^t C x = k x^t A^{-\frac{1}{2}} B A^{-\frac{1}{2}} x = k (A^{-\frac{1}{2}} x)^t B (A^{-\frac{1}{2}} x) \geq 0, \quad \forall x \in \mathbb{R}^n$$

da B positiv semidefinit.

(*) ist die Hauptachsentransformation, da $A^{-\frac{1}{2}}$ und B und damit C reell symmetrisch sind:

$$C = U_C D_C U_C^t$$

mit U_C Orthogonalmatrix aus Eigenvektoren von C und D_C Diagonalmatrix aus Eigenwerten $\mu_i, i \in \mathbb{N}_n$ von C .

3.2.3 Hauptachsentransformation

Transformation im \mathbb{R}^4

Wegen der Symmetrie von Q zerfällt das charakteristische Polynom, somit sind alle Eigenwerte reell, und Q ist normal [80]. Da Q normal ist und die Eigenwerte reell sind, besitzt Q Eigenvektoren, die eine Orthonormalbasis des \mathbb{R}^4 aufspannen. Das gleiche gilt für A . Wegen der positiven Semidefinitheit von Q sind die Eigenwerte nicht negativ. Für gewöhnlich ist Q sogar positiv definit und die Eigenwerte größer Null.

Da die zu symmetrischen Bilinearformen gehörenden Matrizen diagonalisierbar sind, ist auch Q diagonalisierbar, also zu einer Diagonalmatrix ähnlich, sogar wegen der Existenz einer Orthonormalbasis aus Eigenvektoren orthogonal ähnlich.

Die Diagonalisierung erfolgt über die Eigenvektoren von Q , die eine Orthonormalbasis des \mathbb{R}^4 erzeugen können [81]. Seien $x_i, i \in \mathbb{N}_4$ die vier normierten Eigenvektoren zu den reellen Eigenwerten $\lambda_i, i \in \mathbb{N}_4$, dann kann durch eine orthogonale Ähnlichkeitstransformation Q in eine Diagonalmatrix D überführt werden. Die reguläre Transformationsmatrix $U = (x_1, x_2, x_3, x_4)^t$ besitzt die Eigenvektoren als Zeilen und ist damit eine Orthogonalmatrix $U \in O(4)$. Da die Eigenvektoren orthonormal sind, gilt

$$U U^t = U^t U = E \quad \Leftrightarrow \quad U^{-1} = U^t.$$

E ist die Einheitsmatrix.

Die Diagonalisierung lautet

$$Q = UDU^t \Leftrightarrow D = U^tQU$$

und wird Hauptachsentransformation genannt. Q und D heißen kongruent oder ähnlich zueinander.

Wegen des Trägheitssatzes von Sylvester [75], besitzen die Eigenwerte der Matrizen nach einer Ähnlichkeitstransformation die gleichen Vorzeichen (echt negativ, Null, echt positiv). Da es sich bei der Ähnlichkeitstransformation um die Hauptachsentransformation handelt, sind die Eigenwerte beider Matrizen sogar identisch. D ist eine Diagonalmatrix, so dass die Diagonalelemente von D die Eigenwerte von D und damit von Q sein müssen:

$$D = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_4 \end{pmatrix}.$$

Nicht näher spezifizierte Einträge der Matrix sind wie im Folgenden identisch Null. Das Quadrik-Fehlermaß im neuen Koordinatensystem lautet

$$\underline{Q}(\underline{v}) = \underline{v}^t D \underline{v}. \quad (3.20)$$

$\underline{v} \in \mathbb{R}^4$ ist dabei

$$\underline{v} = U^t v \Leftrightarrow v = U \underline{v}$$

und ergibt sich aus

$$Q(v) = v^t Q v = v^t U D U^t v = \underline{v}^t D \underline{v} = \underline{Q}(\underline{v}).$$

Für $\underline{Q}(\underline{v})$ umfasst das Minimum immer den Nullpunkt. Dies zeigt, dass das Zentrum der Quadrik auf den Koordinatenursprung rückt. Die Form bleibt wegen der identischen Eigenwerte von Q und D gleich. Die Eigenvektoren hingegen sind wegen des Ausrichtens des Koordinatensystems entlang der Achsen der Quadrik die Einheitsvektoren e_i zu den entsprechenden Eigenwerten λ_i .

Transformation im \mathbb{R}^3

Diese Hauptachsentransformation im \mathbb{R}^4 stellt zwar eine einfache Form dar und vereinfacht auch die Lage der Quadrik im \mathbb{R}^4 , aber die dieser Arbeit zugrundeliegenden verschobenen Quadriken im \mathbb{R}^3 , die nur der kompakten Schreibweise wegen in die homogenen Koordinaten des \mathbb{R}^4 überführt wurden, lassen sich anschaulich besser in zwei Schritten vereinfachen.

Zur Wiederholung sei hier nochmal der Aufbau der homogenen Quadrik beschrieben (siehe Abschnitt 3.1.3):

$$Q = \left(\begin{array}{c|c} A & B \\ \hline B^t & C \end{array} \right), \quad A = \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{pmatrix} \in \mathbb{R}^{3 \times 3}, \quad B = \begin{pmatrix} q_{14} \\ q_{24} \\ q_{34} \end{pmatrix} \in \mathbb{R}^3, \quad C = q_{44} \in \mathbb{R}.$$

Die Matrix A gibt die Form der Quadrik vor, der Vektor B die Verschiebung und der Wert C stellt eine Art Skalierung dar:

$$\tilde{v}^t \begin{pmatrix} A & B \\ B^t & C \end{pmatrix} \tilde{v} = \epsilon \Leftrightarrow \tilde{v}^t \begin{pmatrix} A & B \\ B^t & 0 \end{pmatrix} \tilde{v} = \epsilon - C.$$

Um so kleiner C ist, desto größer wird die mittels ϵ festgelegte Isofläche.

1. Schritt:

Die Aussagen für die Hauptachsentransformation von Q treffen entsprechend auf A zu. Die Hauptachsentransformation der Untermatrix A soll auch der erste Schritt der Vereinfachung sein. Auch hier bezeichnen x_i , $i \in \mathbb{N}_3$ die drei normierten Eigenvektoren zu den reellen Eigenwerten λ_i , $i \in \mathbb{N}_3$. Die Eigenwerte und Eigenvektoren von A stimmen nicht mit denen von Q überein. Es kann lediglich gesagt werden, dass Q mindestens so viele Eigenwerte ungleich Null besitzt wie A . Weiter ist $U = (x_1, x_2, x_3)^t \in O(3)$ die orthogonale Transformationsmatrix, mit der A in eine Diagonalmatrix D überführt werden kann:

$$A = UDU^t \quad \Leftrightarrow \quad D = U^tAU \quad \text{mit } D = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_3 \end{pmatrix}.$$

In die homogene Form lässt sich diese Teiltransformation wie folgt integrieren:

$$\begin{aligned} Q' &= U'^tQU' = \begin{pmatrix} U^t & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A & B \\ B^t & C \end{pmatrix} \begin{pmatrix} U & 0 \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} U^tA & U^tB \\ B^t & C \end{pmatrix} \begin{pmatrix} U & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} U^tAU & U^tB \\ B^tU & C \end{pmatrix} \\ &= \begin{pmatrix} D & B' \\ B'^t & C \end{pmatrix} = \begin{pmatrix} \lambda_1 & & 0 & q'_{14} \\ & \lambda_2 & & q'_{24} \\ 0 & & \lambda_3 & q'_{34} \\ q'_{14} & q'_{24} & q'_{34} & q_{44} \end{pmatrix} \end{aligned}$$

mit $B' = U^tB = (q'_{14}, q'_{24}, q'_{34})^t$, also $B'^t = (U^tB)^t = B^tU$. U' und damit U'^t sind dabei wieder Orthogonalmatrizen und behalten somit bei ihrer Anwendung Winkel und Abstände bei.

2. Schritt:

Der zweite Schritt der Vereinfachung von Q ist die Translation bzw. Verschiebung der Quadrik in den Ursprung. Es handelt sich bei dieser Art von Transformation um eine Kongruenz, die den Abstand der Punkte untereinander beibehält (Bezeichnung: $T \in I(4)$) [75]. Sie lässt sich mit der Transformationsmatrix

$$T = \begin{pmatrix} 1 & 0 & -\frac{q'_{14}}{\lambda_1} \\ & 1 & -\frac{q'_{24}}{\lambda_2} \\ 0 & 0 & 1 & -\frac{q'_{34}}{\lambda_3} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

ausdrücken und ergibt

$$\begin{aligned}
 Q'' = T^t Q' T &= \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ -\frac{q'_{14}}{\lambda_1} & -\frac{q'_{24}}{\lambda_2} & -\frac{q'_{34}}{\lambda_3} & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \lambda_3 & \\ q'_{14} & q'_{24} & q'_{34} & q_{44} \end{pmatrix} \begin{pmatrix} 1 & & -\frac{q'_{14}}{\lambda_1} \\ & 1 & -\frac{q'_{24}}{\lambda_2} \\ & & 1 & -\frac{q'_{34}}{\lambda_3} \\ & & & 1 \end{pmatrix} \\
 &= \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ -\frac{q'_{14}}{\lambda_1} & -\frac{q'_{24}}{\lambda_2} & -\frac{q'_{34}}{\lambda_3} & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \lambda_3 & \\ q'_{14} & q'_{24} & q'_{34} & q_{44} - \sum_{i=1}^3 \frac{q_{i4}^2}{\lambda_i} \end{pmatrix} \\
 &= \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \lambda_3 & \\ & & & q_{44} - \sum_{i=1}^3 \frac{q_{i4}^2}{\lambda_i} \end{pmatrix} = \begin{pmatrix} D & 0 \\ 0 & C' \end{pmatrix}.
 \end{aligned}$$

T verschiebt dabei den Punkt $\bar{v} = (-q'_{14}/\lambda_1, -q'_{24}/\lambda_2, -q'_{34}/\lambda_3)^t$ in den Ursprung. Für $\lambda_i = 0$ ist der entsprechende Eintrag bzw. das Summationsglied Null. Dies gilt für alle nun folgenden Überlegungen. \bar{v} ist der Mittelpunkt der Quadrik, und diese Translation entspricht der Verschiebung des Koordinatensystems aus Abschnitt 2.1.2 bei der Überführung des Ebenenabstandes in eine Norm.

Ergebnis:

$$Q'' = T^t U^t Q U' T = (U' T)^t Q (U' T) \quad (3.21)$$

ist eine Diagonalmatrix und besitzt denselben Rang wie Q , was auch heißt, dass die Anzahl der echt positiven Eigenwerte und der Eigenwerte Null gleich geblieben ist. Da diese Erhaltung auch für die Untermatrix A gilt, ist $q_{44} - \sum_{i=1}^3 \frac{q_{i4}^2}{\lambda_i} = 0$, wenn $\text{rg}(Q) = \text{rg}(A)$, und damit genau dann, wenn das Quadrik-Fehlermaß den minimalen Wert Null annimmt. Dies ist mindestens für den Nullpunkt $\tilde{v} = (0, 0, 0, 1)^t$ der Fall. Ist das Vorzeichen des Eigenwerts (echt negativ, Null, echt positiv), der von A nach Q hinzukommt, positiv, so nimmt das Quadrik-Fehlermaß den Wert Null nicht an. Bei einem hinzukommenden Null-Eigenwert wird am Minimum hingegen Null angenommen. Die transformierten Quadriken besitzen den Nullpunkt als Mittelpunkt. Die Achsenabschnitte bzw. die Halbachsen sind wegen der identischen Eigenwerte der Untermatrix A und der Untermatrix D und der abstandserhaltenden Kongruenz mittels der Matrix T gleich geblieben.

Die Achsen der Quadrik sind nach der Transformation entlang des Koordinatensystems ausgerichtet und haben somit die Einheitsvektoren als Hauptachsen. Die normierten Eigenvektoren von A sind durch die Hauptachsentransformation von A auf die Einheitsvektoren abgebildet worden. Daher sind die Eigenvektoren die Hauptachsen der ursprünglichen Quadrik.

Das Quadrik-Fehlermaß lautet im neuen Koordinatensystem

$$\underline{Q}(\tilde{v}) = \tilde{v}^t Q'' \tilde{v}, \quad \tilde{v} = (x, y, z, 1)^t \in \mathbb{R}^4 \quad (3.22)$$

und wird Hauptachsenform von $Q(\tilde{v})$ genannt. Die Form der Hauptachsenform ist identisch mit der ursprünglichen Form, lediglich die Lage der Quadrik wurde verändert.

Die Transformation aus Gleichung (3.21) verschiebt anschaulich das orthogonale Koordinatensystem in das Zentrum der Quadrik und richtet die Achsen an den Hauptrichtungen der Quadrik aus. Es findet also eine Verschiebung durch T und Drehung durch U' statt.

Beispiel:

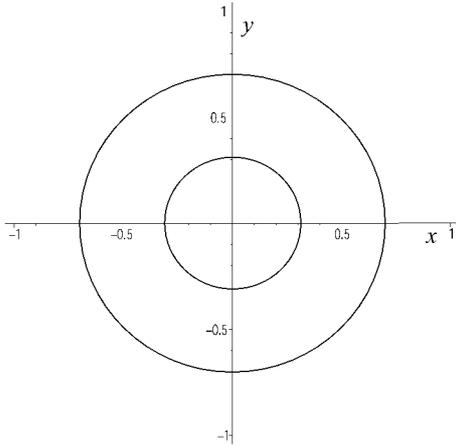


Abb. 3.16: Transformation der beiden Zylinder des Beispiels aus Abbildung 3.15 in die Hauptachsenform

Zum Beispiel ergibt die Transformation aus Gleichung (3.21) für das Beispiel aus Abbildung 3.15 die nebenstehende Zeichnung. Die beiden elliptischen Zylinder werden von oben dargestellt, um zu zeigen, dass sie in den Koordinatenursprung gerückt wurden und die Achsen des Koordinatensystems auf den Achsen der Ellipsen liegen (was sie aber in diesem Fall auch schon vorher getan haben).

Die Eigenwerte der Untermatrix A sind augenscheinlich $\{2, 2, 0\}$ und die dazugehörigen normierten orthogonalen Eigenvektoren sind die Einheitsvektoren, so dass $U' = U'^t = E$ und damit $Q' = Q$. Es findet also keine Drehung des Koordinatensystems statt, da die Achsen der Quadrik schon entsprechend ausgerichtet sind. Würden die Eigenvektoren in U' eine andere Reihenfolge haben, würden lediglich Achsen vertauscht werden, würden sie in die entgegengesetzte Richtung zeigen, würde eine Spiegelung stattfinden, die wegen der Spiegelsymmetrie der Quadriken keine Auswirkung hätte.

Da die Lösungsgerade nicht durch den Ursprung geht, wird aber eine Verschiebung stattfinden müssen. Die Translationsmatrix lautet

$$T = \begin{pmatrix} 1 & 0 & -\frac{q'_{14}}{\lambda_1} \\ & 1 & -\frac{q'_{24}}{\lambda_2} \\ 0 & 1 & -\frac{q'_{34}}{\lambda_3} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\frac{-1}{2} \\ & 1 & -\frac{-1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

da $\lambda_3 = 0$ und $B' = U^t B = B$.

$$\begin{aligned} Q'' = T^t Q' T &= \begin{pmatrix} 1 & 0 & 0 \\ & 1 & 0 \\ 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & -1 \\ & 2 & -1 \\ 0 & 0 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & \frac{1}{2} \\ & 1 & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 2 & 0 & -1 \\ & 2 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \frac{1}{2} \\ & 1 & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

ist die Hauptachsenform von Q und wurde entsprechend dem Aufpunkt der Lösungsgeraden um $(1/2, 1/2, 0)^t$ verschoben. \square

Quadriken in Normalform

Wegen der nun ausgerichteten Form des Quadrik-Fehlermaßes, lassen sich die Isoflächen für $\epsilon \in \mathbb{R}$ auf die normalisierte Form einer Quadrik bringen.

$\lambda_i > 0, i \in \mathbb{N}_3$:

Für ein Ellipsoid ergibt sich die gebräuchliche Schreibweise

$$\lambda_1 \underline{x}^2 + \lambda_2 \underline{y}^2 + \lambda_3 \underline{z}^2 + C' = \epsilon \quad (3.23)$$

$$\frac{\underline{x}^2}{\sqrt{\frac{\epsilon - C'}{\lambda_1}}^2} + \frac{\underline{y}^2}{\sqrt{\frac{\epsilon - C'}{\lambda_2}}^2} + \frac{\underline{z}^2}{\sqrt{\frac{\epsilon - C'}{\lambda_3}}^2} = 1. \quad (3.24)$$

Die Umformung (3.24) ist nur für $\epsilon > C'$ erlaubt. Mit $\epsilon < C'$ wäre die leere Menge Lösung und für $\epsilon = C'$ der einzelne Punkt 0. Des Weiteren ist (3.24) nur zulässig, wenn λ_1, λ_2 und λ_3 echt positiv sind.

$\lambda_1 = 0$:

Ist genau ein Eigenwert Null, zum Beispiel λ_1 , so fällt $\lambda_1 \underline{x}^2$ komplett weg und die gebräuchliche Schreibweise eines elliptischen Zylinders

$$\frac{\underline{y}^2}{\sqrt{\frac{\epsilon - C'}{\lambda_2}}^2} + \frac{\underline{z}^2}{\sqrt{\frac{\epsilon - C'}{\lambda_3}}^2} = 1$$

entsteht. Wegen des Null-Eigenwertes ist die Dimension des Kernes echt positiv, so dass die Abbildung entsprechend der erzeugten Quadrik entartet oder ausgeartet genannt wird.

$\lambda_1 = \lambda_2 = 0$:

Sind genau zwei Eigenwerte Null, zum Beispiel λ_1 und λ_2 , so beschreibt

$$\frac{\underline{z}^2}{\sqrt{\frac{\epsilon - C'}{\lambda_3}}^2} = 1 \quad \Leftrightarrow \quad \underline{z}^2 = \frac{\epsilon - C'}{\lambda_3}$$

ein Ebenenpaar. Die beiden Ebenen liegen $\sqrt{\frac{\epsilon - C'}{\lambda_3}}$ vom Ursprung entfernt und parallel zur xy -Ebene. Die Normalenrichtung ist somit die z -Richtung, also die Richtung des Eigenvektors e_3 des einzigen echt positiven Eigenwertes λ_3 von A . Die Minimalebene ist die xy -Ebene, da nur dort $Q(\tilde{v}) = C'$ gilt.

Ähnliche Überlegungen können auch für den elliptischen Zylinder und das Ellipsoid angestellt werden. Beim elliptischen Zylinder ist der Achsenabschnitt im obigen Beispiel $\sqrt{\frac{\epsilon - C'}{\lambda_2}}$ auf der y -Achse und $\sqrt{\frac{\epsilon - C'}{\lambda_3}}$ auf der z -Achse. Die Normalenrichtungen des Zylinders liegen entsprechend im Raum, der durch die Eigenvektoren e_2 und e_3 aufgespannt wird. Die Mittelgerade liegt somit in Richtung e_1 , der Richtung, in die das Ellipsoid entartet ist und der Eigenwert Null ist. Diese Gerade ist auch Lösung des Minimalproblems, da nur genau hier der Minimalwert C' angenommen wird.

Bei einem Ellipsoid sind die Achsenabschnitte entsprechend. Ist Gleichung (3.24) zulässig, sind also λ_1, λ_2 und λ_3 größer Null und $\epsilon > C'$, dann beschreibt die Gleichung ein Ellipsoid mit Achsenabschnitten $\sqrt{\frac{\epsilon - C'}{\lambda_1}}$, $\sqrt{\frac{\epsilon - C'}{\lambda_2}}$ und $\sqrt{\frac{\epsilon - C'}{\lambda_3}}$ an den x -, y - und z -Achsen. Auf dem Ellipsoid wird jede

Normalenrichtung angenommen und es gibt keine Richtung, in der das Ellipsoid entartet ist. Das Minimum ist nur der Ursprung 0, da nur dort $Q(\tilde{v})$ minimal C' ergibt.

An den Standardformen der Quadriken ist abzulesen, dass die Ausdehnung der Quadrik in Richtung eines Eigenvektors umgekehrt proportional zur Wurzel des dazugehörigen Eigenwertes ist. Ein großer Eigenwert bedeutet also eine geringe und ein kleiner Eigenwert eine große Ausdehnung. Für einen Eigenwert Null geht die Ausdehnung gegen Unendlich und erzeugt in der diesem Eigenwert zugehörigen Eigenvektorrichtung eine Degeneration. Ein Eigenwert Null öffnet das Ellipsoid an einer Seite zu einem elliptischen Zylinder, ein weiterer Null-Eigenwert öffnet diesen Zylinder an einer weiteren Seite zu einem Ebenenpaar.

Beispiel:

Im Beispiel zur Abbildung 3.16 lautet das Minimum

$$C' = q_{44} - \sum_{i=1}^3 \frac{q_{i4}^2}{\lambda_i} = 2 - \left(\frac{(-1)^2}{2} + \frac{(-1)^2}{2} \right) = 1$$

und wird mit Abstand

$$\tilde{v} = \begin{pmatrix} -q'_{14}/\lambda_1 \\ -q'_{24}/\lambda_2 \\ -q'_{34}/\lambda_3 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix}$$

vom Ursprung entlang des Eigenvektors $(0, 0, 1)^t$ zum Eigenwert 0 angenommen, was der zuvor in Gleichung (3.19) angegebenen Geraden entspricht. Die elliptischen Zylinder sind für $\epsilon_1 = 1, 2$ und $\epsilon_2 = 2$ gezeichnet worden. Dadurch ergeben sich die Achsenabschnitte für ϵ_1 auf der x - und y -Achse bei

$$\sqrt{\frac{\epsilon_1 - C'}{\lambda_1}} = \sqrt{\frac{\epsilon_1 - C'}{\lambda_2}} = \sqrt{\frac{1,2 - 1}{2}} \approx 0,316,$$

da $\lambda_1 = \lambda_2 = 2$. Für ϵ_2 lauten beide Achsenabschnitte $\sqrt{1/2} \approx 0,707$. □

3.2.4 Allgemeiner Ebenenabstand als Norm

Jede symmetrische positiv definite Matrix $M \in \mathbb{R}^{n \times n}$, ($M \in SPD$) induziert eine Norm in der folgenden Form [8]:

$$\|x\|_M = \sqrt{x^t M x} \quad \forall x \in \mathbb{R}^n \quad (3.25)$$

und wird als Riemann-Norm bezeichnet. Für $M = E$ ergibt sich als Sonderfall die Euklidische Norm.

Beweis:

Sei $x, y \in \mathbb{R}^n$ und $\alpha \in \mathbb{R}$.

- Definitheit:

$$\|x\|_M = \sqrt{x^t M x} > 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\} \quad (\text{da } M \text{ positiv definit})$$

- Homogenität:

$$\|\alpha x\|_M = \sqrt{(\alpha x)^t M (\alpha x)} = \sqrt{\alpha^2 x^t M x} = |\alpha| \sqrt{x^t M x} = |\alpha| \|x\|_M$$

- Dreiecksungleichung:

$$\begin{aligned}
 \|x + y\|_M^2 &= (x + y)^t M (x + y) = x^t M x + y^t M x + x^t M y + y^t M y \\
 &= x^t M x + 2x^t M y + y^t M y \quad (\text{da } M \text{ symmetrisch}) \\
 &\stackrel{(*)}{\leq} x^t M x + 2\sqrt{x^t M x} \sqrt{y^t M y} + y^t M y \\
 &= (\sqrt{x^t M x} + \sqrt{y^t M y})^2 = (\|x\|_M + \|y\|_M)^2 \\
 \Leftrightarrow \|x + y\|_M &\leq \|x\|_M + \|y\|_M \quad (\text{wegen der Monotonie der Wurzel})
 \end{aligned}$$

Für (*) ist noch zu zeigen, dass

$$x^t M y \leq \sqrt{x^t M x} \sqrt{y^t M y}.$$

Wegen der Definitheit von $\|\cdot\|_M$ gilt für beliebige $\alpha, \beta \in \mathbb{R}$ und $y \in \mathbb{R}^n \setminus \{0\}$:

$$\begin{aligned}
 0 &\leq \|\alpha x + \beta y\|_M^2 = \alpha^2 x^t M x + 2\alpha\beta x^t M y + \beta^2 y^t M y \quad (\text{setze } \alpha = y^t M y) \\
 \Rightarrow 0 &\leq (y^t M y)^2 x^t M x + 2\beta y^t M y x^t M y + \beta^2 y^t M y \quad (\text{teile durch } y^t M y > 0) \\
 \Leftrightarrow 0 &\leq x^t M x y^t M y + 2\beta x^t M y + \beta^2 \quad (\text{setze } \beta = -x^t M y) \\
 \Rightarrow 0 &\leq x^t M x y^t M y - 2(x^t M y)^2 + (x^t M y)^2 = x^t M x y^t M y - (x^t M y)^2 \\
 \Leftrightarrow (x^t M y)^2 &\leq x^t M x y^t M y \\
 \Leftrightarrow x^t M y &\leq \sqrt{x^t M x y^t M y} = \sqrt{x^t M x} \sqrt{y^t M y} \quad (\text{da } \|\cdot\|_M \text{ definit}).
 \end{aligned}$$

Für $y = 0$ gilt trivialerweise:

$$\begin{aligned}
 x^t M 0 &\leq \sqrt{x^t M x} \sqrt{0^t M 0} \\
 \Leftrightarrow 0 &\leq 0. \quad \blacksquare
 \end{aligned}$$

Zwar ist gemäß (3.25) für reguläre Q eine Norm im \mathbb{R}^4 durch

$$\|x\|_Q = \sqrt{x^t Q x} \quad \forall x \in \mathbb{R}^4$$

gegeben, jedoch ist eine Quadrik gewünscht, die auch im \mathbb{R}^3 eine Norm beschreibt. In der Einführung aus Abschnitt 2.1.2 ist gezeigt worden, dass die Wurzel der Summe der Quadratabstände von sich in einem Punkt schneidenden Ebenen durch eine Translation eine Norm darstellt. Um die Homogenität des Fehlermaßes herstellen zu können, darf die Quadrik außer in A nur Nullen enthalten.

Durch eine Translation kann wie im zweiten Schritt der Hauptachsentransformation eine allgemeine Quadrik in die Gestalt

$$\begin{pmatrix} A & 0 \\ 0 & * \end{pmatrix}$$

gebracht werden. Diese soll nun bestimmt werden, ohne zusätzlich die Quadrik zu rotieren und unter Angabe der Transformationsvorschrift.

Im Folgenden wird davon ausgegangen, dass Q und damit A regulär sind, da dies den Normalfall für das Quadrik-Fehlermaß in der Anwendung darstellt (siehe Abschnitt 4.5). Bei der Translation um ein festes $\bar{v} \in \mathbb{R}^3$ wird v überführt in \underline{v} gemäß

$$\underline{v} = v + \bar{v},$$

so dass für $Q(v) = v^t A v + 2B^t v + C$ der Verschiebungsanteil $2B^t v$ entfällt. \bar{v} muss dazu folgendermaßen gewählt werden:

$$\begin{aligned} Q(\underline{v}) &= \underline{v}^t A \underline{v} + 2B^t \underline{v} + C \\ &= (v + \bar{v})^t A (v + \bar{v}) + 2B^t (v + \bar{v}) + C \\ &= v^t A v + 2\bar{v}^t A v + \bar{v}^t A \bar{v} + 2B^t v + 2B^t \bar{v} + C \quad (\text{da } A \text{ symmetrisch}) \\ &= v^t A v + 2(B^t + \bar{v}^t A)v + \bar{v}^t A \bar{v} + 2B^t \bar{v} + C \end{aligned}$$

$$\begin{aligned} \Rightarrow 2(B^t + \bar{v}^t A) &= 0 \quad \Leftrightarrow \quad \bar{v}^t A = -B^t \quad \Leftrightarrow \quad \bar{v}^t = -B^t A^{-1} \quad (3.26) \\ \Leftrightarrow \bar{v} &= (-B^t A^{-1})^t = -(A^{-1})^t B = -(A^t)^{-1} B = -A^{-1} B \quad (\text{da } A \text{ symmetrisch}) \end{aligned}$$

\bar{v} entspricht wie nicht anders zu erwarten dem optimalen Verschmelzungspunkt, also dem Mittelpunkt der Quadrik. Mit der Translationsmatrix

$$T = \begin{pmatrix} E & \bar{v} \\ 0 & 1 \end{pmatrix} \in I(4)$$

ergibt sich somit

$$\begin{aligned} Q' &= T^t Q T = \begin{pmatrix} E & 0 \\ \bar{v} & 1 \end{pmatrix} \begin{pmatrix} A & B \\ B^t & C \end{pmatrix} \begin{pmatrix} E & \bar{v} \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} A & B \\ \bar{v}^t A + B^t & \bar{v}^t B + C \end{pmatrix} \begin{pmatrix} E & \bar{v} \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} A & A\bar{v} + B \\ \bar{v}^t A + B^t & (\bar{v}^t A + B^t)\bar{v} + \bar{v}^t B + C \end{pmatrix} \\ &\stackrel{(*)}{=} \begin{pmatrix} A & 0 \\ 0 & \bar{v}^t A \bar{v} + 2B^t \bar{v} + C \end{pmatrix} \end{aligned}$$

Dass (*) gilt, ist den Gleichungen aus (3.26) zu entnehmen.

Im verschobenen Koordinatensystem entspricht also

$$\underline{Q}(\underline{v}) = \underline{\tilde{v}}^t \underline{Q}' \underline{\tilde{v}} = \underline{\tilde{v}}^t (T^t Q T) \underline{\tilde{v}} = (T \underline{\tilde{v}})^t Q (T \underline{\tilde{v}}) = \tilde{v}^t Q \tilde{v} = Q(v),$$

so dass

$$\tilde{v} = T \underline{\tilde{v}} \quad \Leftrightarrow \quad \underline{\tilde{v}} = T^{-1} \tilde{v}.$$

Das Minimum bleibt nach der Verschiebung dasselbe, nur dass es im ursprünglichen Koordinatensystem bei \bar{v} und im verschobenen im Ursprung angenommen wird, und lautet

$$Q(\bar{v}) = \underline{Q}(0) = \bar{v}^t A \bar{v} + 2B^t \bar{v} + C = C'.$$

Aufgrund der Konstruktion des Fehlermaßes über die Summe von Quadratabständen ist das Quadrik-Fehlermaß immer positiv und der Wertebereich von C kann daher eingeschränkt werden:

$$\bar{v}^t A \bar{v} + 2B^t \bar{v} + C \geq 0 \quad \Leftrightarrow \quad C \geq -\bar{v}^t A \bar{v} - 2B^t \bar{v}.$$

Wegen der Homogenität einer Norm muss das Minimum einer Norm Null sein. Es gibt aber keine lineare reguläre Abbildung, die $Q(\underline{v}) = \underline{v}^t A \underline{v} + C'$ in $\underline{v}^t A' \underline{v}$ überführt, was einer Parallelverschiebung aller zum Maß beitragenden Ebenen in den Ursprung bedeuten würde. Es gibt aber eine einfache Rechenvorschrift, um eine zum Quadrik-Fehlermaß verwandte Norm herzustellen:

$$Q^N(\underline{v}) = \sqrt{Q(\underline{v}) - C'} = \sqrt{\underline{v}^t A \underline{v}} \quad \Leftrightarrow \quad Q(\underline{v}) = (Q^N(\underline{v}))^2 + C'$$

$Q^N(\underline{v})$ ist eine Norm, da A symmetrisch und positiv definit ist. Die Wurzel ist erlaubt, da per Konstruktion $Q(\underline{v}) - C' \geq 0$. Die Monotonie der Wurzel behält die Ordnung der Norm gegenüber dem Fehlermaß bei, linearisiert sie nur, so dass Isoflächen äquidistanter Fehler gleich weit voneinander entfernt liegen.

3.2.5 Klassifikation

In diesem Abschnitt werden die verschiedenen Lösungsmannigfaltigkeiten des Minimums des Quadrik-Fehlermaßes und deren Isoflächen vorgestellt.

(0 0 0 0):

Neben der Nullquadrik ($Q = 0$), die für $Q(\tilde{v}) = 0$ den ganzen Raum als Lösung besitzt und aus der leeren Menge von Ebenenabständen hervorgeht, gibt es drei weitere Arten von Lösungsmannigfaltigkeiten unterschiedlicher Dimension, bei denen $Q(\tilde{v})$ minimal ist. Bei der Nullquadrik sind alle Eigenwerte Null (0 0 0 0). Im Folgenden werden die Eigenwerte bzw. deren Vorzeichen als Quadrupel in der Form (* * * *) geschrieben, wobei die ersten drei Eigenwerte der Untermatrix A zugeordnet sind. Die Reihenfolge der Eigenwerte von A ist dabei unwesentlich. Als Vorzeichen wird für "*" entweder "+", "-" oder "0" angegeben, wobei das Vorzeichen "-" wegen der positiven Eigenwerte des Quadrik-Fehlermaßes nicht vorkommt.

Da Q positiv semidefinit ist, ist die Isofläche der Lösungsmannigfaltigkeit ein Ellipsoid, das manchmal degeneriert ist. Degeneriert heißt dabei, dass die Ausdehnung in eine der drei Achsen des Ellipsoids unendlich ist. Für die Nullquadrik sind das alle drei Achsen. Andere Degenerationen sind bei zwei Richtungen, in denen das Ellipsoid unendlich ausgedehnt ist, zwei parallele Ebenen und bei einer Richtung mit unendlicher Ausdehnung ein Zylinder. Der Zylinder kann eine elliptische Grundform besitzen und wird daher auch elliptischer Zylinder genannt.

(+ 0 0 0):

Ein einfacher Fall stellt sich mit nur einer Ebene ein. Da $Q(\tilde{v})$ genau nur auf der Ebene selbst Null ist, ist diese gleichzeitig die Lösungsmannigfaltigkeit des Minimums des Quadrik-Fehlermaßes. Die Isofläche sind Ebenenpaare, die die Lösungsebene in ihrer Mitte haben. Da die einzige Richtung, in der die Isooberfläche nicht degeneriert ist, die Normalenrichtung der Fläche ist, ist der Eigenvektor zum einzigen echt positiven Eigenwert der Quadrik der Normalenvektor. Diese einfache Quadrik entspricht der fundamentalen Quadrik Q^f aus Gleichung (3.3), die für jedes Ebenendreieck den Quadratabstand angibt.

Die Quadrik an jedem Knoten setzt sich dann aus diesen fundamentalen Quadriken der angrenzenden Ebenen zusammen. Die Bestimmung eines Minimums findet aber erst für ein Kontraktionspaar statt, für das die Quadriken beider Knoten zusammengefasst werden.

Der einfache Fall einer Ebene entspricht dem Fall mehrerer identischer Ebenen. Der Fehler, den $Q(\tilde{v})$ ausdrückt, wird zwar größer, aber die Lage und Art der Lösung bleibt gleich.

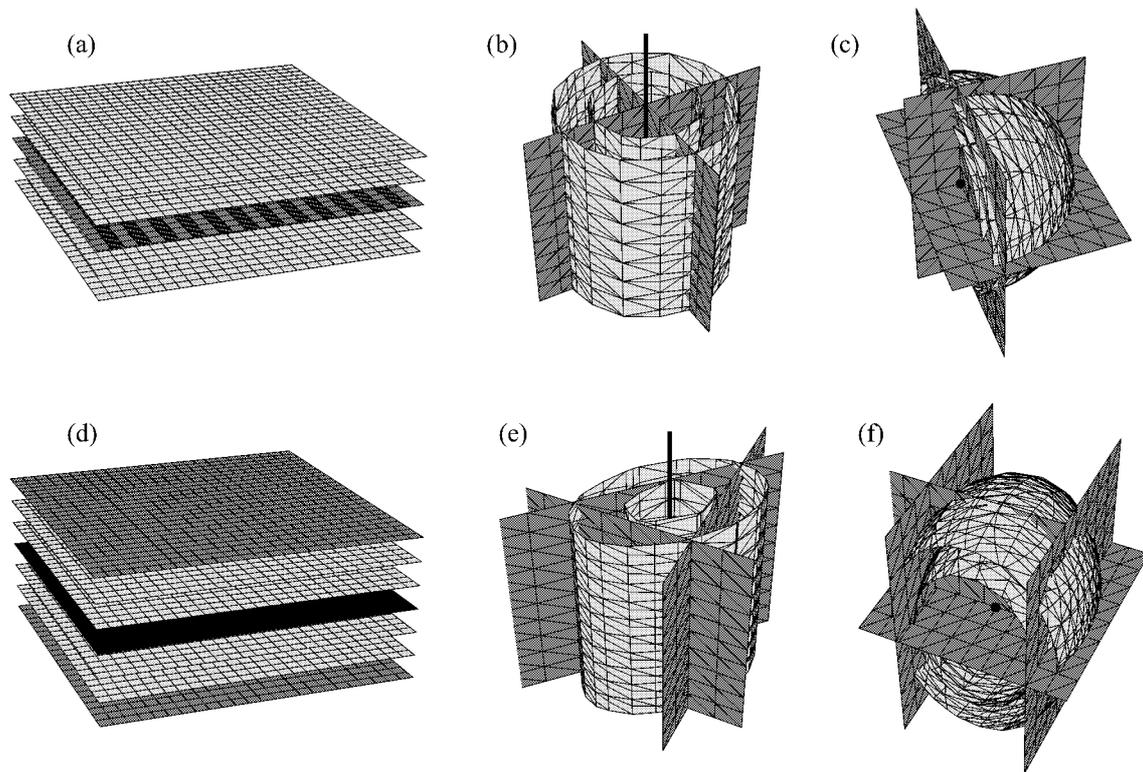


Abb. 3.17: Die sechs Skizzen zeigen Beispiele von Ebenenkonstellationen (dunkel) für verschiedene Klassen von Quadriken (hell) des Quadrik-Fehlermaßes und deren Minima (schwarz)

In Abbildung 3.17 (a) ist dieser einfache Fall skizziert. Die Isoflächen sind wie in allen anderen Fällen hellgrau dargestellt. Die Ebenenpaare sind wie auch in den restlichen fünf Skizzen für zwei verschiedene ϵ -Werte der Gleichung $Q(\vec{v}) = \epsilon$ eingezeichnet. Die Lösungsmannigfaltigkeit ist für gewöhnlich schwarz dargestellt und die vorgegebenen Ebenen dunkelgrau. Da im Fall (a) diese identisch sind, ist die gemeinsame Ebene schraffiert gezeichnet.

Die schwarz angegebene Lösung ist in der ersten Spalte aus Abbildung 3.17 eine Ebene, in der zweiten Spalte eine Gerade und in der dritten Spalte ein einzelner Punkt. In der oberen Zeile der Abbildung nimmt das Fehlermaß den Wert Null an, ist die Lösung also die Schnittmenge der Ebenen. Die untere Zeile hat eine echt positives Fehlermaß und damit haben die Ebenen keine gemeinsame Schnittmenge. Die Lösung liegt im Zentrum der Isoflächen.

(+ 0 0 +):

Wird die Quadrik durch zwei nicht identische Ebenen bestimmt, können sie wie in Abbildung 3.17 (d) parallel oder (b) nicht parallel zueinander liegen. Die schon aus (a) bekannten Ebenenpaare sind auch Isoflächen der parallelen Ebenen und besitzen genau in ihrer Mitte die Lösungsebene. Der gleiche Fall tritt ein, falls weitere Ebenen parallel zueinander sind und mindestens zwei nicht identisch sind.

(+ + 0 0):

Zwei nicht parallele Ebenen schneiden sich in einer in der Zeichnung schwarz dargestellten Schnittgeraden, die gleichzeitig Lösung des Minimalproblems ist. Statt Ebenenpaare besitzt in (b) das Fehlermaß Zylinder als Isofläche. Die Grundform der Zylinder in diesem Beispiel ist ein Kreis, da die zwei Ebenen senkrecht zueinander sind. Würden die Ebenen nicht orthogonal aufeinanderstehen, wären die

Zylinder ellipsenförmig. Ähnliche Lösungen würde man auch mit mehreren Ebenen erhalten, wenn sie alle durch die gleiche Schnittgerade gehen.

(+ + 0 +):

Die Normalen der Ebenen in Abbildung 3.17 (e) liegen wie in dem eben betrachteten Fall (b) in einer Ebene bzw. die Ebenen sind parallel zueinander. Die Ebenen besitzen aber keine gemeinsame Schnittgerade, so dass $Q(\tilde{v})$ echt positiv ist. Die Isoflächen sind auch elliptische Zylinder, und die Lösung ist ebenfalls eine Gerade in deren Zentrum.

(+ + + 0):

Das Beispiel im Fall (c) beschreibt eine Konstellation, die ab drei Ebenen auftreten kann. Alle Ebenen gehen genau durch einen gemeinsamen Punkt, der gleichzeitig auch Zentrum der ellipsoidischen Isofläche ist. Hier sind die Isoflächen Kugeln, da die drei Ebenen genau senkrecht aufeinander stehen und damit die Eigenwerte von A identisch sind.

(+ + + +):

Der in der Anwendung mit Abstand am häufigsten auftretende Fall ist in (f) skizziert. Die mindestens vier Ebenen liegen so, dass sie keine gemeinsame Schnittmenge besitzen und die Normalen wie in (c) den gesamten \mathbb{R}^3 aufspannen. Das Minimum liegt im Zentrum der ellipsoidischen Isoflächen. Dieser Punkt liegt in der Zeichnung ausnahmsweise auf der Schnittgeraden zweier Ebenen, da die beiden anderen Ebenen parallel zueinander sind. Dieser letzte Fall ist der einzige mit einer positiv definiten Fehlerquadrik.

Dimension der Lösung

Es ist zu beobachten, dass die Dimension des Raumes, den die Normalen aufspannen (“RE”) und die Dimension der Lösung (“LD”) in der Summe drei ergibt:

$$\text{RE} + \text{LD} = 3. \quad (3.27)$$

Beweis:

Die Ebenenormalen liegen im Raum, der von den Eigenvektoren von A aufgespannt wird, die zu den echt positiven Eigenwerten gehören. Die Lösung liegt orthogonal zu den Ebenennormalen und wird von den zu dem Eigenwert Null gehörigen Eigenvektoren von A aufgespannt. Da eine Orthonormalbasis über die Eigenvektoren angegeben werden kann, gibt es für A genau drei Eigenvektoren, womit Gleichung (3.27) bewiesen wäre. ■

“RE” entspricht dem Rang von A und damit der Dimension des Bildraumes und “LD” der Dimension des Kernes:

$$\text{RE} = \dim(\text{Im}(A)) = \text{rg}(A), \quad \text{LD} = \dim(\text{Ker}(A)) = \text{defekt}(A).$$

Auch dies beweist Gleichung (3.27), da hier gilt, dass $\text{rg}(A) + \text{defekt}(A) = 3$. Wegen der Symmetrie von A ist ferner

$$\text{Im}(A) \oplus \text{Ker}(A) = \mathbb{R}^3,$$

also das Bild und der Kern ergeben eine orthogonale Zerlegung des Raumes.

Dass nicht alle Formen einer Quadrik bei den Isoflächen auftreten, liegt daran, dass über die Ebenenabstandsquadrate nicht alle quadratischen Formen erzeugt werden können. Es treten keine negativen

Eigenwerte auf. Flächen zweiter Ordnung wie Hyperboloide, Kegel, Paraboloiden sind keine Isoflächen des Quadrik-Fehlermaßes.

Tabelle 3.1 fasst die eben gemachten Aussagen zusammen.

Ebenen	Abb. 3.17	ME	Eigenwerte		RE	Lösungsmannig- faltigkeit	LD	Isofläche
			A	Q				
keine	–	0	(0 0 0)	(0)	0	\mathbb{R}^3	3	{}
identische parallele; 2 verschieden	(a)	1	(+ 0 0)	(0)	1	Ebene selbst	2	par. Eb.paar
	(d)	2	(+ 0 0)	(+)	1	Ebene zwischen Isofl.paar	2	par. Eb.paar
genau eine Schnittgerade par. zu Geraden; keine Schnittgr.	(b)	2	(+ + 0)	(0)	2	Schnittgerade	1	ellip. Zylinder
	(e)	3	(+ + 0)	(+)	2	Gerade im Zylinderzentrum	1	ellip. Zylinder
genau ein Schnittpunkt nicht par.; kein Schnittpunkt	(c)	3	(+ + +)	(0)	3	Schnittpunkt	0	Ellipsoid
	(f)	4	(+ + +)	(+)	3	Punkt im Ellipsoidenzentr.	0	Ellipsoid

Tab. 3.1: Klassifikation der Quadriken; Erläuterungen: “Abb.” gibt an, welche Grafik aus Abbildung 3.17 ein Beispiel dieser Problemklasse darstellt. “ME” in Spalte zwei heißt “minimale Ebenenanzahl” und gibt an, wieviele Ebenen mindestens für diese Art von Lösung nötig sind. Die Anzahl der Ebenen ist nach oben offen. Nur für die erste Zeile ohne Ebenen müssen es genau Null Ebenen sein. Spalte drei und vier geben an, ob die Eigenwerte Null oder echt positiv sind. Die dritte Spalte bezieht sich auf die Eigenwerte der Untermatrix A und die vierte auf den Eigenwert, der für ganz Q hinzukommt. In der fünften Spalte bezeichnet “RE” die “Raumdimension der Ebenennormalen”, was die Dimension des von den Normalen der zu bewertenden Ebenen aufgespannten Raumes angeben soll. Die Dimension von Null für keine Normalen ist Definitionssache, macht die Formel (3.27) aber allgemeingültig. “LD” in der vorletzten Spalte gibt die Dimension der Lösung an.

3.2.6 Optimum

Bisher wurde hauptsächlich das Verhalten des Quadrik-Fehlermaßes untersucht, indem ihre Isoflächen näher beschrieben wurden. An den Isoflächen ist gut zu erkennen, in welche Richtungen der Fehler schnell größer wird und in welche Richtung kaum bis gar kein Fehler hinzukommt. In den entarteten Richtungen kommt zum Beispiel kein Fehler hinzu, in den Richtungen der Eigenvektoren großer Eigenwerte hingegen steigt der Fehler besonders stark, insbesondere zeigt der größte Eigenwert die Richtung stärkster Fehlerveränderungen an. Der Fehler in Richtung eines Eigenvektors steigt also umgekehrt proportional zum dazugehörigen Eigenwert.

Anhand dieser Information der Isoflächen ist auch klar, in welche Richtungen der Verschmelzungspunkt leichter zu verschieben ist und für welche Verschiebung mehr Energie nötig ist. Im Verfahren wird der Ort für den Verschmelzungspunkt gewählt, der den geringsten Fehler besitzt. Die Hauptachsenform richtet das Koordinatensystem so aus, dass die Achsen der Quadrik auf den Koordinatenachsen liegen und das Zentrum im Nullpunkt liegt. Die normierte allgemeine Form der Quadrik

$$Q(\underline{\tilde{v}}) = \lambda_1 \underline{x}^2 + \lambda_2 \underline{y}^2 + \lambda_3 \underline{z}^2 + C' \quad (3.28)$$

zeigt, dass im Koordinatenursprung der einzige Extremwert angenommen wird, und dass es sich dabei um ein Minimum handelt, da für $\lambda_1, \lambda_2, \lambda_3 \neq 0$ gilt:

$$\begin{aligned} \text{grad}(Q(\underline{\tilde{v}})) &= (2\lambda_1 \underline{x}, 2\lambda_2 \underline{y}, 2\lambda_3 \underline{z})^t = 0 \\ \Leftrightarrow \underline{x} &= 0 \wedge \underline{y} = 0 \wedge \underline{z} = 0 \end{aligned}$$

und zudem $\underline{Q}(\underline{v}) > 0 \quad \forall \underline{v} \neq (0, 0, 0, 1)^t$ ist.

Für echt positive Eigenwerte wird also vom Ursprung ausgehend der Fehler immer größer. Das Minimum im Mittelpunkt der Quadrik bleibt auch nach Verschiebung des Koordinatensystems bestehen. Der Mittelpunkt liegt dort laut Abschnitt 3.2.3 bei $\bar{v} = (-q'_{14}/\lambda_1, -q'_{24}/\lambda_2, -q'_{34}/\lambda_3)^t$. Für $\lambda_i = 0$ ist der entsprechende Eintrag Null und die Quadrik ist in Richtung dieses Eigenwertes entartet. In Gleichung (3.6) wurde er auch schon mit Hilfe von Adjunkten mit $1/Q_{44}(Q_{41}, Q_{42}, Q_{43})^t$ angegeben.

Sind Eigenwerte der Matrix A identisch Null, so ist das Minimum nicht mehr eindeutig und der Ort, an dem es angenommen wird, erstreckt sich über den von den zu den Null-Eigenwerten gehörenden Eigenvektoren aufgespannten Raum. Dies ist bei einem Null-Eigenwert eine Gerade und bei zwei Null-Eigenwerten eine Ebene. Sind alle Eigenwerte Null, wird das Minimum im ganzen Raum angenommen. Dieser Fall liegt aber nur für die initiale Nullquadrik vor, die durch eine leere Menge an Ebenenabständen erzeugt wird. Das Minimum ist dann Null.

Die Eigenvektoren liegen bei der Hauptachsenform auf den Koordinatenachsen und entsprechend auf den Achsen der Quadrik. Im ursprünglichen Koordinatensystem liegen die Eigenvektoren ebenfalls auf den Achsen der Quadrik und geben bei Entartung durch Null-Eigenwerte den Raum vor, auf den sich das Minimum erstreckt.

Der im Minimum angenommene Wert ist in der Hauptachsenform

$$\underline{Q}((0, 0, 0, 1)^t) = C' = q_{44} - \sum_{\substack{i=1 \\ \lambda_i \neq 0}}^3 \frac{q_{i4}^2}{\lambda_i} = Q(\bar{v})$$

und ist es auch in der ursprünglichen Form, da das Koordinatensystem lediglich verschoben und gedreht wurde.

Der Minimalwert ist Null, wenn die Ebenen eine gemeinsame nicht leere Schnittmenge haben, da nur dann ein Punkt x_0 existiert, der auf allen Ebenen liegt, so dass

$$\sum_{i=1}^k \|n_i^t x_0\|_2^2 = 0 \quad \Leftrightarrow \quad n_i^t x_0 \quad \forall i \in \mathbb{N}_k.$$

Dies tritt genau dann ein, wenn das Vorzeichen des Eigenwertes, der von A nach Q hinzukommt, Null ist, also

$$\text{rg}(A) = \text{rg}(Q) \quad \Leftrightarrow \quad Q(\bar{v}) = 0.$$

Die implizite Gleichung der Quadriken $Q(v) = \epsilon$ zur Erzeugung der Isoflächen liefert für den Minimalwert $Q(v) = C'$ die Lösungsmenge. Die Isofläche existiert somit nur für $\epsilon > C'$, für $\epsilon < C'$ besitzt $Q(v) = \epsilon$ keine Lösung.

Beispiele zur Berechnung des Minimums sind im Anhang C.1 und in Abschnitt 3.2.3 zu finden.

3.2.7 Oberflächenkrümmung

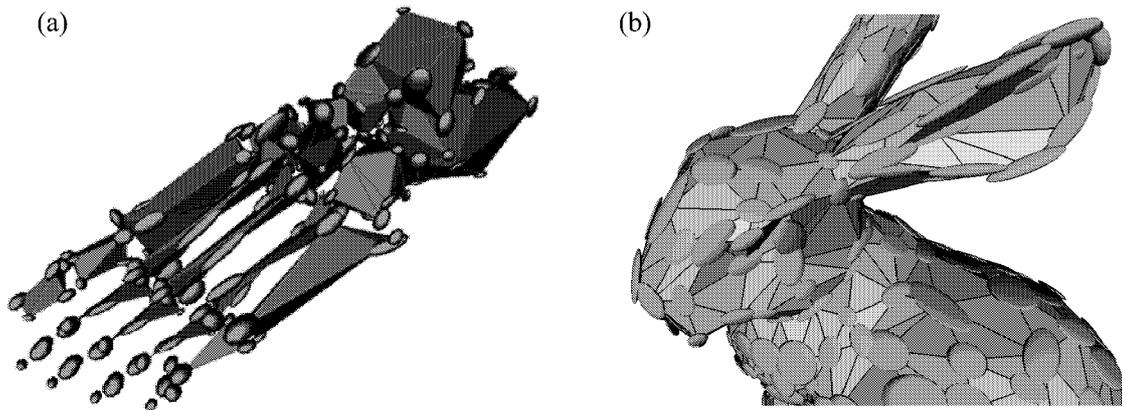


Abb. 3.18: Ellipsoide als Isofläche des Quadrik-Fehlermaßes an den jeweiligen Knoten eines vereinfachten (a) Fußskelettes und des (b) Stanford Bunnys; Bilder aus [31] und [82]

Abbildung 3.18 zeigt Isoflächen für das Quadrik-Fehlmaß gemäß $Q(\vec{v}) = \epsilon$, deren Zentrum jeweils im zugehörigen Knoten liegt. Da beide Modelle zu dem abgebildeten Zeitpunkt schon stark vereinfacht sind, sind die Quadriken aller Voraussicht nach positiv definit und die Isoflächen damit Ellipsoide. Modell (a) ist ein auf 250 Dreiecke vereinfachtes Fußskelett mit original 4204 Dreiecken. Das “Stanford Bunny” von (b) hat original 69451 Dreiecke und ist in der Abbildung auf 1000 Dreiecke vereinfacht worden. Es ist gut zu erkennen, dass die Ellipsoide sich der Oberfläche der Modelle anpassen und sich entlang der ebenen Bereiche am stärksten ausdehnen. An Kanten sind zigarrenförmige Ellipsoide zu sehen, da nur entlang der Kante eine kostengünstige Verschiebung des zur Quadrik gehörigen Knotens möglich ist. Die Ellipsoide sind für identische ϵ gezeichnet worden, so dass das Volumen der Ellipsoide etwas über die Wahrscheinlichkeit einer Verfeinerung an den Knoten aussagt. Voluminöse Ellipsoide zeigen an, dass dort eine Verschiebung leichter möglich ist als im Bereich kleiner Ellipsoide.

In Richtung der mittleren Normalen haben die Ellipsoide die geringste Breite und zeigen an, dass sich das Fehlermaß in diese Richtung am stärksten verändert und damit schnell groß wird. Die Ellipsoide strecken sich in Richtung geringer Krümmung und sind in Richtung starker Krümmung flach. Der Verschmelzungspunkt wandert am ehesten in Richtung der stärkeren Ellipsoidausdehnung. Innerhalb des Ellipsoids ist der Fehler über das Quadrik-Fehlermaß auf ϵ begrenzt.

Wie in Abschnitt 3.2.3 beschrieben, hängt die Ausdehnung des Ellipsoids umgekehrt proportional mit der Größe des Eigenwertes in der entsprechenden Eigenvektor-Richtung zusammen. Die Krümmung ist somit groß in Richtung der Eigenvektoren, deren zugehörige Eigenwerte ebenfalls groß sind, umgekehrt zeigen kleine Eigenwerte eine geringe Krümmung an, bis hin zu dem Fall, dass die Krümmung und entsprechend der Eigenwert Null sind.

4

Ergebnisse

In diesem Kapitel sollen Ergebnisse der dieser Arbeit zugrundeliegenden Implementation vorgestellt werden. Das der Vereinfachung dienende Fehlermaß ist bereits in Kapitel 2 ausführlich von anderen Verfahren abgegrenzt und mit ihnen verglichen worden. Das Ziel der Arbeit ist eine schnelle, qualitativ hochwertige Vereinfachung, deren Detailgrad aufgrund äußerer Vorgaben adaptiv anpassbar ist.

Die visuell guten Ergebnisse der Netzvereinfachung auf der Basis des Quadrik-Fehlermaßes sind schon an vielen Abbildungen, wie zum Beispiel der Geländeoberfläche in Abbildung 2.20 und der Kuh in Abbildung 3.5, zu sehen gewesen. Auch sind einige Spezialfälle bereits beleuchtet worden. Dieses Kapitel soll generelle Aussagen bezüglich des Verfahrens machen und die Einschränkungen bzw. Schwächen aufzeigen.

Nach einer Einführung beschäftigt sich der anschließende Abschnitt mit der theoretischen und praktischen Geschwindigkeit der Implementation. Darauf folgt eine Bewertung des benötigten Speicherplatzes zur Berechnung der Vereinfachung und der Qualität derselben. Eine Analyse der Anzahl singulärer Quadriken und Bestrafungen und der adaptiven Anpassung der Netzdichte schließen sich an. Die bis dahin meist sehr positiven Ergebnisse des Verfahrens sollen zum Schluss einigen Problemen desselben gegenübergestellt werden.

4.1 Einführung

Das Netz, das es zu vereinfachen gilt, wird in der zugrundeliegenden Anwendung mit dem Marching-Cube-Algorithmus erzeugt, der eine Isofläche durch einen Volumendatensatz^G einer Elektronendichteverteilung legt.

Eine primitive Vereinfachung ist auch schon mittels des Marching-Cube-Algorithmus möglich, indem die Schrittweite der Abtastung vergrößert wird. Diese primitive Methode ist in Abbildung 4.1 (b) zu sehen und zeigt, dass die Qualität bei gleicher Dreiecksanzahl nicht mit einer selektiven Simplifizierung mittels des dieser Arbeit zugrundeliegenden Verfahrens (c) mithalten kann. Die Struktur der Originalfläche unter (a) wird in (c) viel besser wiedergegeben, Löcher sind erhalten geblieben und die Spiralförmigkeit ist noch gut zu erkennen. Die Schrittweiterehöhung vernachlässigt Daten und zwingt die Knoten auch des einfacheren Netzes auf ein äquidistantes Gitter. Zudem sind über die Erhöhung der

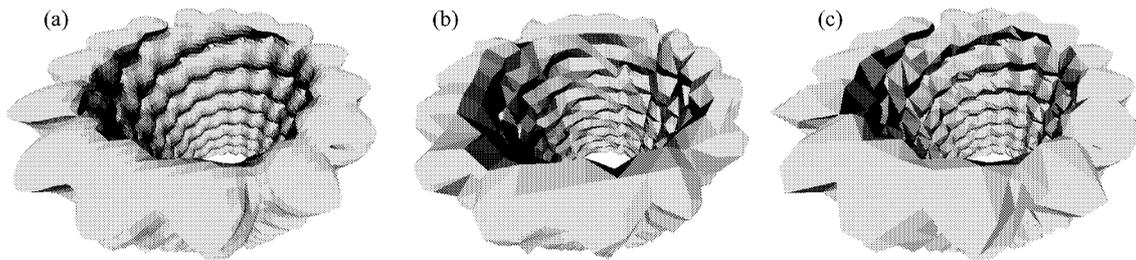


Abb. 4.1: (a) Eine mittels des Marching-Cube-Algorithmus aus Volumendaten gewonnene Oberfläche mit 60.278 Dreiecken bzw. 30.140 Knoten, die in (b) durch die Schrittweite des Marching-Cube-Algorithmus von drei und in (c) mit dem Verfahren dieser Arbeit auf 5.954 Dreiecke bzw. 2.978 Knoten vereinfacht wurde

Schrittweite nur diskrete Vereinfachungsstufen mit Vielfachen der originalen Knotenzahl von circa $\frac{1}{i^2}$, $i \in \mathbb{N}$ möglich. Die quadratische Verringerung liegt daran, dass eine Oberfläche gröber abgetastet wird. Weiche Übergänge zwischen Detailstufen durch eine kontinuierliche Datenstruktur oder Geomorphs sind wegen der fehlenden Detailstufen und der fehlenden Knotenbeziehungen der diskreten Stufen nicht möglich.

Es existieren aber auch adaptive Verfahren des Marching-Cube-Algorithmus, die bessere simplifizierte Netze hervorbringen [83]. Da sie aber auch nur die diskreten Volumendaten^G heranziehen können, kann die Qualität nicht besser als die der generellen Netzvereinfachungsverfahren werden. Zudem ist der Ansatz des implementierten Netzvereinfachungsalgorithmus genereller, da Dreiecksnetze verschiedener Herkunft behandelt werden können.

Die Visualisierung der Dreiecksnetze wie in Abbildung 4.1, aber auch sonstige Abbildungen der Netze aus der Anwendung, geschieht mittels der einfachsten Art der Schattierung^G, nämlich einer flachen Schattierung, bei der das gesamte Dreieck die gleiche Farbschattierung erhält. Bei einer geringen Zahl an Dreiecken, wo die unterschiedlichen Dreiecke schlecht zu unterscheiden sind oder wo es auf die Form und Lage der Dreiecke ankommt, wird manchmal ein zusätzliches Dreiecksgitter eingeblendet.

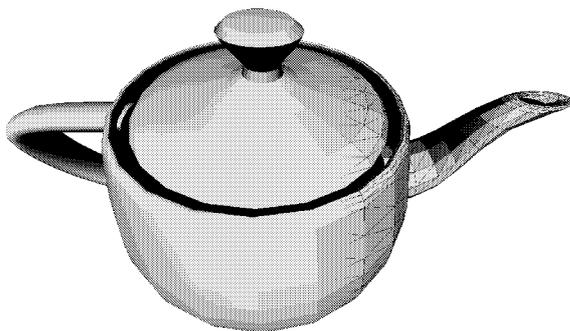


Abb. 4.2: Eine Kanne, die im linken Bereich mittels der Gouraud-Schattierung visualisiert wurde und sonst eine flache Schattierung besitzt; im rechten Bereich wurde zudem ein Gitter zur besseren Unterscheidung der Dreiecke voneinander eingeblendet

Anhand der Kanne aus Abbildung 4.2 ist gut zu erkennen, dass eine andere Wahl der Schattierung auch mit wenig Dreiecken eine glatte Oberfläche simulieren kann. Im rechten Bereich der Abbildung ist ein Dreiecksgitter auf die flache Schattierung des Netzes gelegt worden. Die flache Schattierung erstreckt sich weiter bis über die Mitte der Kanne. Der linke Abbildungsbereich ist mittels der Gouraud-Schattierung^G visualisiert worden, die eine gleichmäßig glatte Oberfläche simuliert, aber natürlich die Dreieckskanten nicht abrunden kann, wodurch die Silhouette entsprechend der Netzfeinheit eckig bleibt. Die Gouraud-Schattierung ist heutzutage oft in Hardware realisiert und damit nur unwesentlich langsamer als die flache Schattierung. Trotzdem macht in dieser

Arbeit eine flache Schattierung generell mehr Sinn, da zum einen die Vereinfachung viel besser zu erkennen und damit zu bewerten ist und zum anderen nicht bei jedem Objekt die Glattheit ein richtigeres visuelles Ergebnis liefert - warum sollte bei der Geländeoberfläche in Abbildung 2.20 ein

geglättetes Dreiecksnetz richtiger sein? Für die sinnvolle Nutzung der interpolierenden Schattierung^G ist zusätzliches Wissen über die Oberflächenbeschaffenheit des Netzes nötig.

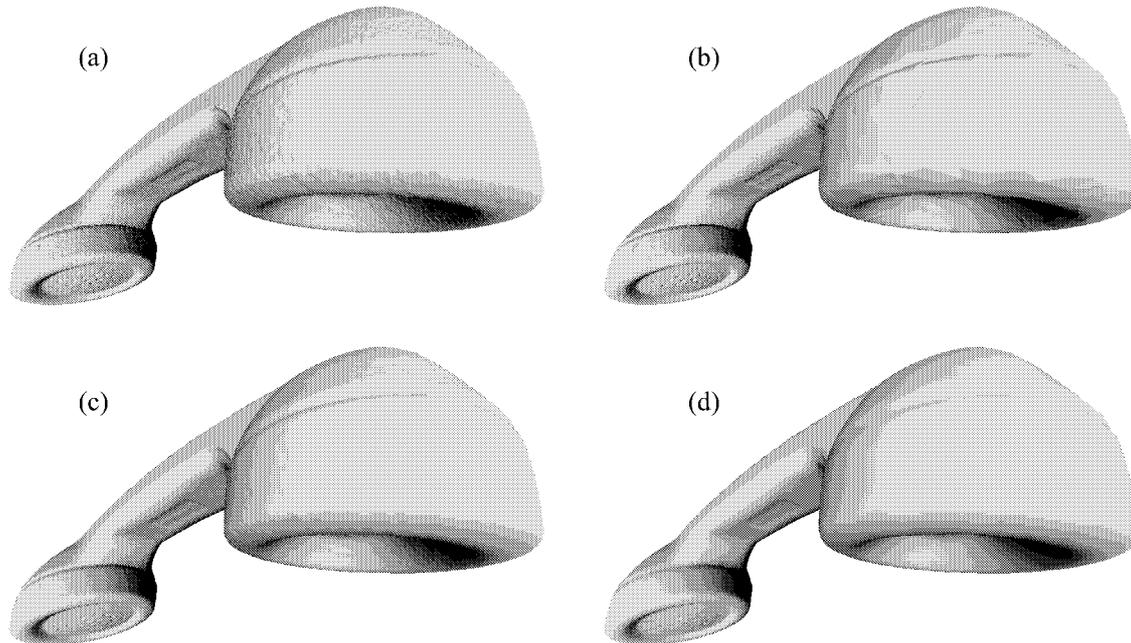


Abb. 4.3: Telefonhörer mit flacher Schattierung mit (a) originalen 83.044 und (b) 8.000 Knoten und mit Gouraud-Schattierung mit (c) 83.044 und (d) 8.000 Knoten

Den visuellen Unterschied zwischen Original und Vereinfachung würde eine geglättete Oberfläche mindern, was gut in Abbildung 4.3 zu erkennen ist. Die beiden linken Bilder stellen das Originalnetz (83.044 Knoten) mit flacher und mit Gouraud-Schattierung dar. Die Vereinfachung (8.000 Knoten) neben ihnen in gleicher Schattierung zeigt die direkte visuelle Differenz an, die bei der flachen Schattierung offensichtlich größer ist. Es hat sogar den Anschein, als hätte (d) das feinste Netz, da es am Glattesten wirkt. Diese Glattheit rührt von der Gouraud-Schattierung. Das Originalnetz hat durch Messungenauigkeiten eine etwas unruhigere Oberfläche. An den Löchern in der Sprechmuschel oder der Prägung zwischen den Muscheln ist aber auch zu erkennen, dass die Gouraud-Schattierung scharfe Details verwischt.

4.2 Geschwindigkeit

4.2.1 Zeitkomplexität

Zu Beginn der Geschwindigkeitsanalyse des implementierten Verfahrens soll die asymptotische Zeitkomplexität des Algorithmus abgeschätzt werden. Hierzu wird die O -Notation aus der praktischen Informatik genutzt [6].

Um sinnvolle Abschätzungen für reale Netze machen zu können, sind einige Voraussetzungen anzunehmen:

1. Die Anzahl der Dreiecke, die an einer Kante anliegen, sei durch eine konstante Größe beschränkt.
2. Ebenso sei die Anzahl von Kanten an einem Knoten durch eine konstante Größe beschränkt.
3. Der Schwellwert ist so gewählt, dass die Anzahl imaginärer Kanten nicht die Größenordnung der realen Netzkanten übersteigt.

Die erste Annahme ist nötig, da innerhalb des Algorithmus im Inneren von Schleifen auf alle Dreiecke zugegriffen wird, die an einer Kante liegen. Daher sollte sich die Größe unterhalb der Größenordnung der Knotenanzahl m bewegen. Für eine Mannigfaltigkeit beträgt diese Größe 2, es sind aber auch Nicht-Mannigfaltigkeiten mit mehr oder weniger Dreiecken an einer Kante erlaubt, vorausgesetzt die Anzahl liegt unterhalb der Komplexität von $O(m)$, was in nahezu allen in der Praxis vorkommenden Netzen der Fall ist. Von den später herangezogenen Netzen der praktischen Betrachtungen aus Tabelle 4.1 haben lediglich zwei Kanten des Drachenmodells mehr als 2 angrenzende Dreiecke. Betrachtet man die restliche Anzahl von Kanten gegenüber der Knotenanzahl, ist offensichtlich, dass es nur wenig mehr als 2 Dreiecke sein können, die an diesen beiden Kanten anliegen.

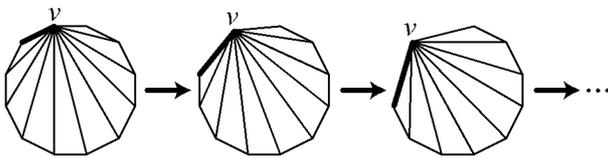


Abb. 4.4: Schlimmster Fall (worst case) für die Berechnungskosten während der Vereinfachung, da bei der Kontraktion der dicken Kante rund die Hälfte aller Netzkanten neu berechnet werden müssen

Die zweite Voraussetzung verhindert zum Beispiel Vereinfachungsfolgen entsprechend Abbildung 4.4, bei der in jedem Schritt durch die Kontraktion der dick gezeichneten Kante und der damit einhergehenden Verschiebung von v rund die Hälfte aller Kanten neu bewertet werden müssen. Es ist klar ersichtlich, dass alleine dadurch die Komplexität des Algorithmus bei der Abarbeitung der e Verschmelzungskandidaten mindestens auf $O(e^2)$ ansteigt, da für jede der $O(e)$ Kontraktionen eine Neubewertung

von $O(e)$ Kanten stattfinden muss. Bei anfänglich e Kanten ergibt sich die Anzahl der Neubewertungen während der gesamten Vereinfachung exakt über

$$\sum_{i=0}^{\frac{e-1}{2}} \frac{e-1}{2} - i = \sum_{i=0}^{\frac{e-1}{2}} i = \frac{e^2 - 1}{8} \in O(e^2),$$

da immer nur ein Dreieck, ein Knoten und eine Kante am Rand entfällt. Diese Voraussetzung scheint eine starke Einschränkung darzustellen, was aber nicht der Fall ist. Würden solche Netze häufig vorkommen, so könnte vor die Simplifizierung eine Regularisierung (siehe Abschnitt 2.7.2) geschaltet werden, die in $O(m)$ die Anzahl der Kanten an einem Knoten annähernd auf die schon in Abschnitt 2.7.2 beschriebene durchschnittliche Anzahl von 6 bringt. Zudem werden Kanten mit Knoten an vielen Dreiecken in dem nicht durch die Flächengröße gewichteten Quadrik-Fehlermaß besonders große Fehler zugewiesen, so dass sie in der Abarbeitungsliste der Kontraktionspaare hinten stehen. In der Praxis sind solche Fälle bei den in dieser Arbeit herangezogenen Modellen nicht aufgetaucht und können durch den Marching-Cube-Algorithmus auch nicht erzeugt werden, da laut Abbildung A.2 des Abschnitts A.1.4 in jedem Voxel^G an den Knoten maximal 4 Kanten anliegen. An einer Voxelkante stoßen 4 Voxel aufeinander. Wandert der Knoten an eine Voxelcke so sind es in Ausnahmefällen auch 8 Voxel, von denen vom Knoten Netzkanten ausgehen. Benachbarte Voxel haben in diesen Fällen immer eine gemeinsame Kante, so dass ein Knoten in Netzen, die mit dem Marching-Cube-Algorithmus erzeugt wurden, maximal $3 \cdot 8 = 24$ Kanten besitzen kann.

Die letzte Annahme wurde schon in Abschnitt 3.1.2 motiviert. Dort wurde erwähnt, dass die Anzahl der Knotenverschmelzungskandidaten bei großem Schwellwert auf $O(m^2)$ ansteigt, so dass die Gesamtkomplexität um den Faktor m^2/e erhöht würde. Sinnvolle Größen des Schwellwertes fügen den realen Kanten imaginäre Kanten in der Größenordnung von $O(m)$ hinzu, da entfernte Kandidatenpaare unerwünschte Effekte während der Vereinfachung bewirken. Die Gesamtkomplexität bleibt somit bei vernünftiger Wahl des Schwellwertes bei $O(m)$. Auswirkungen des Schwellwertes auf die Vereinfachung werden im Abschnitt 4.4.3 untersucht.

Die einzige variable Größe, die in die Abschätzung der Komplexität einzugehen braucht, ist die Knotenzahl m , da durch die oben gemachten Einschränkungen beliebig entartete Netze ausgeschlossen sind, die nicht annähernd die Eulersche Formel aus Abschnitt 2.1.1 erfüllen. Danach gilt ungefähr, dass ein Netz zweimal so viele Dreiecke t und dreimal so viele Kanten e wie Knoten enthält:

$$\frac{t}{m} \approx 2 \quad \wedge \quad \frac{e}{m} \approx 3 \quad \Rightarrow \quad O(m) = O(f) = O(e).$$

Für die Modelle aus Tabelle 4.1 ist die Abweichung des Verhältnisses der Dreiecke und Kanten zu den Knoten trotz einiger Randdreiecke, wie in Abbildung 4.5 zu sehen, eher gering. Die Abweichung ist maximal 0,06 bei dem Blatt und wird tendenziell kleiner je größer die Netze werden.

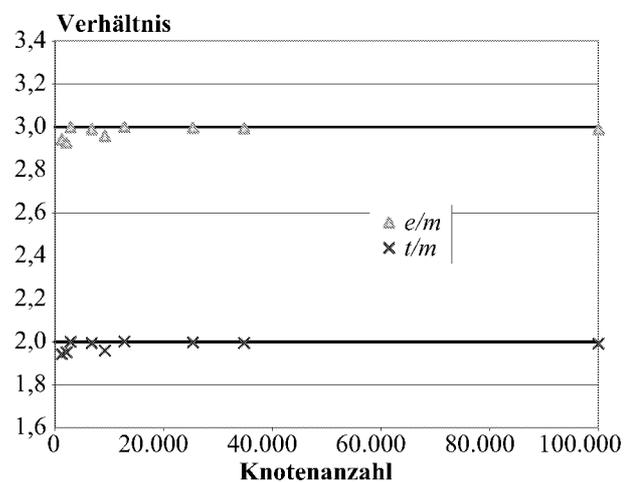


Abb. 4.5: Verhältnis der Dreiecke t und Kanten e zu den Knoten m eines Dreiecksnetzes

Das implementierte Verfahren lässt sich in drei Teile gliedern:

1. Initialisierung der zur Simplifizierung benötigten Daten (Punkt 1 bis 4 der Zusammenfassung des Algorithmus aus Abschnitt 3.1.8)
2. Fortschreitende Reduzierung der Netzkomplexität (Punkt 5 der Zusammenfassung)
3. Anpassung der Detailstufe des Netzes an Randbedingungen (Punkt 6 der Zusammenfassung)

Initialisierung:

In der Initialisierungsphase wird an jedem Knoten seine Quadrik durch die anliegenden Dreiecke bestimmt. Da die Anzahl an Kanten an einem Knoten begrenzt ist, ist auch die Anzahl anliegenden Dreiecke konstant, so dass für jeden Knoten die Bestimmung der initialen Quadrik konstant ist. Für alle Knoten ergibt sich somit die Zeitkomplexität von $O(m)$.

Durch die dritte Annahme bleibt die Anzahl der Kanten durch die Hinzunahme der imaginären Kanten bei $O(m)$. Die Bestimmung der Knotenverschmelzungskandidaten, insbesondere der imaginären Kanten, erfolgt durch die Einschränkung der Anzahl laut [6] in $O(m \log m)$.

Die Bestrafung von Randkanten bedarf eines Durchlaufs über alle Kanten. An jeder Kante muss die Anzahl anliegender Dreiecke gezählt werden. Durch die Voraussetzung von Punkt eins ist die Anzahl an Dreiecken an einer Kante konstant und dadurch die gesamte Bestrafung in $O(m)$ zu erledigen.

Die Bestimmung des optimalen Verschmelzungspunktes und die Kosten an diesem Punkt durch die Heranziehung der Quadriken beider Kantenknoten ist in konstanter Zeit zu erledigen. Für alle Knotenpaare ist die Zeitkomplexität dieses Teilschritts somit $O(m)$.

Als letzter Schritt der Initialisierungen ist eine Sortierung der Verschmelzungskandidaten nach Kosten ihrer Kontraktion nötig, damit eine Entnahme der kostengünstigsten Kontraktion in der nächsten Phase in konstanter Zeit möglich ist. Eine Sortierung von m Elementen ist in $O(m \log m)$ möglich [18].

Die gesamte Initialisierungsphase ist damit in $O(m \log m)$ abgeschlossen, da die oben aufgelisteten Schritte nacheinander ausgeführt werden.

Fortschreitende Knotenverschmelzung:

Die Phase der Simplifizierung entnimmt in konstanter Zeit sukzessive alle $e \in O(m)$ Verschmelzungskandidaten und führt die entsprechende Kontraktion aus, wenn sie nicht unerwünschte Zustände herbeiführt. Die wirklich abzuarbeitende Zahl an Kontraktionen beläuft sich ohne imaginäre Kanten exakt auf $m - z$, da während einer Kontraktion dem Netz immer genau ein Knoten und um die zwei Kanten entnommen werden. z ist die Zahl an Zusammenhangskomponenten, die der Anzahl an im Basisnetz ohne Dreiecksflächen übrig bleibenden nicht weiter zu vereinfachenden Netzknotten entspricht. Handelt es sich bei dem Netz um ein Dreiecksnetz ohne unverbundene Knoten, so sind maximal $z = t$ Zusammenhangskomponenten möglich, wenn das Netz nur aus unverbundenen Dreiecken besteht. Die Knotenzahl bei solch einem Netz beträgt $m = 3t$ und damit liegt die Anzahl zu kontrahierender Kanten wie für ein gewöhnliches Netz in $m - z = 2t \in O(m)$.

Die Rückstellung einer Verschmelzung ist implementiert worden, falls es zu identischen Dreiecken bei der Kontraktion kommt oder dabei die Normalen eines Dreiecks eine zu große Richtungsänderung erfährt. Die Häufigkeit der Rückstellung ist aber für jedes Kandidatenpaar durch eine konstante Zahl begrenzt, so dass sie keinen Einfluss auf die Komplexität hat, im ungünstigen Fall nur einen hohen konstanten Geschwindigkeitsfaktor der Berechnung hinzufügt.

Die Kontraktion selbst bedarf dem Besuch aller Dreiecke und Kanten der Kantenknoten, was durch die Vorbedingungen in konstanter Zeit möglich ist. Die Bestimmung des Kontraktionspunktes und der dazugehörigen Kosten aller sich verändernder Kanten ist auch in konstanter Zeit zu erledigen. Allein das Neueinsortieren der veränderten Kanten in die nach Kosten sortierte Kantenmenge ist zeitkritisch. Würde wie bisher immer benannt, die Kantenmenge über eine Liste realisiert sein, so würde das sortierte Einfügen eines Elementes in die Liste $O(m)$ Zeit beanspruchen, so dass die Gesamtlaufzeit der fortschreitenden Knotenverschmelzung in $O(m^2)$ läge. Wie ausführlicher im Anhang A.2.2 beschrieben, ist in der Implementation statt der Liste ein Suchbaum eingesetzt worden, der das sortierte Einfügen eines Elementes in $O(\log m)$ erlaubt und somit das gesamte Neueinsortieren in $O(m \log m)$ ermöglicht. Genauer betrachtet, nimmt der Aufwand für das Einsortieren fortlaufend ab, so dass sich für den Normalfall eines verschwindenden Knotens während einer Kontraktion der Gesamtaufwand

$$\log m + \log(m - 1) + \log(m - 2) + \cdots + \log 1 = \sum_{i=1}^m \log i$$

ergibt, wenn am Ende genau ein Knoten übrig bleibt. Für mehr gleichzeitig verschwindende Knoten und mehr Restknoten wird die Summe größer, so dass die obige Summe eine obere Schranke für den Gesamtaufwand darstellt. Der konstante Faktor für die Anzahl neu einzusortierender Kanten ist

weggelassen worden, da die endliche Summe nur einer Abschätzung der Größenordnung dient. Ferner gilt

$$\sum_{i=1}^m \log i = \log m! \in O(m \log m),$$

was behauptet wurde.

Das Hinzufügen einer neuen Progressive Mesh-Stufe ist wie das im nächsten Schritt beschriebene Auslesen einer Stufe in konstanter Zeit möglich.

Die gesamte Phase der fortschreitenden Knotenverschmelzung hat somit insgesamt eine Laufzeitkomplexität von $O(m \log m)$.

Da die beiden bisherigen Phasen in $O(m \log m)$ durchzuführen sind und hintereinander ausgeführt werden, ist der gesamte Vorgang der Simplifizierung mit der dazugehörigen Dateninitialisierung in

$$O(m \log m)$$

möglich. Die Dauer der Simplifizierung wächst also nicht übermäßig mehr als linear und damit nicht überproportional für große Netze.

Anpassung der Detailstufen:

Die Anpassung der Detailstufe des Netzes an vorgegebene Randbedingungen wird online durchgeführt und kann daher konventionell nicht abgeschätzt werden. Das während der Vereinfachung erstellte Progressive Mesh ist eine Liste, die in beide Richtungen durchlaufen werden kann, je nachdem, ob eine Vereinfachung oder Verfeinerung gewünscht ist. Die Umsetzung eines Elementes des PM ist in konstanter Zeit möglich, da laut Vorgaben eine konstante Zahl an Dreiecken an einer Kante anliegt, die hinzugefügt bzw. entfernt wird, und jeder Knoten ebenso einer konstanten Anzahl an Kanten und damit auch Dreiecken angehört, die aktualisiert werden müssen.

Nach der Simplifizierung ist das Netz radikal auf wenige Knoten reduziert. Um den ursprünglichen Zustand des Originalnetzes herzustellen, muss das komplette PM durchlaufen werden. Dies ist somit in $O(m)$ möglich, da es wegen der $O(m)$ Verschmelzungskandidaten auch $O(m)$ PM-Stufen gibt. Ebenso ist bei der Annahme einer idealisierten Maschine ohne Lastschwankungen und einer einmaligen Vorgabe einer Randbedingung jede Stelle des PM in $O(m)$ anwählbar, um die Randbedingung zu erfüllen oder bei nicht möglicher Erfüllung der Bedingungen das Netz auf die minimal oder maximal mögliche Knotenanzahl einzustellen.

4.2.2 Modelldaten

Um empirische Aussagen über das implementierte Verfahren machen zu können, werden in den folgenden Abschnitten Modelldaten herangezogen, die in den Tabellen 4.1 bzw. 4.2 aufgeführt sind. Alle Werte und die daraus resultierenden Ergebnisse sind auf dem im Anhang A.1.5 beschriebenen Rechner erzielt worden.

Jedes Modell besitzt Abbildungen einiger Detailstufen innerhalb dieser Arbeit. Die Reihenfolge der Modelle innerhalb der Tabellen entspricht der Reihenfolge, in der die Abbildungen in dieser Ausarbeitung vorkommen. Der Name der Modells steht in der ersten Spalte der Tabellen, die Abbildungsnummer in Spalte zwei der ersten Tabelle.

Modell	Abb.	Knoten	Dreiecke	Kanten			Zus.- komp.	Hauptspeicher [kB]	
				1	2	> 2		Netz	PM
Gelände	2.20	100.000	199.114	884	298.229	0	1	61.028	28.352
Kuh	3.5	2.903	5.804	0	8.706	0	1	1.860	1.184
Wölbung	3.7	9.216	18.050	380	26.885	0	1	5.660	2.952
Blatt	3.8	1.252	2.432	70	3.613	0	1	836	876
Hase	4.12	34.834	69.451	223	104.065	0	1	21.604	9.856
Flugzeug	4.13	6.795	13.546	0	20.319	0	11	4.236	2.248
Dichtewolke	4.14	12.835	25.680	8	38.507	0	4	7.881	4.170
Drachen	4.16	25.418	50.761	69	76.104	2	3	15.596	8.400
Knochen	4.23	2.154	4.204	0	6.306	0	26	1.384	944

Tab. 4.1: Übersicht von Eigenschaften zur Bewertung ausgewählter Modelle

Modell	Geschwindigkeit [s]			Bestrafung		Quadrik (Init.)		Quadrik (Simpl.)	
	Init.	Reduz.	Reset	Ident.	Dreh.	sing.	regulär	sing.	regulär
Gelände	2,584	7,933	1,713	10	603	6.881	292.232	291	696.087
Kuh	0,050	0,170	0,040	197	14	0	8.706	0	20.420
Wölbung	0,171	0,510	0,140	2	326	26.133	1.132	28.941	35.292
Blatt	0,020	0,060	0,010	3	9	389	3.294	147	8.364
Hase	0,861	2,672	0,601	20	115	1.162	103.126	138	246.333
Flugzeug	0,130	0,440	0,100	513	250	1.536	18.783	582	47933
Dichtewolke	0,300	0,842	0,200	52	62	362	38.153	14	90.384
Drachen	0,511	1,723	0,441	132	95	1.165	75.010	2.401	174.256
Knochen	0,040	0,100	0,020	118	5	0	6.306	0	13.204

Tab. 4.2: Übersicht von Modelleigenschaften während der Simplifizierung

Der Rest der Tabelle 4.1 gibt Auskunft über allgemeine Informationen zu den Netzen der Modelle. Es wird die Anzahl der Knoten, Dreiecke und Kanten des Dreiecksnetzes aufgezählt. Bei den Kanten wird unterschieden, an wie vielen Dreiecken die Kante anliegt. Unterhalb der Zahl 1 ist also zu entnehmen, wieviele randständige Kanten im Netz vorkommen und somit wenn gewünscht eine Randbestrafung erfahren, da die Kante nur an einem Dreieck anliegt. Die Dichtewolke der Anwendung besitzt 8 Randkanten, da der Marching-Cube-Algorithmus auch Mannigfaltigkeiten mit Rand erzeugen kann, wenn die Isofläche bis an den Rand der Volumendaten^G reicht. Die Zahlen unterhalb von “> 2” zeigen an, wie viele Kanten in mehr als zwei Dreiecken vorkommen. Dies sind nur 2 im Drachenmodell, an deren Stelle das Netz entartet ist. Diese entarteten Bereiche des Dreiecksnetzes zeigen, dass das Verfahren sich unkritisch gegenüber solchen Stellen verhält, da auch dieses Modell verarbeitet wird.

Die Werte in der Tabelle oder in den folgenden Abschnitten sind alle, wenn nicht anders gesagt, mit eingeschalteter Bestrafung für alle Einschränkungen des Vereinfachungsverfahrens angegeben. Es werden Randdreiecke mit senkrechten Ebenen versehen und Kontraktionen, die sich drehende oder identische Dreiecke erzeugen, zurückgestellt.

Die Spalte Zusammenhangskomponenten zählt die Anzahl von Teilnetzen, aus denen das gesamte Modell besteht. Auch die Dichtewolke der Anwendung besitzt mehrere Zusammenhangskomponenten, da der Marching-Cube-Algorithmus mehrere unabhängige Isoflächen erzeugen kann. Außer bei

der expliziten Untersuchung des Schwellwertes ist der Schwellwert bei allen folgenden Betrachtungen Null, so dass die Komponenten während der gesamten Simplifizierung erhalten bleiben und nicht verschmelzen.

Die letzten beiden Spalten der ersten Tabelle geben den Speicherbedarf in Kilobyte für das reine Netz und für das Progressive Mesh an, wobei das Progressive Mesh den Speicherbereich des Ursprungsnetzes mitbenutzt, indem die eigentlichen Netzknoten und -dreiecke im eigentlichen Netz verbleiben und nur die Änderungen jeder Detailstufe im PM abgelegt werden. Dabei ist zu beachten, dass das PM alle Netzdaten enthält, jedoch zur Erzeugung eines nutzbaren Netzes zusätzlich Knoten- und Dreiecksvektoren benötigt.

Die Modelle entstammen extra verschiedener Herkunft. Die für die eigentliche Anwendung wichtige Isofläche einer Elektronendichtewolke vertritt das Modell Dichtewolke in der vorletzten Zeile der Tabellen. Ein Höhenfeld einer Satellitenaufnahme stellt das Gelände dar. Die Kuh, der Drache und der Hase stammen von 3D-Scannern, wobei der Drache am Fuß verändert und durch eine konstruierte Ebene erweitert wurde, was auch die 3 Zusammenhangskomponenten erklärt. Konstruierte Modelle stellen die Wölbung, das Blatt und das Flugzeug dar. Das Modell Knochen der letzten Tabellenzeile entstammt einer Tomographie-Aufnahme in der Medizin, ist also auch eine Isofläche eines Volumendatensatzes. Dass nicht nur Isoflächen von Elektronendichtewolken in dieser Arbeit herangezogen wurden, liegt neben dem Beweis der Generalität des Ansatzes insbesondere daran, dass optische Qualitätsunterschiede besser an real bekannten Objekten zu erkennen sind.

Die maximale Modellausdehnung ist für alle diesbezüglich kritischen Werte für alle Modelle auf 2 gesetzt. Der Hase hat also die gleiche Größe wie das Flugzeug.

Die Tabelle 4.2 enthält Werte zur Simplifizierung der jeweiligen Modelle. Die Geschwindigkeit in Sekunden ist für drei Phasen angegeben, die ungefähr mit den Phasen der Betrachtung der Zeitkomplexität übereinstimmen. Die erste Geschwindigkeitsspalte enthält die Zeit, die für die Initialisierung gebraucht wird, die zweite die Zeit, die die fortschreitende Knotenschmelzung beansprucht. Diese beiden Schritte stellen die eigentliche Simplifizierung dar. Die dritte Spalte zur Geschwindigkeit ist angegeben, da in der Implementation nach der Simplifizierung der nur zur Vereinfachung benötigte Speicherplatz freigegeben wird und das bis auf einige Knoten reduzierte Netz über das Progressive Mesh in den ursprünglichen Zustand zurückgeführt wird. Die Zeit der Spalte Reset gibt also insbesondere Auskunft über die Zeit, die ein kompletter Durchlauf durch das PM braucht.

Die beiden Spalten der Bestrafung geben an, wie häufig eine Kante wegen der Identität eines Dreiecks oder einer Drehung eines Dreiecks zurückgestellt wurde. Die Zahlen können auch dieselbe Kante mehrmals berücksichtigen. Bei Mannigfaltigkeiten kommt es zum Ende immer zu identischen Dreiecken.

Die letzten vier Tabellenspalten geben die Anzahl und Art der Quadriken an, aufgeschlüsselt nach der Phase (Initialisierung oder fortschreitende Knotenschmelzung), in der sie entstehen, und danach, ob sie als singular oder regulär eingestuft wurden. Für reguläre Quadriken wird der optimale Verschmelzungspunkt bestimmt, für singuläre Quadriken einfach der beste dreier ausgewählter Punkte genommen.

4.2.3 Empirische Laufzeiten

In diesem Abschnitt sollen die realen Laufzeiten der Implementation untersucht werden. Eine theoretische Betrachtung ist bereits in Abschnitt 4.2.1 vorgestellt worden.

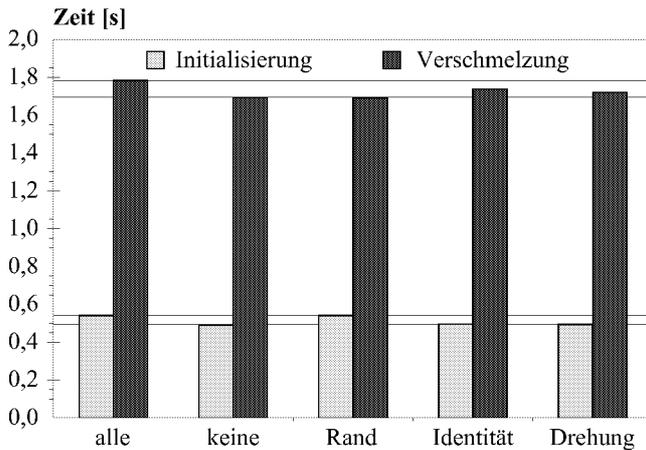


Abb. 4.6: Zeitunterschiede für das Drachenmodell für die Initialisierungs- und Verschmelzungsphase aufgrund unterschiedlich aktivierter Bestrafungen

Anhand des nebenstehenden Diagramms 4.6 soll für das Drachenmodell gezeigt werden, wie sich die einzelnen Bestrafungen aus Abschnitt 3.1.5 auf die Laufzeit der Initialisierungs- und Verschmelzungsphase, also die eigentliche Simplifizierung, auswirkt. Die Auswirkung ist linear zur Knotenanzahl, so dass die Ergebnisse leicht auf andere Modelle übertragbar sind. Die maximale Anzahl an zurückgestellten und damit doppelt zu bewertenden Verschmelzungskandidaten liegt für alle Modelle beim Flugzeug mit 3,76% der Kanten vor, was die Geschwindigkeit in gleichem Maße nur unwesentlich beeinflusst. Die Anzahl an Doppelbewertungen ist für den Hasen mit 0,13% am geringsten.

Das Diagramm hat die Dauer für die Initialisierung und Verschmelzung getrennt aufgetragen. Die Messung wurde mit allen und keiner Bestrafung wie für jede separat aktivierte Bestrafung durchgeführt. Da die Randbestrafung alleine in der Initialisierungsphase durch zusätzliche Quadriken für die Randknoten des Originalnetzes durchgeführt wird und in dieser Phase die beiden anderen Bestrafungen nicht zum Tragen kommen, ist, wie gut zu erkennen, nur die Randbestrafung für die etwas längere Initialisierungszeit verantwortlich. Dafür wirkt sich die Randbestrafung nicht auf die Zeit für die fortschreitende Verschmelzung aus. Die Zeitdifferenz zwischen keiner und aller eingeschalteten Bestrafungen ist dort auf die Überprüfung der Dreiecksidentität und der Dreiecksdrehung zurückzuführen, was nur während der Verschmelzung überprüft werden kann. Die Bestrafungen sind nicht unabhängig voneinander, so sind laut Tabelle 4.2 bei komplett aktivierten Bestrafungen 95 Dreiecksdrehungen und 134 Dreiecksidentitäten zurückgestellt worden. Schaltet man die Überprüfungen getrennt an, so ergeben sich 96 Dreiecksdrehungen und 194 Dreiecksidentitäten, die es zu berücksichtigen galt. Die gesamte Zeitdifferenz der Simplifizierung mit und ohne Bestrafungen liegt bei rund 6% und liegt damit in einem vertretbaren Zeitrahmen im Verhältnis zum Nutzen.

Bei allen weiteren Messungen sind alle Bestrafungen eingeschaltet gewesen, da dies das realistischste Szenario ist, weil dadurch für gewöhnlich visuell bessere Ergebnisse erzielt werden. Die Zeiten beruhen auf mehreren Messungen, von denen jeweils das beste Ergebnis, also die schnellste Zeit, genommen wurde. Leider sind die Schwankungen der Messungen nicht unerheblich, so dass Ungenauigkeiten in den Diagrammen darauf zurückgeführt werden können. Einen starken Einfluss auf die Geschwindigkeit hat neben dem Compiler und seinen Optimierungsmöglichkeiten auch durch die Nutzung der C++-Standard-Musterklassen-Bibliothek (STL, Standard Template Library) deren Implementation. Die beim Microsoft Visual C++-Compiler mitgelieferte STL-Implementation war fehlerbehaftet und auch nach eigener Beseitigung der Fehler um 30 – 40% langsamer als die freie STL-Implementation STLport [84], die daher auch eingesetzt wurde.

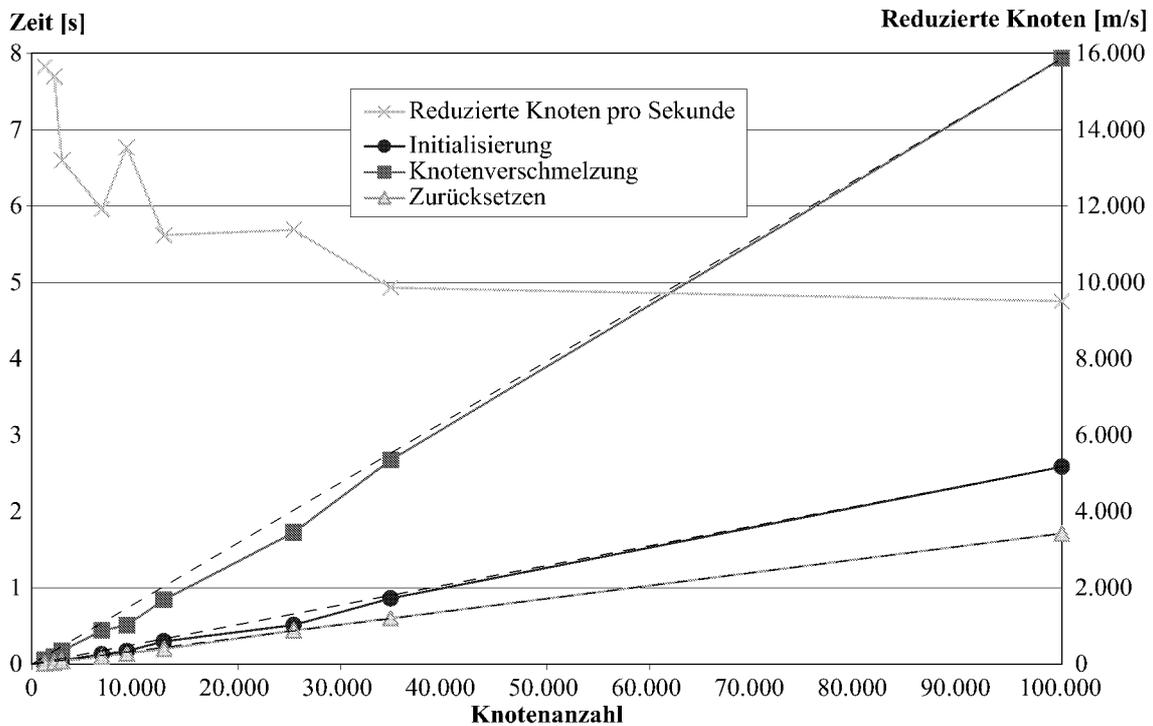


Abb. 4.7: Geschwindigkeit der einzelnen Phasen der Vereinfachung

In Diagramm 4.7 sind die in Tabelle 4.2 eingetragenen Zeitwerte für die Initialisierung, Knotenverschmelzung und das Zurücksetzen auf die Knotenanzahl aufgetragen. Ein linearer Anstieg der Zeit ist jeweils durch eine gestrichelte Linie gekennzeichnet. Dadurch ist gut zu erkennen, dass das Zurücksetzen wie in der Theorie linear zur Knotenanzahl möglich ist. Die anderen beiden Zeiten zeigen wie ebenfalls in der Theorie ihren hinzukommenden logarithmischen Charakter, da bei steigender Netzknotenanzahl die Zeit überproportional wächst. Noch etwas besser ist der Abfall der pro Sekunde simplifizierten Netzknoten bei steigender Netzgröße an der Kurve zu erkennen, die auf der rechten Achse aufgetragen die Anzahl der Knotenreduktionen pro Sekunde beschreibt. Die Reduktionszeit beinhaltet dabei die Zeit der Initialisierung und der Verschmelzung. Die Kurve fällt entsprechend dem Laufzeitverhalten beider Phasen von $O(m \log m)$, ist aber im für die Anwendung interessanten Bereich nur leicht abfallend und liegt um 10.000 Knotenreduktionen pro Sekunde. Die 10.000 Knotenreduktionen pro Sekunde liegen über dem Vergleichswert der Einführung in Tabelle 2.1, da für den Verfahrenvergleich Werte aus den ursprünglichen Arbeiten herangezogen wurden. Für das Quadrik-Fehlermaß ist der auf den Referenzrechner A.1.5 übertragene Wert 11.300 Dreiecke, also rund 5.650 Knoten, pro Sekunde und damit fast um die Hälfte kleiner als der Wert der eigenen Implementation.

Die Initialisierung beansprucht rund ein Viertel der Zeit der gesamten Simplifizierung und die Knotenverschmelzung die restlichen drei Viertel der Zeit. Das Verhältnis ist somit ungefähr eins zu drei. Das Zurücksetzen des komplett simplifizierten Netzes auf das Originalnetz verläuft linear zur Knotenanzahl und hat einen Durchsatz von über 53.000 Schritten pro Sekunde. Dieser Durchsatz rechtfertigt schon die Speicherung eines Progressive Meshes für nur zwei einzustellende Detailstufen, da dies über das PM schneller zu realisieren ist als über die Neuberechnung einer bestimmten Stufe mittels der Simplifizierung.

Im Abschnitt 4.4.1 über die Netzqualität wird die Qualitätssteigerung aufgrund des optimalen Verschmelzungspunktes gegenüber einer Punktauswahl aus den Kantenendpunkten und deren Mittelpunkt beschrieben. Bei der Simplifizierung ohne optimalen Verschmelzungspunkt lässt es einen im ersten Augenblick verwundern, dass das Verfahren länger braucht. Dies ist aber darauf zurückzuführen, dass für die nicht optimale Verschmelzung drei Punkte bewertet werden müssen statt nur einem beim optimalen Verfahren. Für die Bestimmung des optimalen Verschmelzungspunktes sind gemäß Abschnitt 3.1.4 weniger Operationen nötig als die 21 Multiplikationen und 9 Additionen der Bewertung über das Quadrik-Fehlermaß mit $v^t Qv$. Dadurch ergibt sich eine Verlangsamung um circa 7,5% bei der Verschmelzung ohne optimalen Punkt.

Werden Modelle simplifiziert, die für sich oder während der Berechnung nicht in den Hauptspeicher passen, so wird das Verfahren sehr langsam, da die Speicherlokalität gering ist und es daher oft zu zeitintensiven Auslagerungen von Hauptspeichereinhalten in den virtuellen Speicher auf der Platte und umgekehrt (swapping) kommt. Die Vereinfachung wird beliebig im Netz durchgeführt und wandert eher als dass sie an einer Stelle mehrfach durchgeführt wird, da außer in flachen Regionen eine Verschmelzung meist die Oberflächenkrümmung und damit den Quadrik-Fehler erhöht. In Abschnitt 2.7.1 wird auf Verfahren eingegangen, die für speicherkritische Modelle gedacht sind.

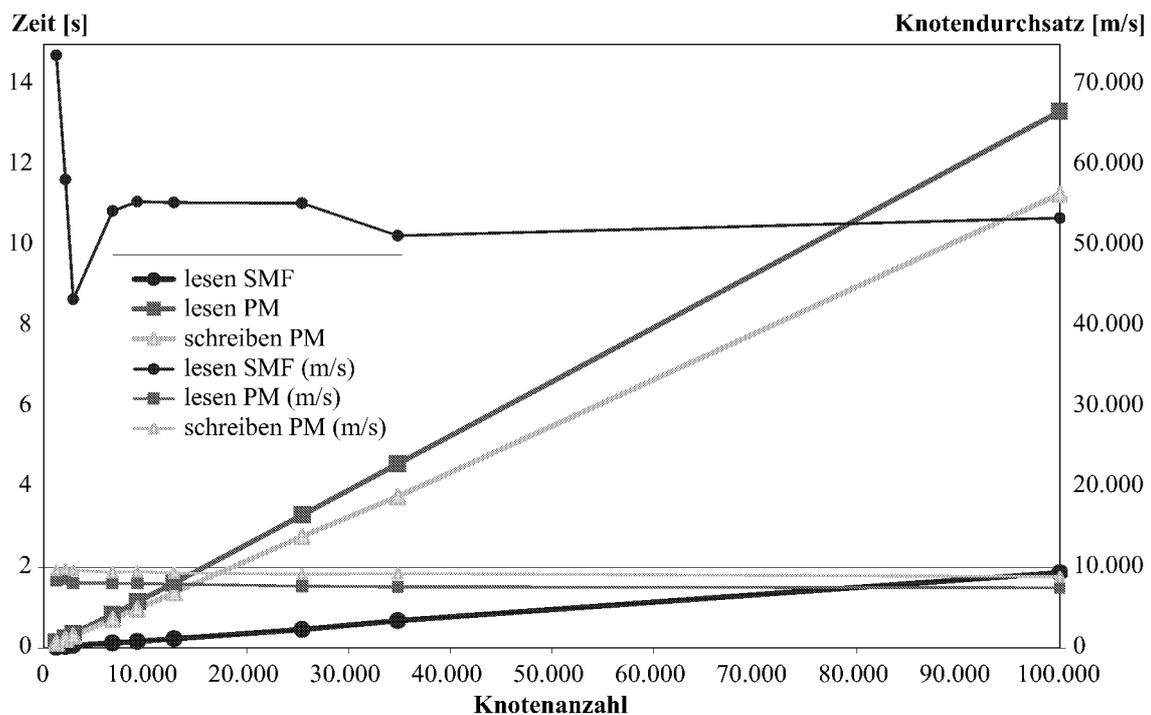


Abb. 4.8: Geschwindigkeit des Lesens und Schreibens der Dreiecksnetze in den Formaten SMF und PM

Ein letzter Aspekt der Geschwindigkeit ist die Zeit für das Einlesen und Ausgeben der Netzdaten. Diagramm 4.8 führt auf der linken Zeitachse die absoluten Werte auf, die zum Lesen der Netzformate SMF und PM, sowie zum Schreiben des PM-Formates nötig sind. Auf der rechten Achse sind die gleichen Vorgänge als relative Werte des Knotendurchsatzes aufgetragen. Das SMF-Format [34] ist ein freies, verbreitetes Format zur Speicherung von Netzdaten. Das PM-Format ist ein eigenes Format zur Sicherung des berechneten Progressive Meshes.

Die Zeit für das Einlesen einer SMF-Datei ist mit einem Durchsatz von rund 55.000 Knoten pro Se-

kunde möglich. Zum Einlesen gehört dabei auch die Überprüfung und wenn nötig Korrektur des Netzes, die Normalisierung und die Normalenberechnung, da diese Daten im Progressive Mesh enthalten sind. Der Ausreißer des dritten Wertes für die SMF-Datei der Kuh ist mit der noch zu berichtigenen Netzkonsistenz zu begründen, da Knoten im Netz enthalten sind, die nicht benötigt werden. Der Durchsatz bei dem PM-Format liegt durch die Komplexität des PM-Datenformates (siehe Abschnitt A.2.2) noch leicht unterhalb der Berechnungsgeschwindigkeit der Simplifizierung. Aus diesem Grund lohnt es sich nicht, die Vereinfachung zu speichern, wenn das Dreiecksnetz schnell wieder herzustellen ist, da die Neuberechnung der Vereinfachung schneller geht als das Einlesen derselben. Anders sieht es aber aus, wenn das Netz erst aufwendig berechnet werden muss. Dies ist genau in der zugrundeliegenden Anwendung der Fall, bei der erst durch den Marching-Cube-Algorithmus das Netz der Isofläche berechnet werden muss, bevor es vereinfacht werden kann. Daher wird in der Anwendung angeboten, das Netz entsprechend der Schrittweiten- und Isoflächen-Einstellung für den Marching-Cube-Algorithmus nach der Simplifizierung automatisiert wegzuschreiben, und, wenn bereits vorhanden, ohne Ausführung des Marching-Cube-Algorithmus schon vereinfacht über das PM-Format einzulesen. Der leicht abfallende Durchsatz hängt damit zusammen, dass beide Formate nicht binär sondern Text-basiert vorliegen, und somit große ganzzahlige Werte, wie sie zum Beispiel für die Knotenindizes benutzt werden, mehr Platz beanspruchen als kleine und dadurch eine größere zu übertragene Datenmenge verursachen. Beim SMF-Format kommt hinzu, dass die Daten aus unterschiedlichen Quellen stammen und somit in unterschiedlicher Genauigkeit und stark schwankendem Platz für die Zwischenräume vorliegen.

Warum das Lesen langsamer verläuft als das Schreiben ist schwer zu analysieren und stark vom Betriebssystem und Plattensystem abhängig. Durch vorausschauendes Lesen oder nicht zwischengespeichertes Schreiben ist auch eine andere Reihenfolge denkbar.

4.3 Speicherplatz

Dieser Abschnitt befasst sich mit dem Speicherplatzbedarf des Verfahrens. Da die in der Anwendung vorkommenden Netzgrößen nicht an Speichergrenzen heute aktueller Rechner stoßen, ist das Verfahren dahingehend nicht optimiert worden. Ist ein Datum zur schnelleren Berechnung oder Visualisierung nötig, so ist es den Datenstrukturen hinzugefügt worden. Ein Beispiel für Daten, die zur schnelleren Berechnung hinzugefügt wurden, stellen die Referenzlisten auf Kanten und Dreiecke in den Knoten dar, mit deren Hilfe eine schnellere Aktualisierung der jeweiligen Datenstruktur während der Vereinfachung bzw. Anpassung einer Detailstufe möglich ist. Ein Beispiel für Daten zur schnelleren Visualisierung stellt die Speicherung der Normalen an den Dreiecken und Knoten dar, die zwar aus den vorhandenen Daten immer wieder neu berechnet werden könnten, das Vorhalten der Normalen die Visualisierung aber beschleunigt. Ein weiterer Punkt, bei dem sich für die speicherintensivere Lösung entschieden wurde, ist die Wahl des Verschmelzungskandidaten, der auf den optimalen Punkt fiel. Eine Begründung dafür ist im letzten Absatz des Abschnitts 3.1.4 zu finden. Die bessere Qualität der optimalen Wahl wird in Abschnitt 4.4.1 belegt.

Eine einfache Skizze der wichtigsten Datenstrukturen mit ihren wesentlichen Komponenten findet der Leser im Anhang A.2.2. Das Diagramm in Abbildung 4.9 zeigt den nahezu linearen Speicherplatzbedarf der vier wichtigsten Datenstrukturen, dem Knoten- und Dreiecksvektor, der Kantenliste und dem Progressive Mesh. Auf der linken Achse sind die absoluten Zahlen und auf der rechten Achse ist der im Durchschnitt für jedes Element der Struktur benötigte Speicherplatz aufgetragen, wobei das Progressive Mesh durch die für gewöhnlich entsprechende Knotenzahl geteilt wurde. Die Linearität

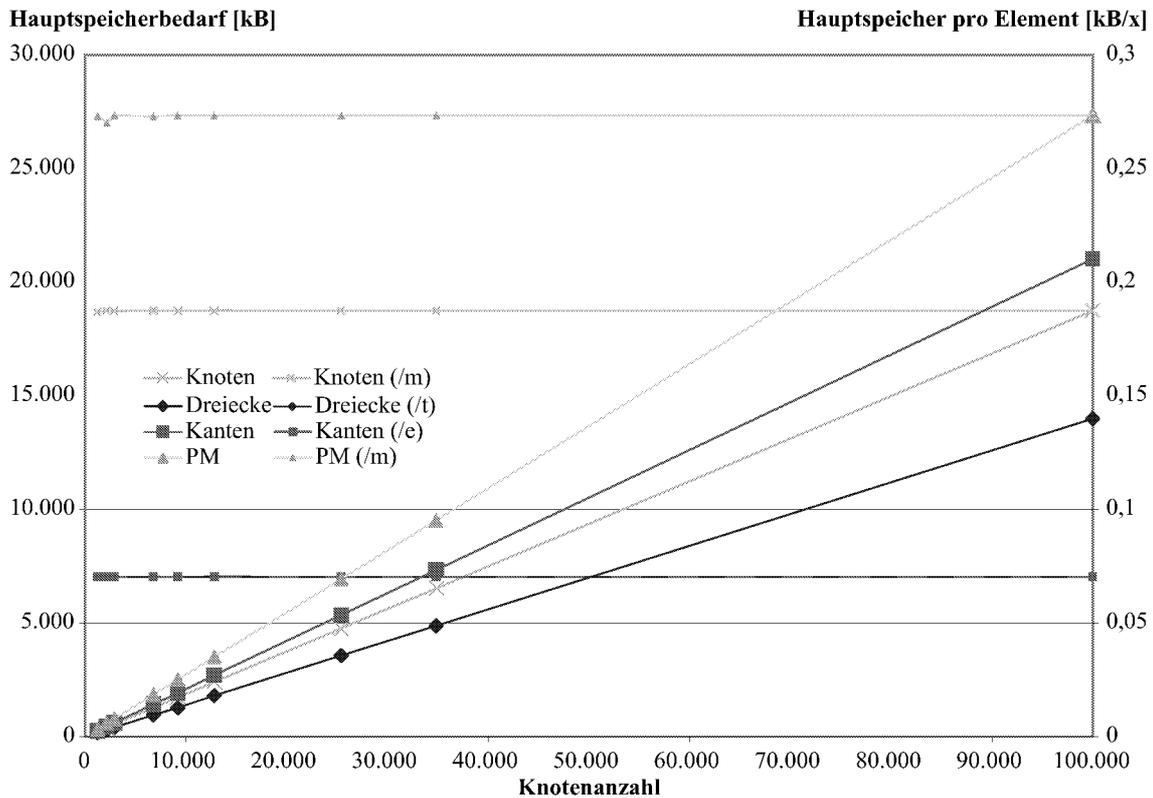


Abb. 4.9: Hauptspeicherbedarf der vier speicherintensivsten Datenstrukturen des Dreiecksnetzes

muss nicht exakt vorliegen, da die Referenzlisten des Knotenvektors und des PM nicht gleich lang sein müssen. Im Durchschnitt liegen aber an jedem Knoten sechs Kanten und sechs Dreiecke an. Der große Speicherbedarf für den Knotenvektor beruht insbesondere auf der mit ihm gespeicherten Quadrik. Dabei wird für die Quadrik keine 4×4 -Matrix vorgehalten, sondern wegen ihrer Symmetrie nur genau die 10 unterschiedlichen Werte. Den meisten Speicher verbraucht aber das Progressive Mesh. Bei diesem ist für das zweitkleinste Modell der Knochen eine leichte Abweichung beim Hauptspeicherbedarf pro Netzknoten zu verzeichnen, da durch die im Verhältnis zur Knotenzahl große Zahl an Zusammenhangskomponenten und dem Schwellwert von Null, der das Zusammenschmelzen von Teilnetzen verhindert, die Restknotenanzahl des Basisnetzes ohne Dreiecksflächen groß ist und dadurch die Anzahl an Kontraktionen um die Anzahl von Zusammenhangskomponenten kleiner ist als die Knotenzahl. Der Speicherbedarf für ein Dreieck und eine Kante ist identisch, was zu der gestrichelten Linie bei 70 Byte führt. Durch die Nutzung einer C++-Standard-Musterklassen-Bibliothek, hat auch deren Implementation Einfluss auf den Speicherverbrauch, der bei der verwendeten STL-Implementation STLport [84] etwas geringer ist als bei der Implementation, die dem Microsoft Visual C++-Compiler beigelegt ist.

Diagramm 4.10 zeigt im Gegensatz zum vorherigen, den gesamten Speicherbedarf des Programms zur Netzvereinfachung, der auch in Tabelle 4.1 angegeben ist. Auch hier ist wieder links der absolute Bedarf und rechts der relative Speicherbedarf pro Knoten aufgetragen. Die Daten sind dem Windows Task-Manager zur Speicherauslastung der Prozesse entnommen worden und umfassen somit auch alle neben den vier wichtigsten Datenstrukturen benötigten Variablen. Aus diesem Grund ist der Speicherbedarf pro Knoten am Anfang stark abfallend. Der hyperbolische Anteil im relativen Hauptspeicherbe-

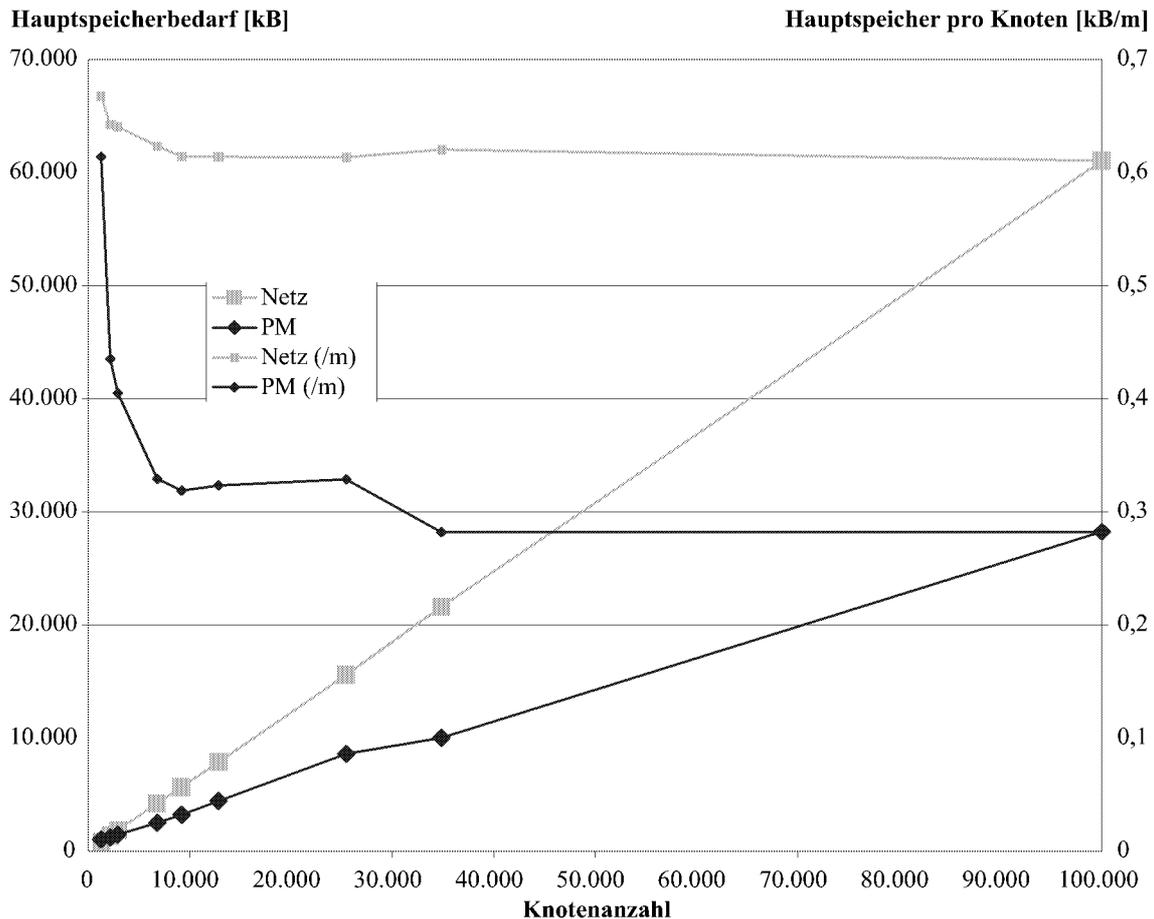


Abb. 4.10: Hauptspeicherbedarf des Programms nach Einlesen des Netzes und zusätzlicher Bedarf zur Speicherung des PM nach der Vereinfachung

darf rührt von der konstanten Menge an nicht mit der Komplexität des Netzes ansteigenden Strukturen des Dreiecksnetzes und des sonstigen Programms. Für große Knotenanzahlen bleibt dann der relative Wert nahezu konstant, da der zusätzliche Speicher relativ zur Knotenzahl vernachlässigbar klein wird. Die Speicherverwaltung des Systems und der STL-Implementation kann dabei zu unvorhersehbaren Schwankungen der Speicherbelegung führen, da oft nicht genau nur der angeforderte Speicherplatz aus Effizienzgründen belegt wird, sondern gegebenenfalls größere Blöcke. Das gleiche Bild stellt sich bei der Freigabe ein, wo nicht sofort der gesamte vom Benutzer freigegebene Speicher an das System zurückgegeben wird, um für eine erneute Anforderung entsprechend Reserven zu haben.

Die Kurve des Netzes in Diagramm 4.10 gibt den Speicherbedarf des Programms nach Laden des jeweiligen Modells an, wobei durch die Speicherung der Quadriken im Knotenvektor der Platz für die Quadriken schon zu diesem Zeitpunkt belegt ist (siehe Anhang A.2.2). Die PM-Kurve enthält die Datenmenge, die nach der Simplifizierung und dem kompletten Zurücksetzen des Netzes auf das Originalnetz zusätzlich hinzukommt. Wird das Netz dynamisch vereinfacht, verringert sich der Speicherbedarf, da sich die Dreieckslisten des Knotenvektors in ihrer Gesamtheit verringern und bei kompletter Vereinfachung alle leer sind. Während der Vereinfachung ist nicht mehr Speicherplatz nötig, als nach der Vereinfachung, da laut Diagramm 4.9 das Progressive Mesh mehr Platz beansprucht als die temporär zur Vereinfachung benötigte Kantenliste. Während der Vereinfachung wird kontinuierlich das

PM aufgebaut und im gleichen Zuge die Kantenliste abgebaut.

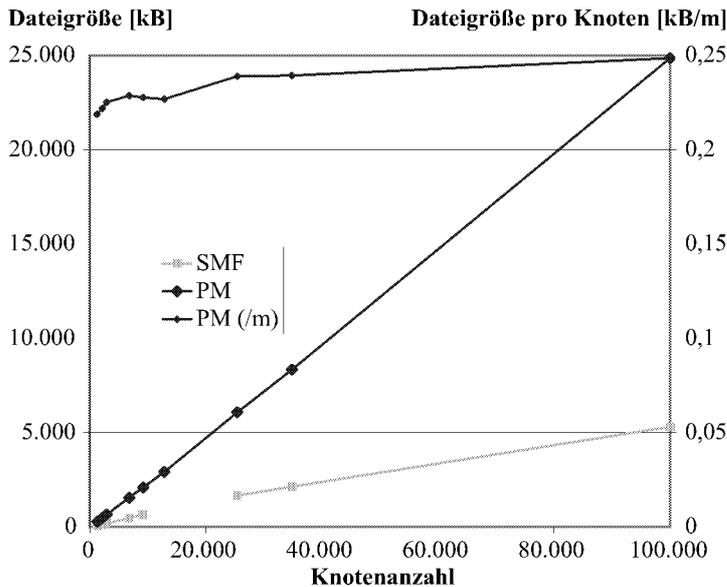


Abb. 4.11: Dateigröße des ausgewählten Modelle im SMF- und PM-Format

Als letzter Punkt der Speicherplatzanalyse soll nicht der Platz im Hauptspeicher sondern auf der Platte betrachtet werden. Das nebenstehende Diagramm 4.11 hat wieder zur Linken den absoluten Platzbedarf und zur Rechten den zur Knotenanzahl relativen Bedarf aufgetragen. Die Dateigröße des SMF-Formates ist nicht relativ aufgetragen worden, da, wie schon weiter oben erwähnt, die Daten durch ihre unterschiedliche Herkunft bei gleicher Knotenanzahl unmotiviert voneinander abweichen. Die absolute Größe ist hingegen eingetragen worden, da diese nicht so empfindlich gegen diese Störungen ist. Der Wert für die Dichtewolke fehlt dabei, da diese nicht aus einer SMF-Datei stammt, sondern durch den Marching-Cube-Algorithmus erzeugt wird. Die Dateigröße des Progressive Mesh ist im Verhältnis zur Knotenanzahl leicht ansteigend, da, wie auch schon oben erwähnt, die PM-Datei als Text- und nicht als Binär-Datei vorliegt und damit insbesondere große ganzzahlige Werte mehr Platz beanspruchen als kleine.

4.4 Qualität

Die Schnelligkeit des Verfahrens und auch der moderate Speicherbedarf sind in den letzten beiden Abschnitten herausgestellt worden. Ein gutes Verfahren zeichnet sich aber insbesondere neben seiner Schnelligkeit durch gute Resultate aus. Daher soll sich dieser Abschnitt mit der Qualität der vereinfachten Modelle befassen. Ein Qualitätsvergleich des ursprünglichen Verfahrens zu den anderen vorgestellten Verfahren, findet der Leser in Abschnitt 2.7.4.

Um die Qualität der vereinfachten Netze zu bewerten, wird ein etwas strengeres Fehlermaß als die in den Simplifizierungsverfahren vorgestellten heuristischen Maße, zu denen auch das in der Implementation verwandte lokale Quadrik-Fehlermaß gehört, gebraucht. Der globale geometrische Fehler soll hierbei den die visuelle Qualität ausmachenden komplexeren sichtbaren Fehler vertreten. Andere Ansätze zur Qualitätskontrolle beziehen weitere Modelleigenschaften mit ein, berechnen Fehler auf Basis des am Ende einer Visualisierungspipeline^G gerenderten^G Bildes oder bewerten die Art der Netzaufteilung.

Die globalen geometrischen Fehler wurden bereits in den Grundlagen, Abschnitt 2.1.2, beschrieben und in der Implementation mit den dort definierten vereinfachten Maßen E_{max} , E_{sum} , und E_{avg}

berechnet. Diese lauten zur Wiederholung:

$$E_{sum}(M_1, M_2) = \frac{1}{k_1 + k_2} \left(\sum_{v \in V_1} d_v(M_2) + \sum_{v \in V_2} d_v(M_1) \right),$$

$$E_{avg}(M_1, M_2) = \frac{1}{k_1 + k_2} \left(\sum_{v \in V_1} d_v^2(M_2) + \sum_{v \in V_2} d_v^2(M_1) \right),$$

$$E_{max}(M_1, M_2) = \max \left(\max_{v \in V_1} d_v(M_2), \max_{v \in V_2} d_v(M_1) \right).$$

Diese Maße dienen in dieser Arbeit lediglich der anschließenden Bewertung der Netzvereinfachung und fließen nicht in die Berechnung ein, da die Berechnung der globalen Maße sehr lange dauert. Als Fehler wird im Folgenden immer die geometrische Differenz der Vereinfachung zum Originalmodell bezeichnet.

Um visuell einen Eindruck von der Qualität der Vereinfachung zu bekommen, sind in Abbildung 4.12, 4.13 und 4.14 jeweils zwei Detailstufen mit ihrem Originalmodell abgebildet. Die Daten sind unterschiedlicher Herkunft. Das "Stanford Bunny" in 4.12 ist ein gleichmäßig fein abgetastetes Modell aus einem 3D-Scanner und das Flugzeug in 4.13 ein konstruiertes Modell, das viele kleine Details besitzt, einfache Modellbereiche aber angepasst durch ein gröberes Netz beschreibt. Die kleinen feinen Details sind es auch, die das Flugzeug überhaupt nur sinnvoll vereinfachen lassen. Andere konstruierte Modelle besitzen hingegen manchmal kaum noch Simplifizierungspotential. Die letzte Abbildung 4.14 ist eine Isofläche der eigentlichen Anwendung. Dass nicht nur anwendungsspezifische Modelle betrachtet werden, für die die Implementation vordringlich erstellt wurde, liegt daran, dass Unterschiede in den Detailstufen intuitiv viel eher in bekannten Objekten wahrgenommen werden, als in Gebilden außerhalb der vom Betrachter vertrauten Welt, wie den Isoflächen der Elektronendichtewolken - es fällt also ein dem Hasen fehlendes Ohr viel eher auf, als ein fehlender Zacken in Abbildung 4.14.

Die unterschiedliche Herkunft der Modelle soll zudem die Flexibilität des Verfahrens zeigen, da jedes Verfahren seine bevorzugte Objektklasse hat, für die es besonders gut arbeitet, und Objekte, die mit ihm schlechter vereinfacht werden können. Luebke empfiehlt in [71] das Quadrik-Fehlermaß zur Vereinfachung von so genannten organischen Objekten, wie es das "Stanford Bunny", aber auch die Isofläche einer Elektronendichtewolke, darstellt.

Die Modellausdehnung aller neuen Modelle ist wie bei den bisherigen Modellen 2 in ihrer maximalen Ausdehnungsrichtung.

Fein abgetastete Modelle lassen sich für gewöhnlich stärker mit nahezu keinem visuellen Unterschied vereinfachen, als grobe Modelle oder Modelle, deren Netzdichte schon an die Oberflächenbeschaffenheit und ihrer Krümmung angepasst ist, was oft bei konstruierten Objekten wie dem Flugzeug vorkommt. Das Vereinfachungspotential ist bei gleicher Fehlerschranke also für feine Netze größer. Damit für den letzten Fall der differenzierten Netzdichte die stellenweise feinen Details das Verfahren nicht ungünstig stark beeinflussen, wäre eine Vereinfachung sinnvoll, die die Flächengröße der Dreiecke mit in die Kostenberechnung aufnimmt. Ein Vorschlag dafür ist im Ausblick, Abschnitt 5.2.2, zu finden.

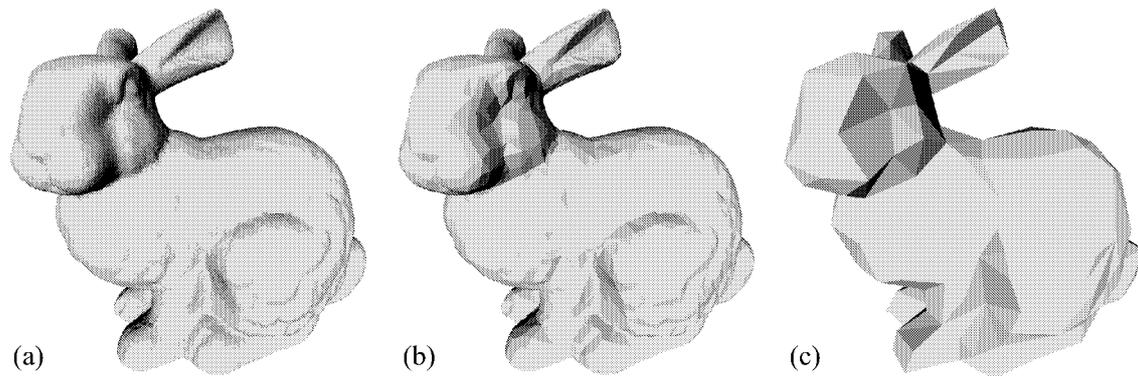


Abb. 4.12: (a) Originalmodell des “Stanford Bunnys” mit 34.834 Knoten, Vereinfachungen mit (b) 3.000 ($\approx 9\%$) und (c) 300 ($\approx 1\%$) Knoten

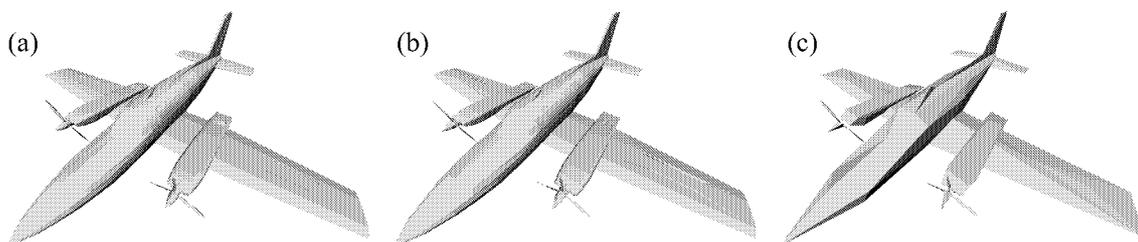


Abb. 4.13: (a) Originalmodell eines Flugzeugs mit 6.795 Knoten, Vereinfachung mit (b) 1.000 ($\approx 15\%$) und (c) 200 ($\approx 3\%$) Knoten

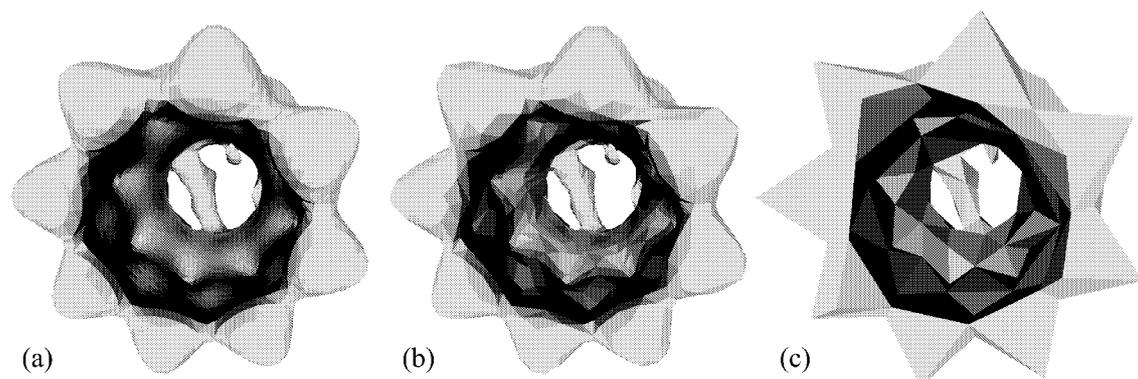


Abb. 4.14: (a) Originalmodell einer vom Marching-Cube-Algorithmus generierten Isofläche einer Elektronendichtewolke mit 12.835 Knoten, Vereinfachung mit (b) 2.000 ($\approx 16\%$) und (c) 200 ($\approx 2\%$) Knoten

Wegen der unterschiedlichen Vereinfachungsmöglichkeiten, ist auch der Vergleich von Fehlern von Simplifikationen auf Basis der Prozente vereinfachter Knoten nicht sehr aussagekräftig. Die Kuh aus Abbildung 3.5 hat für rund 14% an Knoten eine ähnliche visuelle Netzgüte wie der Drachen aus Abbildung 4.16 mit rund 4%. Trotzdem ist eine ungefähre Prozentangabe für die Abbildungen der Vereinfachungen dieses Abschnitts gemacht worden.

Aber auch die absolute Knotenzahl einer Detailstufe sagt nicht viel über den geometrischen Fehler der Stufe aus. Komplexe Modelle bedürfen zur Unterschreitung des gleichen Fehlers mehr Knoten

als einfache. Zum Beispiel ist daher der Drachen nur mit größeren Fehlern auf Detailstufen mit geringer Knotenzahl zu bringen, als der Hase, da das Netz des Hasen weniger herausragende Details besitzt als der Drache mit seinen Zähnen und dem Kamm entlang der Wirbelsäule. Ähnliches gilt auch im Vergleich der Wölbung zum Flugzeug, die wie der Drache und Hase Knoten in der gleichen Größenordnung besitzen. Das Flugzeug besitzt eine kompliziertere Struktur als die Wölbung, deren Einfachheit nur noch durch eine ebene Fläche zu überbieten wäre, deren gesamten globalen Fehler ohne eingeschaltete Randbestrafung immer Null sind. Aber auch die Geländeoberflächen wie die aus Abbildung 2.20 besitzen kleine globale Fehler, da ihre Ausdehnung in der Breite und Tiefe viel größer ist als in der Höhe, wodurch die maximale Höhe den geringen maximalen globalen Fehler bestimmt.

Diagramm 4.15 vergleicht die globalen Fehler der Wölbung und des Flugzeugs für die Knotenzahlen 1.000, 400 und 100. Die Fehler für das Flugzeug sind, wie schon gesagt, viel größer als die für die Wölbung. Dabei wächst der Faktor zwischen den entsprechenden Fehlern bei kleiner werdender Knotenzahl für E_{sum} von 1,7 auf 3,4, für E_{avg} von 4,6 auf 17,2 und für E_{max} von 5,1 auf 12,6. Es ist zu beachten, dass die Fehler wie in den meisten folgenden Fehlerdiagrammen logarithmisch aufgetragen sind, was leicht zur Unterschätzung von Differenzen führt, aber es ermöglicht, Zahlen solch unterschiedlicher Größenordnung gut erkennbar zusammen darzustellen.

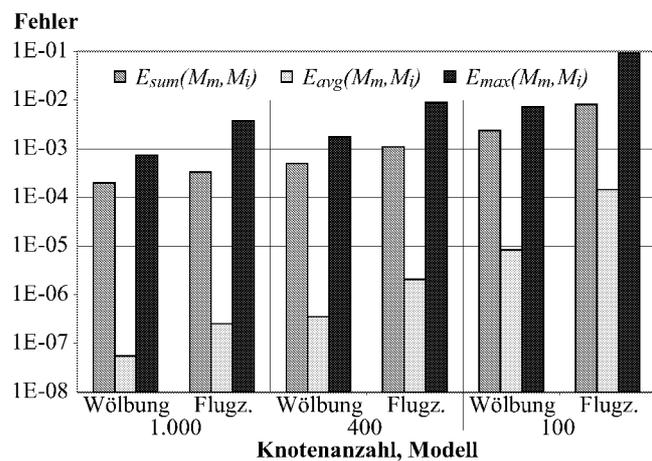


Abb. 4.15: Vergleich der globalen Fehler der einfachen Wölbung und des komplexeren Flugzeugs für unterschiedliche Detailstufen

Anhand der Abbildungen in dieser Arbeit ist auch zu erkennen, dass die Modelle bei fortschreitender Vereinfachung nicht wie bei einigen anderen Verfahren in sich zusammenfallen. Dadurch wird das Volumen des Modells gut erhalten, obwohl das Fehlermaß noch nicht durch die Dreiecksflächen gewichtet wurde, was gemäß Abschnitt 3.2.1 die Volumendifferenz besser minimieren würde.

Eine weitere positive Beobachtung an den Abbildungen ist die behutsame Reduzierung der Details. Für das Kuhmodell verschwinden zuerst die sehr feinen Details um die Augen, es folgt das Verschwinden kleiner Ausstülpungen wie den Euterzitzen. Bei weiterer Simplifizierung fallen die größeren Modellcharakteristika wie die Ohren und Hörner bis hin zu den Extremitäten zusammen, bevor das Modell zu einem Tetraeder wird. Ähnliches lässt sich beim Drachen beobachten. Zuerst verschwinden die durch den Scan-Vorgang unschönen feinen Rillen in der Modelloberfläche, bevor zum Beispiel die Zähne durch kleine Tetraeder ersetzt werden.

Bis zu der Detailstufe, die die grobe Modellstruktur erhält, würde durch die Anwendung von Texturen des Originalmodells auf die sich weiter aufspannenden Dreiecksflächen ein sich vom Original nur wenig unterscheidendes Bild dem Betrachter zeigen.

Die schon häufiger herangezogene Abbildung 4.16 des Drachens soll die Erhaltung der wichtigsten Details für die einzelnen Simplifizierungsstufen noch einmal zeigen und den Einfluss der Entfernung bei der Visualisierung, was im Druck der Abbildungsgröße entspricht, für das Empfinden der visuellen Güte verdeutlichen. Die Detailstufe ist für alle Teilabbildungen in (a) und (d) identisch. Ebenso besitzen (b) und (e) sowie (c) und (f) die gleiche Zahl an Netzknoten.

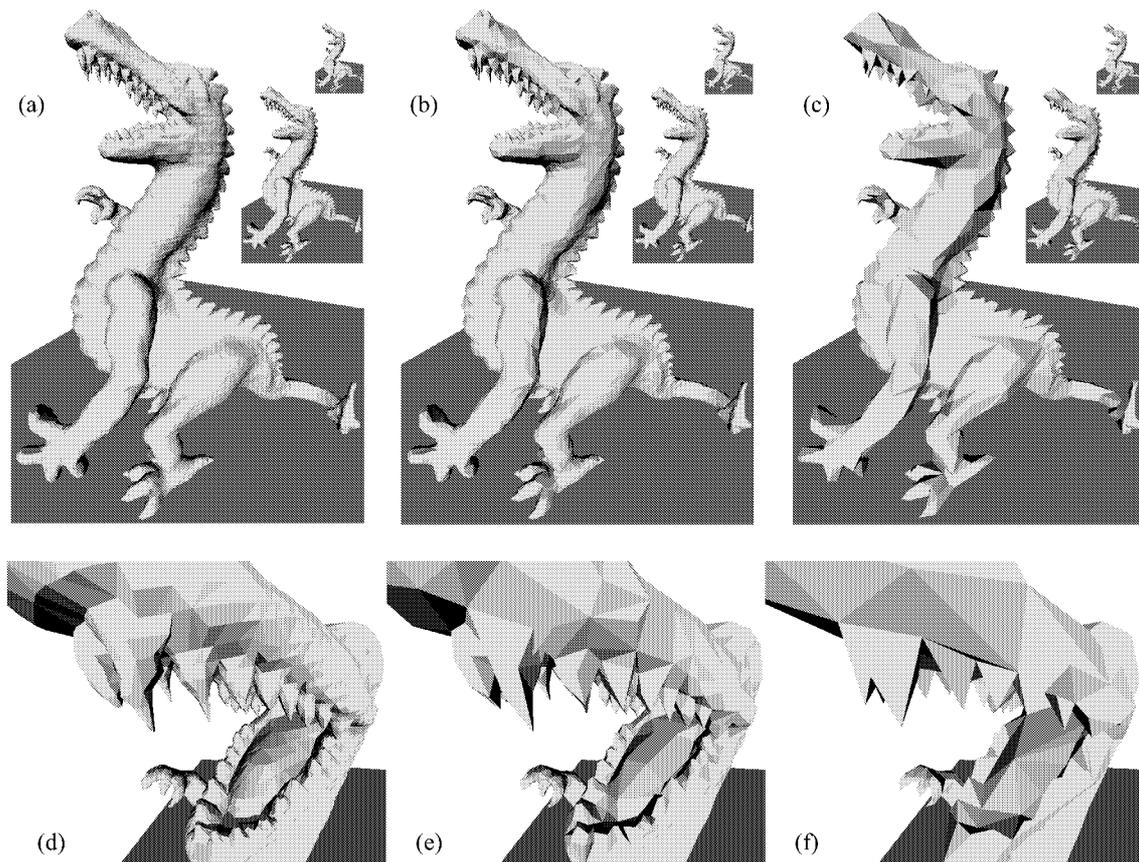


Abb. 4.16: (a), (d) Originalmodell eines Drachens mit 25.418 Knoten, Vereinfachung mit (b), (e) 5.000 ($\approx 20\%$) und (c), (f) 1.000 ($\approx 4\%$) Knoten; (a), (b) und (c) Ansicht des Gesamtmodells aus unterschiedlichen Entfernungen; (d), (e) und (f) Ausschnittsvergrößerung des Mauls

In der Nahansicht des Mauls sind alle drei Detailstufen gut voneinander zu unterscheiden, für die größte Komplettansicht fällt dies zwischen (a) und (b) schon schwerer und ist für die nächst kleinere Abbildung nicht mehr möglich. In dieser Entfernung ist nur noch (c) von (a) und (b) zu unterscheiden, was in der kleinsten Abbildung auch nicht mehr möglich ist, in der somit das Modell mit ungefähr 4% der Knotenzahl ein visuell gleiches Ergebnis liefert wie das Originalmodell.

Was für den Druck gilt, gilt auch für den Bildschirm. Auch am Bildschirm sind weiter entfernte Objekte für den Betrachter unempfindlicher gegenüber Vereinfachungen, da mehr Dreiecke auf einen Bildschirmpixel abgebildet werden, und damit durch Veränderungen der Bildfehler durch die Entfernung geringer wird.

Neben den in der Anwendung implementierten Randbedingungen der Bild- und Kraftwiederholrate zur automatischen adaptiven Anpassung der Detailstufe zeigt diese Abbildung auch eine einfache betrachterabhängige Anpassung, aufgrund derer das Modell beim Entfernen vom Betrachter immer weiter simplifiziert wird.

4.4.1 Globale Fehler der Netzvereinfachung

Die soeben gemachten Aussagen zur visuellen Güte sollen nun exemplarisch durch Messungen der globalen Fehler bestätigt werden.

Da ein Vergleich der Fehlerverläufe der einzelnen Modelle wegen der unterschiedlichen Netzdichten und -komplexitäten müßig ist, wurde sich bei der Betrachtung auf das Kuhmodell beschränkt. Die Verlaufsscharakteristika der Diagramme anderer Modelle sähen aber ähnlich der der Kuh in Diagramm 4.17 aus.

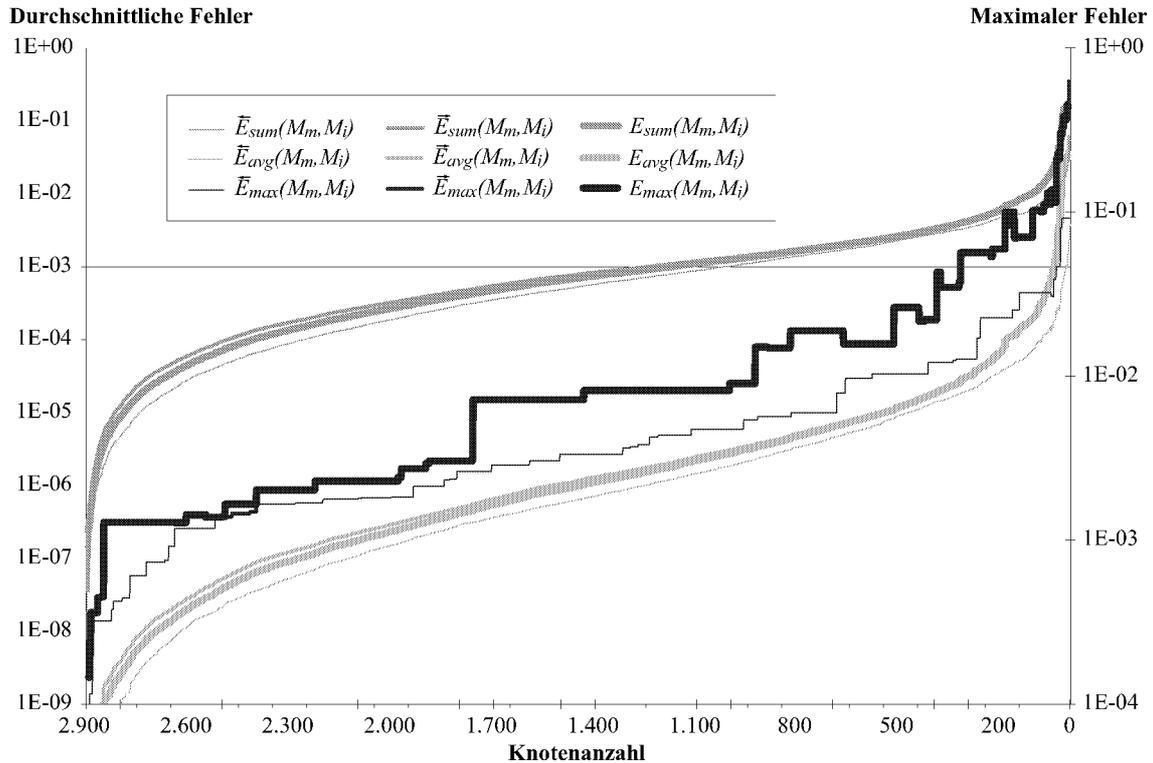


Abb. 4.17: Globale Fehler der Detailstufen des Kuhmodells

Diagramm 4.17 zeigt als dicke Linien den Verlauf der globalen Fehler $E_{sum}(M_m, M_i)$, $E_{avg}(M_m, M_i)$ und $E_{max}(M_m, M_i)$, $i \in \mathbb{N}_m$ der Detailstufen M_i mit der auf der x-Achse aufgetragenen Anzahl an $|V_i| = i$ Knoten V_i zum Originalmodell M_m der Kuh mit $m = 2.903$ Knoten. Zudem sind die Teilsummen bzw. Teilmaxima der gerichteten bzw. einseitigen globalen Fehler

$$\begin{aligned} \bar{E}_{sum}(M_m, M_i) &= \frac{1}{i} \sum_{v \in V_i} d_v(M_m), & \vec{E}_{sum}(M_m, M_i) &= \frac{1}{m} \sum_{v \in V_m} d_v(M_i), \\ \bar{E}_{avg}(M_m, M_i) &= \frac{1}{i} \sum_{v \in V_i} d_v^2(M_m), & \vec{E}_{avg}(M_m, M_i) &= \frac{1}{m} \sum_{v \in V_m} d_v^2(M_i), \\ \bar{E}_{max}(M_m, M_i) &= \max_{v \in V_i} d_v(M_m), & \vec{E}_{max}(M_m, M_i) &= \max_{v \in V_m} d_v(M_i) \end{aligned}$$

für $i \in \mathbb{N}_m$ eingezeichnet. \bar{E}_* misst also die Abstände von den Knoten der Simplifizierung zur Oberfläche des Originalmodells und \vec{E}_* die Abstände der Originalknoten zu der vereinfachten Oberfläche.

Die durchschnittlichen Fehler \bar{E}_{sum}^* und \bar{E}_{avg}^* sind dabei auf der linken und der maximale Fehler \bar{E}_{max}^* auf der rechten logarithmischen Achse aufgetragen. Die unterschiedlichen Achsen ermöglichen dabei die bessere Erfassung des Spektrums, das der Fehler während der Simplifizierung überstreift. Die rechte Achse erfasst einen um fünf Größenordnungen kleineren Fehlerbereich, so dass die Linien des maximalen Fehlers bei einer gemeinsamen Achse immer oberhalb der anderen Linien verlaufen würden. Das Verhältnis vom maximalen zum näheren Summen-Fehler E_{max}/E_{sum} ist tendenziell abfallend und liegt schon ab einer Detailstufe mit 2.500 Knoten unterhalb von 20.

Die Linien sind nicht immer zu sehen, da sie häufig aufeinander liegen. Dies gilt insbesondere für die Hausdorff-Abstände, da das Maximum beider Teilmaxima immer genau auf eines der Teilmaxima fällt. Dabei ist das Gesamtmaximum außer um 2.450 Knoten immer $\bar{E}_{max}^*(M_m, M_i)$. Auch für die durchschnittlichen Fehler ist der Abstand von den Knoten des vereinfachten Netzes zur Originaloberfläche kleiner als der Fehler für die andere Richtung. Dies liegt daran, dass das Quadrik-Fehlermaß lokal genau diesen Abstand von den vereinfachten Knoten zur Oberfläche des Originalmodells minimiert, wobei nicht genau die Oberfläche sondern die Ebenen durch die Dreiecksflächen in die Bewertung einfließen, aber diese Annäherung lokal eine gute Abschätzung darstellt.

Alle Fehler steigen natürlich tendenziell an, müssen dies aber nicht monoton tun, wie auch hier am maximalen Fehler zu sehen. Aber auch andere Fehler müssen nicht monoton steigen, was zum Beispiel für das Blattmodell in Diagramm 4.25 vorkommt. Nicht im Diagramm sichtbar trifft es sogar oft zu, dass an den Stellen, an denen der maximale Fehler abfällt, auch die durchschnittlichen Fehler minimal nachgeben, was aber durch die Mittlung kaum sichtbar ist.

Die Fehler in Abhängigkeit zur prozentual verbleibenden oder zur absoluten Anzahl an Knoten zu bringen ist schwer. Anhand Diagramm 4.17 ist zu erkennen, dass für das Kuhmodell im mittleren Bereich zwischen 2.300 und 400 verbleibenden Knoten der logarithmische Fehler annähernd proportional zur Anzahl eliminiertes Knoten wächst.

Die Kuh hat eine im Verhältnis zur Netzkomplexität geringe Netzdichte, was dazu führt, dass sich schnell signifikante Fehler einstellen und die Fehlerkurven schnell in Fehlerbereiche aufsteigen, die für diese Kombination von Knotenzahl und Komplexität typisch sind. Hingegen bewirkt am Anfang der Vereinfachung diese bei überabgetasteten Modellen kaum einen Fehler. Entgegengesetzt verhält es sich am Ende fast aller Vereinfachungen, wo die Knotenzahl unter eine kritische Grenze fällt, die es ermöglicht, die wesentlichen Modellcharakteristika wiederzugeben. Für die Kuh liegt diese Grenze ungefähr bei 200 Knoten. In Abbildung 3.5 ist dies an den verschwindenden Hörnern der als nächstes dargestellten Detailstufe zu erkennen und im Diagramm 4.17 verlassen die durchschnittlichen Fehlerkurven ihren bis dahin annähernd linearen Verlauf.

Wie im Vergleich der Verfahren in Abschnitt 2.7.4 beschrieben, ist das Quadrik-Fehlermaß durch die Berücksichtigung des lokalen quadratischen Abstandes dahingehend ausgelegt, dass bei den globalen Fehlern der durchschnittliche Quadratabstand E_{avg} möglichst minimal ist, so dass dieser auch im Verhältnis zu anderen Verfahren besonders niedrig ist.

Die Ungleichung

$$E_{max}(M_m, M_i) \geq E_{sum}(M_m, M_i)$$

ist immer erfüllt, da der maximale Fehler größer ist als der einfache durchschnittliche Fehler. Für die betrachteten Modelle mit ihren maximalen Modellausdehnungen von 2 gilt aber für gewöhnlich auch

$$E_{sum}(M_m, M_i) > E_{avg}(M_m, M_i),$$

da der einzelne Netzabstand im Allgemeinen kleiner ist als 1 und damit $d_v^2(M) < d_v(M)$ ist. Für das Kuhmodell bleiben alle Fehler bis zur letzten verbleibenden Fläche unterhalb von 1. Dies muss aber nicht zwingend so sein und wird zum Beispiel vom Blattmodell laut Diagramm 4.25 schon nicht erfüllt. Liegen die Modellausdehnungen über 2, so kann E_{avg} sogar größer als E_{max} werden, da sich E_{avg} ohne die Wurzel an keine Metrik direkt anlehnt.

Im Diagramm 4.17 lässt sich ablesen, dass der maximale Fehler 0,01 und damit weniger als 1% der Modellausdehnung erst bei rund 900 Restknoten übersteigt. Der durchschnittliche Summen-Fehler übersteigt 0,01 sogar erst bei einer Detailstufe mit 100 Knoten. Zu diesem Zeitpunkt liegen die Knoten im Schnitt also erst um weniger als 1% der gesamten Modellgröße vom Originalnetz entfernt.

4.4.2 Auswirkungen der Wahl des Verschmelzungspunktes auf die globalen Fehler

Das implementierte Verfahren wählt wenn möglich als Verschmelzungspunkt den für das Quadrik-Fehlermaß optimalen Punkt mit dem geringsten Fehler. Bei der Geschwindigkeitsanalyse wurde bereits festgestellt, dass diese Wahl um rund 7,5% schneller ist als die einfache Auswahl des Punktes mit dem kleinsten Fehler unter den Kantenendpunkten und deren Mittelpunkt. Hier soll gezeigt werden, dass das eigentliche Ziel der optimalen Knotenwahl, durch möglichst kleine globale Fehler die optische Qualität der Detailstufen zu erhöhen, erfüllt wird.

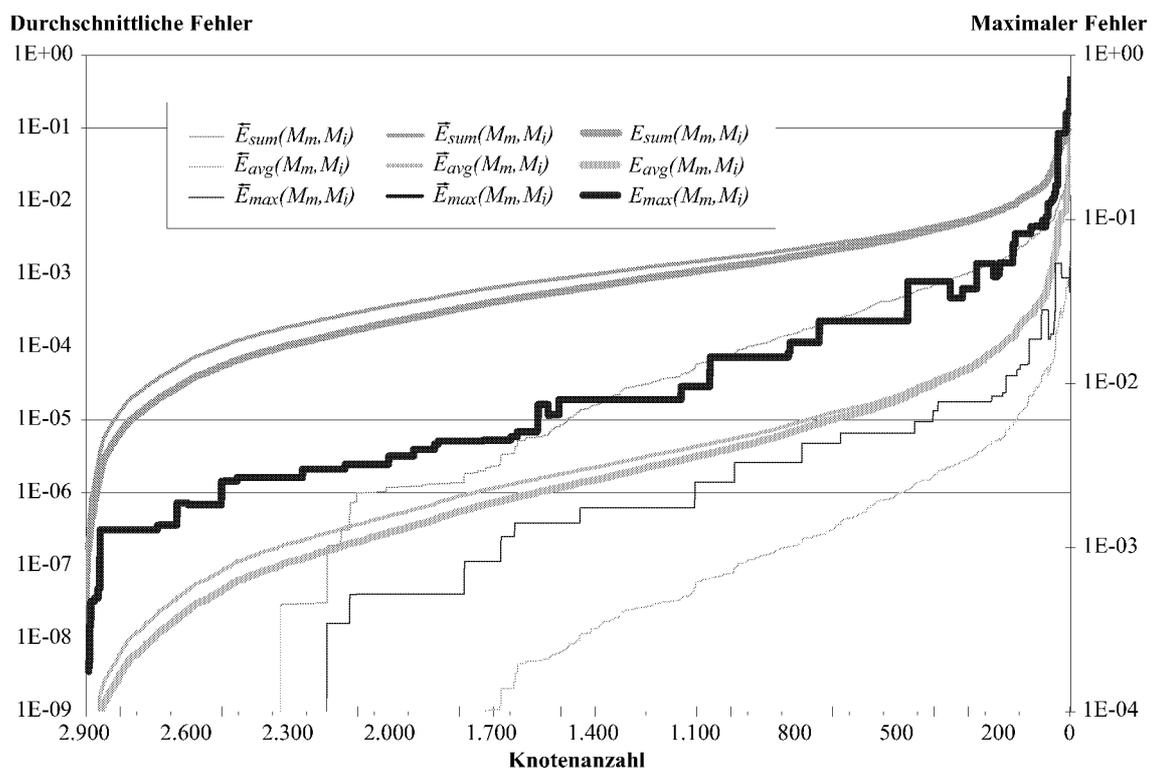


Abb. 4.18: Globale Fehler der Detailstufen des Kuhmodells für Verschmelzungskandidaten an den Kantenendpunkten und deren Mittelpunkt

Diagramm 4.18 zeigt das dem Diagramm 4.17 entsprechende Bild für das Knotentripel der Kantenendpunkte und deren Mittelpunkt als mögliches Kontraktionsziel statt der optimalen Knotenwahl.

Die Fehlerkurven der globalen symmetrischen und der gewichteten Fehler von den Originalknoten zur vereinfachten Modelloberfläche sehen dabei denen im Diagramm 4.17 ähnlich. Auffällig anders verhält sich aber der gerichtete Fehler von den Knoten des vereinfachten Netzes zur Originaloberfläche. Dieser ist sehr gering und verursacht dadurch auch die größere Abweichung von \bar{E}_{sum} zu E_{sum} und \bar{E}_{avg} zu E_{avg} .

Dieser die symmetrischen Fehler herunterziehende Effekt lässt bei fallender Knotenzahl im vereinfachten Netz nach, da die Mittlung der beidseitigen Fehler gemeinsam für beide Teilsommen geschieht. Dadurch verliert die Teilsumme mit einer geringeren Zahl an Knotenabständen an Bedeutung. Für die Teilsumme, die den Abstand von den Originalknoten zum vereinfachten Netz misst, bleibt die Zahl der Summenglieder gleich, für die andere Teilsumme fällt sie kontinuierlich mit der Zahl an Knoten im vereinfachten Netz. Bei getrennter Mittlung wäre die Bedeutung für beide Teilsommen immer identisch, so dass die Knoten von Netzen mit weniger Knoten stärkeren Einfluss haben. Dies könnte man aber für diesen Zweck der Bewertung eines Originalnetzes zu seiner Vereinfachung akzeptieren, da im vereinfachten Netz ein Knoten auch mehrere Knoten, aus denen er hervorgegangen ist, vertritt. In dieser Arbeit wurde aber immer eine gemeinsame Mittlung durchgeführt, wodurch sozusagen der Einfluss der Kurve von \bar{E}_{sum} auf E_{sum} bzw. \bar{E}_{avg} auf E_{avg} bei fallender Knotenzahl abnimmt.

Die geringen Fehler von \bar{E}_* sind direkt auf die Knotenwahl für die Kontraktion zurückzuführen. Bei der Kuh wurden für die Kontraktion keine imaginären Kanten hinzugenommen. Dadurch, dass die Knoten des vereinfachten Netzes auf den Knoten der vorangegangenen Detailstufe selbst liegen oder nur auf die Kantenmitte verschoben werden, liegen im Bild des Knotenbaumes der betrachterabhängigen Datenstruktur aus Abbildung 2.21 die Knoten der zweituntersten Knotenebene oberhalb der Originalknoten auf den Kanten des Originalnetzes. In der nächsten Stufe der Vereinfachung, entsprechend der zweituntersten Ebene im Knotenbaum, können die Knoten sich zwar von den Kanten lösen und auf die Fläche des Originalnetzes wandern, erzeugen aber in \bar{E}_* noch keinen Fehler, da sie noch auf der Originaloberfläche liegen. Dass sie auf dieser bleiben, liegt an der Tatsache, dass ohne imaginäre Kanten die Verschmelzungspunkte auf einer gemeinsamen Netzkante liegen. Diese Netzkante muss auf der Originaloberfläche liegen, da in der vorangegangenen Vereinfachungsstufe die Kantenknoten nur entlang der Originalkanten verschoben werden konnten und somit die dadurch gewanderten Netzkanten auf dem Dreieck verbleiben. Erst in der dritten Stufe, also nachdem ein Knoten drei Mal an einer Kontraktion beteiligt war, kann das Kontraktionsziel einen globalen Fehler verursachen. Dieser Fehler wird aber auch nur dann verursacht, wenn in beiden vorangegangenen Stufen der Kantenmittelpunkt als Verschmelzungspunkt genommen wurde. Würde der Knotenbaum also balanciert sein und jede Ebene nacheinander abgearbeitet werden, so würde sich ein Knoten frühestens bei den

$$m - \frac{m}{c} - \frac{m}{4} = \frac{m}{4}$$

letzten Kontraktionen von der Originalfläche bestehend aus m Knoten lösen. Im Falle der Kuh verursacht aber bereits die 37. Kontraktion die ersten echt positiven globalen gerichteten Fehler \bar{E}_* , da wie vorher erwähnt nicht gleichmäßig im Netz vereinfacht wird, sondern in diesem Fall zum Beispiel erst die feinsten Details um die Augen reduziert werden und damit die dritte Stufe im Knotenbaum schneller erreicht wird.

Die gerichteten Fehler \bar{E}_* wären immer Null, wenn nur die Kantenendpunkte als Kontraktionsziel möglich wären, da dadurch das Verfahren positionstreu würde und in jeder Vereinfachungsstufe nur eine Teilmenge der Originalknoten vorkämen.

Die vereinfachten Maße sind eingeführt worden, um die exakten Netzabstände schneller zu berechnen. Aus der Messung über die gesamte Oberfläche ist eine diskrete Messung an einzelnen Punkten ge-

worden. Die diskreten Punkte an den Netzknoten anzunehmen, geschah hauptsächlich aus praktischen Gründen. Für den Vergleich der globalen Fehler für die unterschiedliche Wahl der Verschmelzungspunkte werden daher nur die gerichteten Fehler \vec{E}_* herangezogen, da die Fehler \vec{E}_* nur aufgrund der speziellen Wahl der Mess- und Verschmelzungspunkte eher zufällig so klein ausfallen. Für \vec{E}_* hingegen liegt bei der Wahl der Messpunkte für kein Verfahren ein besonders großer Vorteil vor. Auch hierbei ist sogar eher die optimale Wahl im Nachteil, weil für die Kantenendpunkte als Verschmelzungspunkt nur ein Knoten pro Detailstufe einen zusätzlichen globalen Fehler verursacht, wohingegen der optimale Verschmelzungspunkt zu Beginn meist zwei Originalknoten von der Netzoberfläche der Vereinfachung löst.

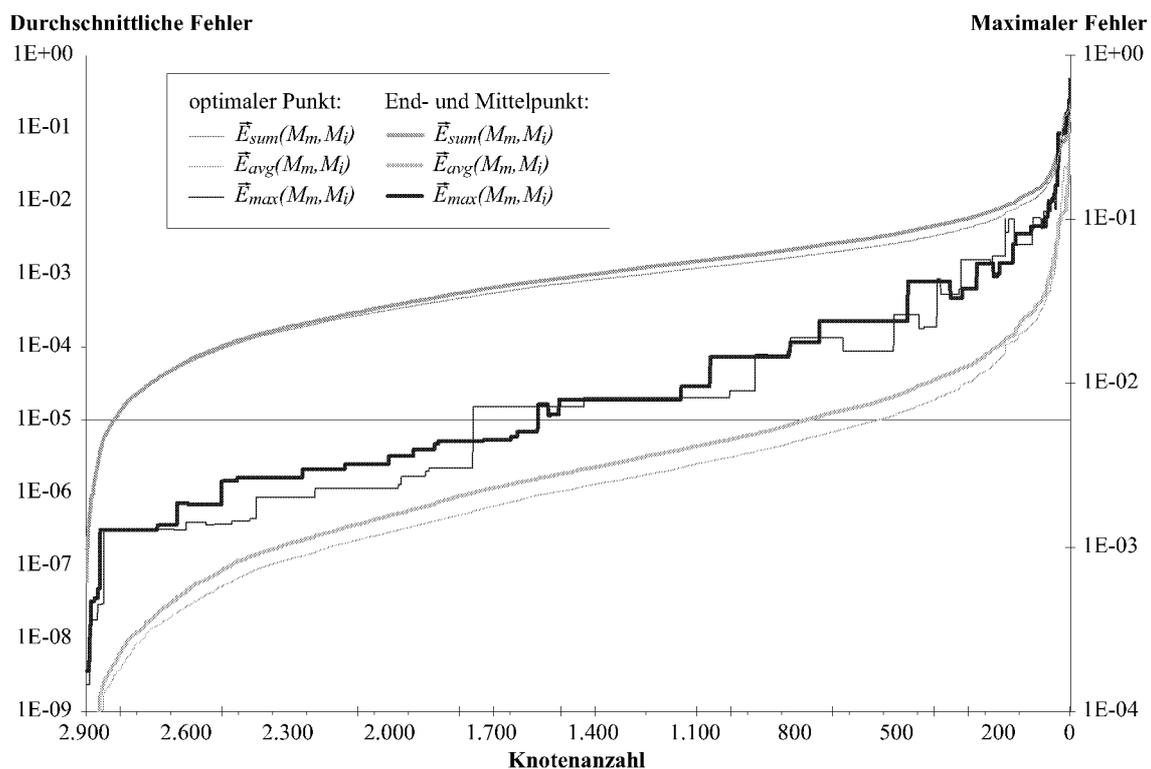


Abb. 4.19: Vergleich der globalen gerichteten Fehler \vec{E}_* von den Knoten des Originalnetzes zur Oberfläche der Detailstufe zwischen der optimalen Wahl des Knotenverschmelzungskandidaten bezüglich des Quadrik-Fehlermaßes und der eingeschränkten Wahlmöglichkeit auf das Knotentripel der Kantenendpunkte und deren Mittelpunkt

Ein Vergleich der globalen gerichteten Fehler \vec{E}_* ist nun im Diagramm 4.19 zu sehen. Da durch die logarithmische Fehlerachse der Unterschied der Fehlerkurven voneinander schwer abzuschätzen ist, wurde in Diagramm 4.20 aufgetragen, um wieviel Prozent der Fehler der Knotenwahl unter dem Knotentripel größer ist als der der optimalen Wahl. Eine positive Prozentangabe bedeutet also ein Vorteil der Strategie des optimalen Verschmelzungspunktes.

Wie nicht anders zu erwarten, da sonst die optimale Wahl wenig sinnvoll wäre, liegen die globalen Fehler dabei in großen Teilen für die implementierte Strategie viel tiefer als für die Knotenwahl innerhalb der Kantenendpunkte und deren Mittelpunkt, die in der Implementation nur für nahezu singuläre Quadriken durchgeführt wurde.

Die Fehlerdifferenz verhält sich zu Beginn der Vereinfachung sehr wechselhaft und stabilisiert sich erst mit der Zeit, da jeder Operationsfehler im Verhältnis zur weiteren Vereinfachung beim Gesamt-

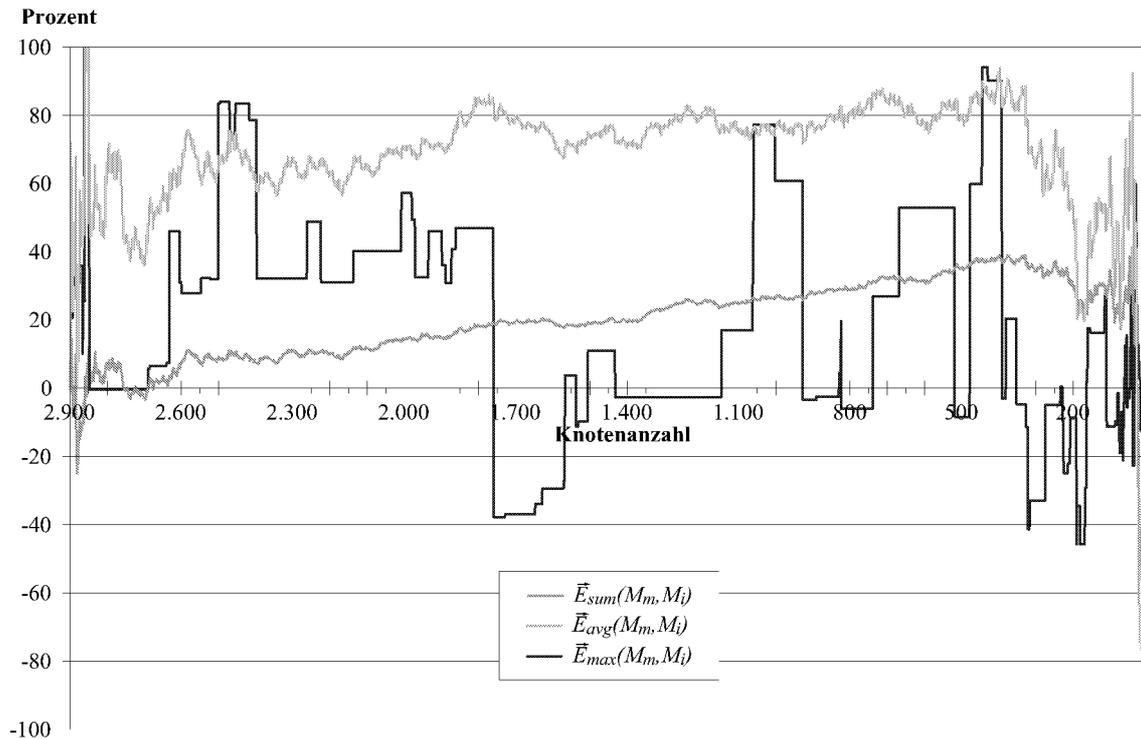


Abb. 4.20: Prozentuale Fehlerdifferenz für die optimale Knotenwahl

fehler und damit bei der Differenz höher ins Gewicht fällt. Ein ähnlich chaotisches Verhalten ist zum Ende der Vereinfachung zu beobachten, wo die kritische Knotenzahl der Vereinfachung überschritten wird und die Anzahl nicht mehr ausreicht, wesentliche Modellcharakteristika wiederzugeben. Für ein Modell der Komplexität der Kuh liegt diese Zahl um die 200 Knoten.

Die größte Differenz ist für \vec{E}_{avg} zu beobachten, was daran liegt, dass genau dieser Fehler lokal durch die optimale Knotenwahl minimiert wird. Die Tendenz ist für den Summen-Fehler ähnlich der des durchschnittlichen Quadratfehlers, nur nicht so ausgeprägt. Nicht ganz so positiv sieht dies für den maximalen Fehler aus, da dieser durch das Quadrik-Fehlermaß auch lokal nicht berücksichtigt wird und somit die Minimierung dieses Maßes nicht zwangsläufig Einfluss auf den maximalen Fehler hat. Im Ganzen ist aber auch der maximale Fehler durch die optimale Wahl kleiner.

Die prozentuale Differenz von \vec{E}_{sum} und \vec{E}_{avg} ist im mittleren Vereinfachungsspektrum tendenziell steigend, da durch die schlechtere Wahl des Verschmelzungspunktes durch die Einschränkung auf das Knotentripel in vorangegangenen Simplifizierungsschritten diese auch direkt Einfluss auf die folgenden Kontraktionsmöglichkeiten haben. Bei der optimalen Wahl ist der Verschmelzungspunkt absolut unabhängig von dem Ort der Kantenendpunkte, und wird lediglich durch die Dreiecksebenen, die diese Knoten vertreten, über die Minimierung des Quadratabstandes zu ihnen festgelegt.

Auch in der Arbeit von Garland und Heckbert [31], in der das grundlegende Verfahren der implementierten Simplifizierung beschrieben wird, wird für das Kuhmodell ein prozentualer Fehlervergleich der beiden Alternativen für die Verschmelzungsposition angegeben. Diese Werte beziehen sich aber auf den symmetrischen Fehler E_{avg} , der trotz der weiter oben beschriebenen Probleme, positive Fehlerdifferenzen erzielt; sie sind nahezu identisch mit den Werten, die die Vereinfachung der eigenen Implementation liefert.

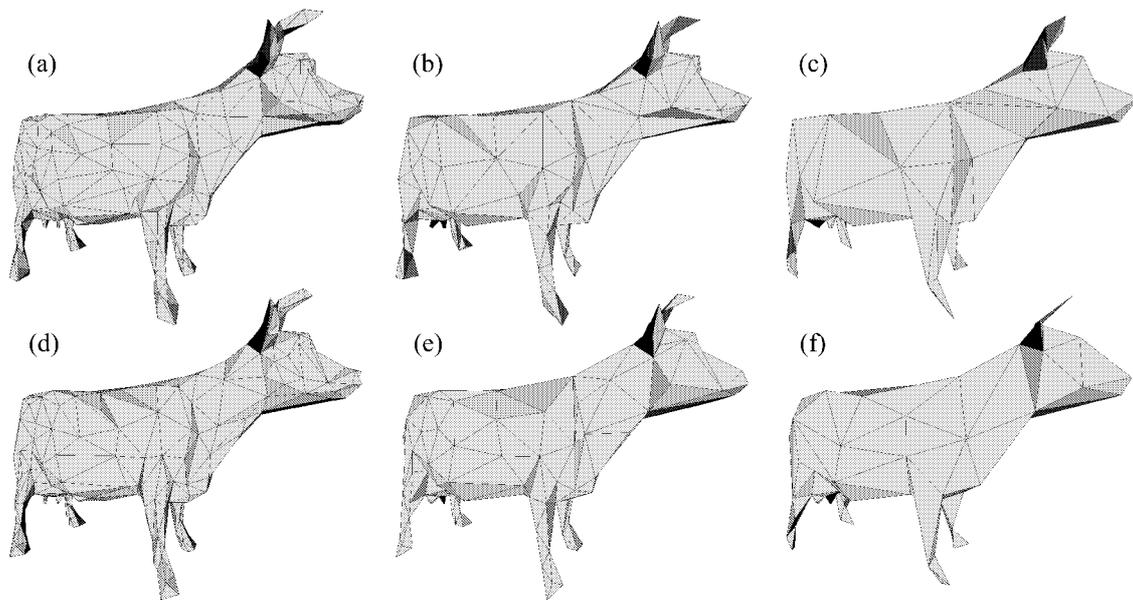


Abb. 4.21: Detailstufen des Kuhmodells mit optimaler Knotenwahl in (a), (b) und (c), sowie mit der eingeschränkten Knotenwahl aus den Kantenendpunkten und deren Mittelpunkt; (a) und (d) besitzen 400, (b) und (e) 200 und (c) und (f) 100 Knoten

Dass die Fehlerdifferenz nicht nur theoretisch messbar, sondern auch optisch sichtbar ist, zeigt Abbildung 4.21. In der ersten Spalte besitzt die Detailstufe 400, in der zweiten 200 und in der letzten 100 Knoten. Den Bildern der oberen Zeile liegt eine optimale Knotenwahl und der unteren eine Knotenwahl innerhalb der Kantenendpunkte und der Mittelpunkte zugrunde.

Die Knotenzahl der Detailstufen wurde so gewählt, dass für Komplettansichten die visuellen Fehler gut erkennbar sind, obwohl die letzte Spalte mit 100 Knoten schon in den Bereich fällt, in dem eigentlich zu wenig Knoten vorhanden sind, um die wesentlichen Modellcharakteristika noch wiedergeben zu können - dies ist unter anderem daran zu erkennen, dass die Beine ihr Volumen verlieren und die Hörner anfangen in sich zusammenzufallen.

Abbildung 4.21 (a) und (d) zeigen, dass (a) dem Originalnetz aus Abbildung 3.5 besser angepasst ist und zum Beispiel die Hörnerspitzen besser erhält. Die beiden folgenden Stufen bestätigen dies und zeigen zum Ende hin, dass zum Beispiel die Kopf- und Schulterform in (c) dem Original wesentlich mehr ähneln als in (f).

In Abschnitt 3.1.4 sind weitere Vorschläge für die Wahl des Verschmelzungskandidaten singulärer Quadriken gemacht worden, die statt dem Knotentripel auch eine Alternative zur optimalen Wahl darstellen. Aber auch diese werden keine besseren durchschnittlichen Fehler erzeugen. Eine einzige sinnvolle Alternative der Knotenwahl wäre eine Beschränkung auf die Kantenendpunkte, was zwar noch ein etwas schlechteres visuelles Ergebnis als das untersuchte Knotentripel ergeben wird, aber die Vorteile einer positionstreuen Vereinfachung hätte, die in Abschnitt 2.7.2 aufgeführt sind.

4.4.3 Einfluss des Schwellwertes auf die globalen Fehler

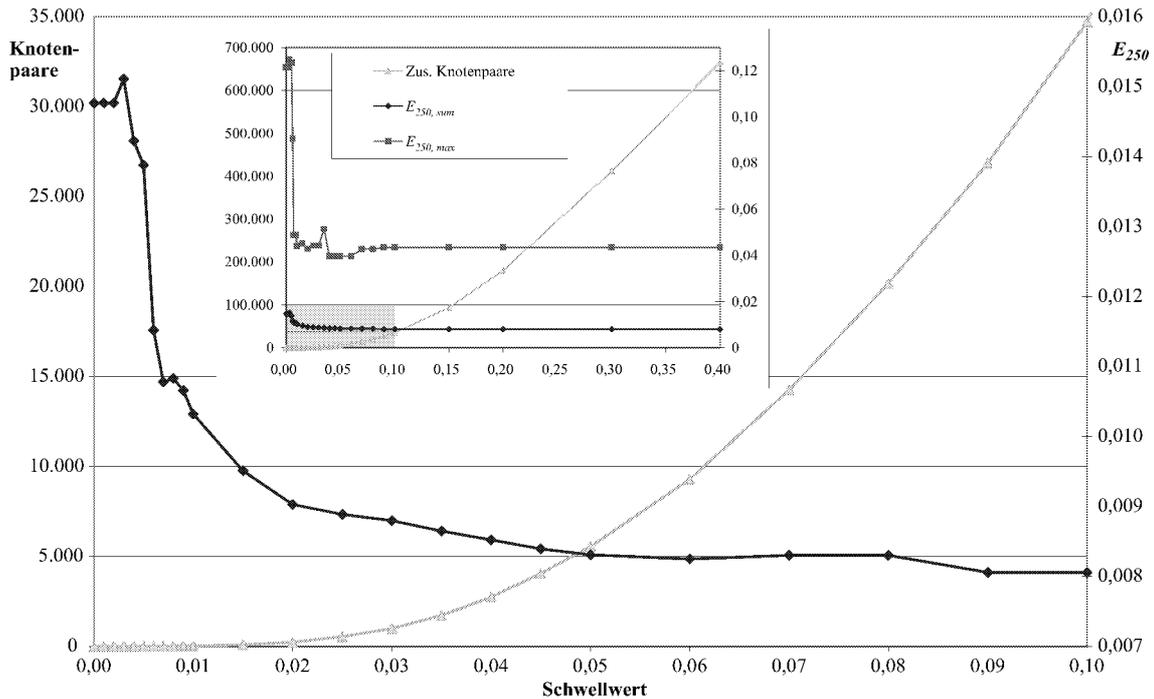


Abb. 4.22: Verlauf der globalen Fehler E_{sum} und E_{max} und der Anzahl von zusätzlichen Knotenverschmelzungspaaren in Abhängigkeit vom Schwellwert für das Modell des Fußknochens mit einer Ausdehnung von 2

In diesem Abschnitt soll der Einfluss des Schwellwertes zur Hinzunahme von Knotenpaaren zu den Verschmelzungskandidaten auf die globalen Fehler untersucht werden.

Die für diesen Zweck untersuchten Fußknochen haben wie alle Modelle zuvor eine maximale Ausdehnung von 2. Die Anzahl der zusätzlichen Knotenpaare steigt quadratisch mit Erhöhung des Schwellwertes und erreicht bei einem Schwellwert von 0,4 schon 665.941 zusätzliche Kandidaten zu den ursprünglich 6.306 realen Kanten.

Das Diagramm 4.22 zeigt diesen rasanten Anstieg. Der hinterlegte Bereich der kleinen Grafik ist in der großen Grafik besser sichtbar aufgetragen. Die zusätzlichen Knotenpaare liegen bis zum Schwellwert 0,002 bei 0 und steigen dann bis 35 bei einem Abstandswert von 0,01 und bis 96 bei einem Wert von 0,015.

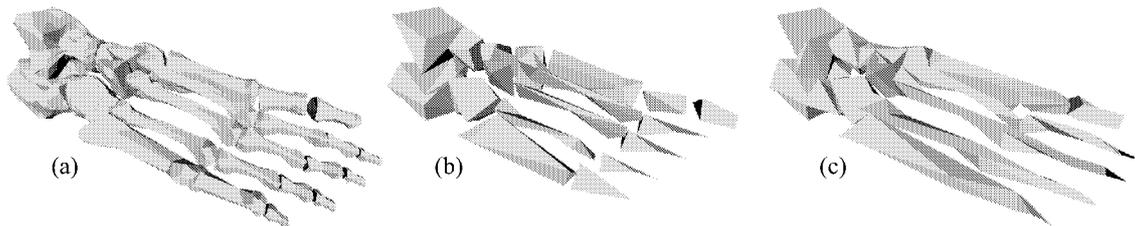


Abb. 4.23: (a) Originalmodell menschlicher Fußknochen mit 4.204 Dreiecken bzw. 2.154 Knoten, (b) Vereinfachung auf 250 Dreiecke bzw. 152 Knoten ohne Schwellwert, (c) Vereinfachung auf 250 Dreiecke bzw. 100 Knoten mit Schwellwert 0,05

Die Untersuchung der Abhängigkeit der Fehler vom Schwellwert wurde am vereinfachten Modell der Fußknochen mit 250 Dreiecken bzw. 177 Knoten vorgenommen. In Abbildung 4.23 sind zwei Modelle zu sehen, die unter anderem zum Vergleich dienen. Das Originalmodell in (a) ist in (b) ohne zusätzliche und in (c) bei einem Schwellwert von 0,05 mit 5.550 imaginären Kanten vereinfacht worden. Alle Bestrafungen sind für die Untersuchung dieses Abschnitts ausgeschaltet gewesen, um den Einfluss des Schwellwertes unverfälscht zu erfassen. Mit Einschaltung der Bestrafung sind die Fehler noch etwas geringer. Der Kurvenverlauf ist aber ähnlich. Insbesondere die Bestrafung der Dreiecksidentität verbessert den Fehler ohne Schwellwert merklich.

Die Anzahl der Zusammenhangskomponenten ist bei dem nicht vereinfachten Knochenmodell 26. Bei der Vereinfachung ohne Verschmelzungskandidaten über den Schwellwert bleiben 18 Zusammenhangskomponenten übrig, da 8 Teilnetze in sich soweit zusammengefallen sind, dass sie keine Dreiecke mehr beinhalten. Durch die Hinzunahme von Verschmelzungskandidaten aufgrund des Schwellwertes verringert sich die Anzahl der Zusammenhangskomponenten tendenziell bis auf 1 bei einem Knotenabstandswert von 0,025. Hierfür ist aber nicht das Wegfallen von Teilnetzen verantwortlich, ganz im Gegenteil bleiben Dreiecke aus kleinen Teilnetzen länger erhalten, sondern das Verschmelzen der Teilnetze.

Die optisch bessere Qualität der Vereinfachung ist gut in der Abbildung 4.23 zu sehen, bei der ohne Knotenabstandswert die kleinen Teilnetze der Zehen verschwinden, durch Hinzunahme von Knotenpaaren über den Abstand hingegen das Originalmodell besser angeglichen wird, indem die einzelnen Knochensegmente verschmelzen. Die weniger werdenden Knoten müssen nicht mehr zur Abgrenzung der nur wenig voneinander entfernten Knochensegmente genutzt werden, sondern sind durch die Verschmelzung frei für die genauere Modellierung der gesamten Oberfläche.

Der durchschnittliche wie auch maximale globale Fehler E_{sum} und E_{max} fallen tendenziell bei der Erhöhung des Schwellwertes. Ab 0,09 bleiben die Fehler stabil. Der maximale Fehler schwankt mehr und besitzt sein gemessenes Minimum zwischen 0,04 und 0,06 und nicht beim stabilen Wert ab dem Schwellwert 0,09.

Die Erhöhung des Schwellwertes verbessert nicht immer den Fehler, was in der Natur des Quadrik-Fehlermaßes liegt, da zwar lokal der bei den Quadriken zu Grunde liegende Ebenenabstand ähnlich dem Abstand zu den Dreiecksflächen ist, diese Ähnlichkeit aber bei Hinzunahme von Punktepaaren, deren Punkte weit von einander entfernt liegen, immer mehr schwindet. Ebenen haben eine unendliche Ausdehnung im Gegensatz zu den Dreiecksflächen, was das Quadrik-Fehlermaß für Punktepaare großer Distanz unzuverlässig werden lässt.

Anhand der nebenstehenden Skizze 4.24 einer extremen Knotenkonstellation im Zweidimensionalen soll die Problematik der Hinzunahme von nicht verbundenen Punktepaaren erläutert werden. v_1 und v_2 liegen zwischen zwei parallelen Geradenstücken. Diese vier Geradenstücke liegen auf einer Geraden. Der Netzabschnitt zwischen v_1 und v_2 beinhaltet aber neben parallelen Geradenstücken auch eine Einbuchtung, dessen Kanten aber nicht direkt an v_1 oder v_2 anliegen. Als Verschmelzungskandidat über den Schwellwert ist die gestrichelte Kante $\{v_1, v_2\}$ zu bewerten und ein Zielpunkt \bar{v} zu berechnen.

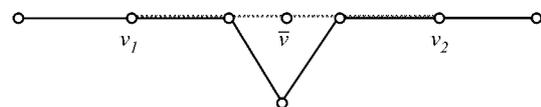


Abb. 4.24: Extreme 2D-Knotenkonstellation mit Schwellwert bedingter Kante $\{v_1, v_2\}$ und ihrem Verschmelzungspunkt \bar{v}

O.B.d.A. sei die Gerade, auf der die Kanten an v_1 und v_2 liegen, die x -Achse. Die Geradenstücke

werden dann alle über die Geradennormalenform $(0 \ 1)(x \ y)^t = 0$ beschrieben. Die Quadriken lauten:

$$Q_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}; \quad Q_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix};$$

$$\bar{Q} = Q_1 + Q_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

$Q(\bar{v}) = \bar{v}^t \bar{Q} \bar{v} = 4y^2$ ist zu minimieren. Dies ist nicht eindeutig lösbar, da x beliebig wählbar, also kein optimaler Verschmelzungspunkt zu finden ist, so dass vom Algorithmus der Mittelpunkt bzw. Endpunkt untersucht wird. Für alle drei Punkte gilt:

$$Q(\bar{v}) = \begin{pmatrix} a & 0 & 1 \end{pmatrix} \bar{Q} \begin{pmatrix} a \\ 0 \\ 1 \end{pmatrix} = 0.$$

Die Kosten von 0 bewirken, dass diese Verschmelzung zu Beginn ausgeführt wird.

Den optisch nicht erwünschten Effekt bei dem Zusammenziehen auf den Punkt \bar{v} ist offensichtlich und führt hier neben dem Verdecken einer Struktur bei Nichtberücksichtigung solcher Effekte zu einer Normalendrehung.

Als Fazit sollte aber festgehalten werden, dass bei bedachter Hinzunahme von Knotenpaaren über einen Schwellwert, die Netzqualität der Vereinfachung gerade bei natürlichen und aus mehreren Komponenten bestehenden Netzen steigt. Zur Untermuerung dient neben den zuvor besprochenen Fußknochen die Abbildung 3.2 in Abschnitt 3.1.2 von dicht beieinanderliegenden Würfeln, die ohne Schwellwert schnell in sich zusammenfallen. Eine Vereinfachung mit Abstandswert gibt sogar bei wesentlich geringerer Netzkomplexität das Originalmodell besser wieder.

Der Schwellwert sollte aber nicht zu groß gewählt werden, da die Qualität irgendwann nicht mehr steigt oder sogar fallen kann. Viel wichtiger ist aber noch, dass die Rechenzeit für die Bestimmung der hinzukommenden Knotenpaare und der Berechnung und Aktualisierung der Kosten unverhältnismäßig ansteigt.

4.4.4 Auswirkungen der Bestrafungen auf die globalen Fehler

Ein letzter Punkt soll sich dem Einfluss der Bestrafungen auf den globalen Fehler widmen. Dass die Randbestrafung für offene Flächen geringere Fehler verursacht, ist offensichtlich, da durch das Ausfransen des Randes die Entfernung vom Originalnetz erheblich steigt.

Diagramm 4.25 zeigt dies für das Blattmodell aus Abbildung 3.8. Die dünnen Linien der Fehler ohne Randbestrafungen liegen um Größenordnungen höher als die dicken Linien der Fehler mit Randbestrafung.

Bei den anderen beiden Bestrafungen der Dreiecksidentität und der Normalendrehung kam es hingegen nicht darauf an, das Modell im geometrischen Rahmen des Originalmodells zu halten, sondern die Einführung wurde rein optisch motiviert. Daher stellt sich die Frage, ob diese Einschränkungen den Fehler negativ beeinflussen.

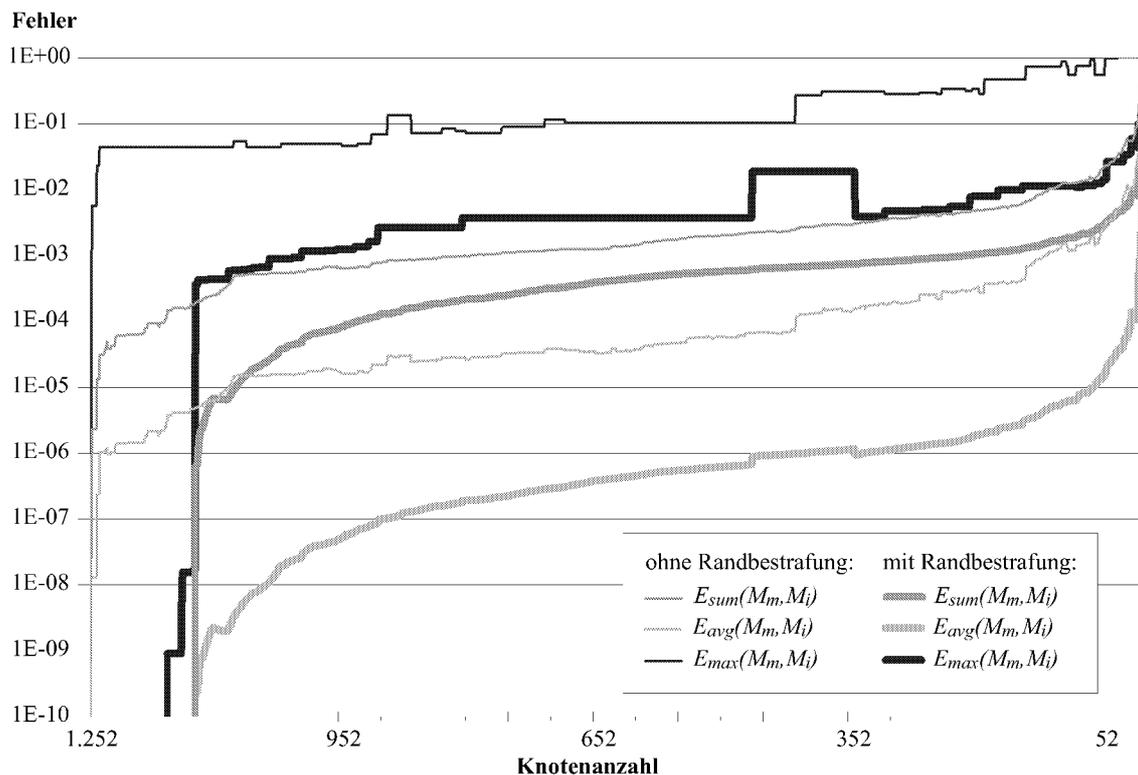


Abb. 4.25: Globale Fehler der Vereinfachung des Blattmodells zum Original als dünne Linien ohne Randbestrafung und als dicke Linien mit aktivierter Randbestrafung

Messungen für das Kuhmodell ergaben, dass mit eingeschalteter Bestrafung der Normalendrehung bereits ab einer Detailstufe mit 2.100 Knoten ein Unterschied der Fehler E_{avg} und E_{sum} mit Bestrafung zu den entsprechenden Fehlern ohne Bestrafungen messbar ist. Diese bleiben aber für alle Detailstufen so gering, dass sie in einem Diagramm mit den gesamten Detailstufen und einem angepassten Fehlerspektrum nicht sichtbar werden.

Für die Bestrafung der Identität ist erst ab einer Detailstufe mit 500 Knoten ein Unterschied zu dem Fehler E_{max} ohne Bestrafung messbar. Dafür ist der Fehlerunterschied aber bei der Detailstufe um 100 Knoten so groß, dass er in einem Diagramm zu sehen wäre und steigt temporär um rund 40%. Da dies aber die einzige Stelle eines Unterschieds ist, wurde auf die Abbildung des Diagramms verzichtet. Insgesamt lässt sich aber für dieses Beispiel kaum ein Unterschied der Fehler feststellen.

Der Fehler mit Bestrafung der Identität hat bei filigranen Modellen wie der Spinne aus Abbildung 3.10 hingegen durchaus positive Wirkung auf die Fehler. Diagramm 4.26 zeigt den Fehlerunterschied für das Spinnenmodell für die letzten 500 Detailstufen, da zwar eine messbare, aber kaum sichtbare Differenz, vorliegt. Die Messung wurde im Gegensatz zum Kuhmodell nur an 12 diskreten Stufen durchgeführt. Es ist aber gut die positive Wirkung der Identitätsbestrafung zum Ende der Vereinfachung zu sehen, da die filigranen Regionen von dem kompletten Zusammenfall möglichst lange zurückgehalten werden. Da es sich nicht um eine komplette Unterdrückung der unerwünschten Kontraktionen handelt, sondern nur um eine Zurückstellung durch Bestrafung, nähern sich die Modelle und damit die Fehler mit und ohne Bestrafung zum Ende wieder an. Die Detailstufen können dabei schon in gewissem Rahmen voneinander abweichen, obwohl das Quadrik-Fehlermaß die Ziele

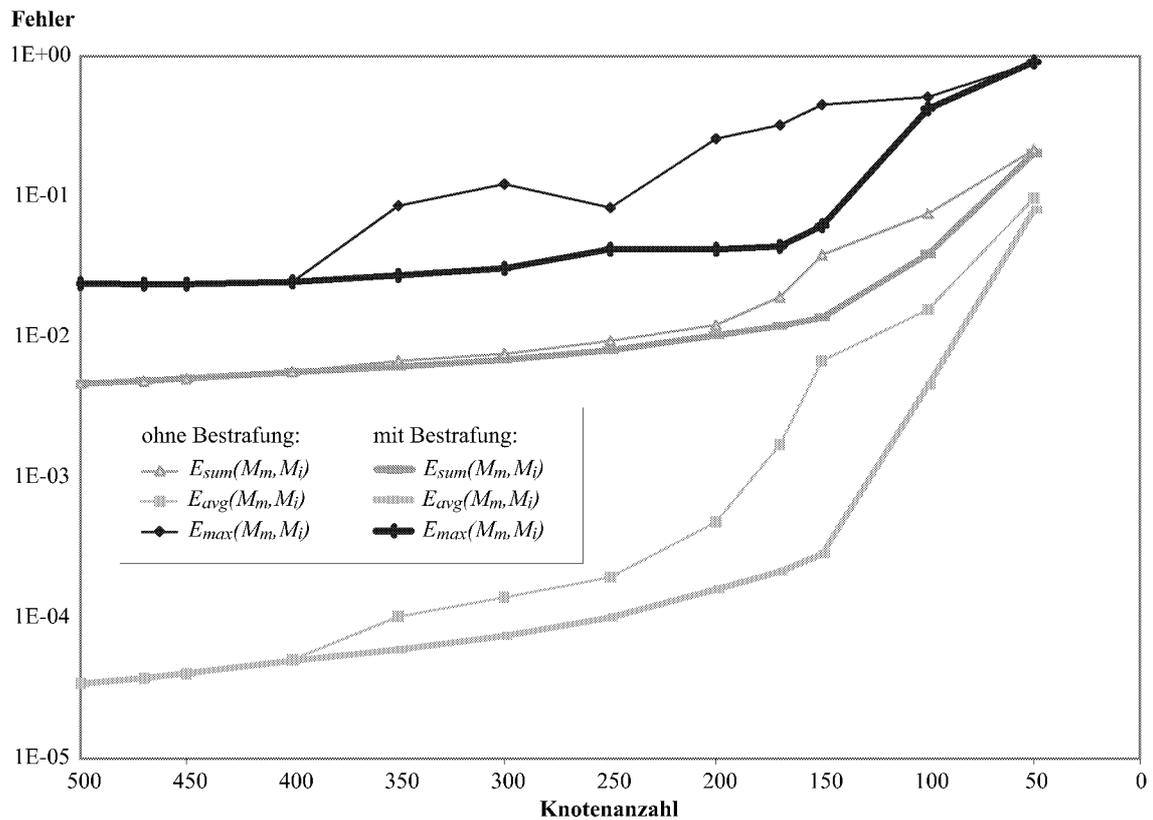


Abb. 4.26: Globale Fehler der Vereinfachung des Spinnenmodells zum Original als dünne Linien ohne Bestrafung der Dreiecksidentität und als dicke Linien mit aktivierter Identitätsbestrafung

nah an den Dreiecksflächen hält, da die Bestrafung gewisse Ebenen mehr gewichtet und dabei durch Neubewertung die Kontraktionsziele verschiebt.

4.5 Quadriken

Die Quadriken sind die zentrale Größe zur Bewertung der Verschmelzungskandidaten. Wie in Abschnitt 3.1.4 erläutert, kommt es in bestimmten Situationen vor, dass es keinen optimalen Verschmelzungspunkt aufgrund des Quadrik-Fehlermaßes oder nur Kandidaten mit nahezu identischen Fehlern gibt. In diesen Fällen ist die Quadrik singulär oder fast singulär. Wann und wie oft Quadriken in der Implementation eine Beinahe-Singularität aufweisen, soll hier besprochen werden. In Tabelle 4.2 sind die Anzahlen für die singulären und regulären Quadriken, die während der Initialisierung und der fortschreitenden Netzvereinfachung bewertet werden, für die untersuchten 9 Modelle aufgelistet.

Die gesamte Anzahl der Quadriken zur Initialisierungszeit stimmen exakt mit der Anzahl an Knotenverschmelzungskandidaten als Summe der Kantenspalten aus Tabelle 4.1 überein. Das Verhältnis der Anzahl der Quadriken zur Simplifizierungszeit zu der Anzahl, die während der Initialisierung bewertet werden, liegt für alle Modelle leicht über 1 : 2,3.

Dieses Verhältnis ergibt sich zum einen aus den im Schnitt 6 Kanten, die an einem Knoten anliegen (siehe Abschnitt 2.7.2) und zum anderen aus der Zahl von Detailstufen. Durch die 6 Kanten pro

Knoten, müssen durchschnittlich 8 Kanten nach einer Kontraktion neu bewertet werden, da jeweils die gemeinsame Kante und zwei nach der Verschmelzung doppelt vorliegende Kanten keiner Neubewertung bedürfen. Während der fortschreitenden Knotenverschmelzung wird zwar die Menge der e Kanten abgearbeitet, aber dabei nur m Detailstufen erstellt, da bei jeder Kontraktion für gewöhnlich ein Knoten, zwei Dreiecke und drei Kanten eliminiert werden, so dass die Abarbeitungsliste der e Kanten in einem Schritt um 3 reduziert wird. Zusammenfassend ergibt sich also ein Verhältnis von

$$e : (2 \cdot 6 - 4)m \approx 3m : 8m \approx 1 : 2,3.$$

Am meisten weichen die Knochen von diesem Verhältnis mit $1 : 2,1$ ab, was an der großen Zahl an Zusammenhangskomponenten liegt, die das Verhältnis der Anzahl zwischen Netzknoten und -kanten zu Gunsten der Knoten gemäß Diagramm 4.5 verschiebt, so dass pro Simplifikationsschritt weniger Kanten neu bewertet werden müssen.

Diagramm 4.27 stellt den prozentualen Anteil an singulären Quadriken an der Gesamtanzahl der Quadrikenbewertungen getrennt für die beiden Phasen der Simplifizierung dar. Da die Zahlen stark voneinander abweichen, ist eine logarithmische Skalierung der Prozente gewählt worden. Zwar ist die Stichprobe der Modelle für allgemeingültige Aussagen gering, trotzdem sollen einige Aussagen aus Diagramm 4.27 abgeleitet werden.

Es fällt auf, dass der prozentuale Anteil der singulären Quadriken für die fortschreitende Vereinfachung immer unterhalb dem Anteil während der Initialisierung liegt. Sei $s \in \mathbb{N}_e$ die Anzahl singulärer Quadriken im initialen Ausgangsnetz, so ist die maximale Anzahl der daraus resultierenden singulären Neubewertungen dann gegeben, wenn die singulär bewerteten Kanten im Netz zusammenhängend vorliegen, damit sie nicht durch reguläre bei einer Neubewertung ausgelöscht werden (siehe Abschnitt 3.2.2). Für zusammenhängende singuläre Gebiete verhält sich die Anzahl der Neubewertungen dann wie für ganze Netze, da auch hier bei jeder Kontraktion 3 singulär bewertete Kanten aus der Kandidatenliste gelöscht werden. Es ergeben sich also pro Schritt rund 8 singuläre Neubewertungen, von denen rund $s/3$ vorkommen können. Der maximale Anteil der Neubewertungen beträgt somit

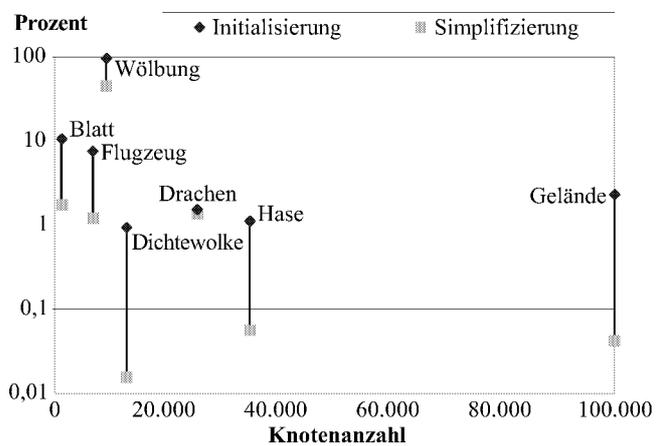


Abb. 4.27: Anteil singulärer Quadriken in Prozent während der Initialisierung und der anschließenden Knotenverschmelzung

$$\frac{8 \frac{s}{3}}{8 \frac{e}{3}} = \frac{s}{e},$$

was dem Anteil an singulären Quadriken im Ausgangsnetz entspricht. Die Singularität kann sich anschaulich gesprochen also nicht weiter ausbreiten.

Das Drachenmodell weist eine erstaunlich große Anzahl an singulären Quadriken auf, obwohl wie vorher erwähnt der Drache eine größere Komplexität als der Hase besitzt und dabei noch eine geringere Knotenzahl aufweist. Die Lösung liegt in der künstlich dem gesannten Drachen hinzugefügten

absolut ebenen Platte aus 409 Knoten und 1.173 Kanten, auf der der Drache steht. Die Anzahl der Plattenkanten entspricht abzüglich der Kanten an den Plattenecken der Anzahl initialer singulärer Quadriken im Drachenmodell. Wegen der eingeschalteten Randbestrafung besitzen die Kanten an den Plattenecken durch zwei Bestrafungsebenen reguläre Quadriken zur Bewertung ihrer Verschmelzungsknoten. Der Drache ohne Platte besitzt somit gar keine singulären Quadriken. Die Platte begründet auch, warum das prozentuale Verhältnis der Quadriken während der fortschreitenden Verschmelzung fast dem Verhältnis zur Initialisierung entspricht. Für die Platte liegt genau der oben beschriebene Fall für eine maximal anzunehmende Zahl an singulären Neubewertungen vor, da die singulär bewerteten Kanten zusammenhängend sind.

Für alle anderen Modelle verringert sich das Verhältnis bei der Knotenverschmelzung, da die Summe einer singulären und regulären Quadrik eine reguläre Quadrik ergibt. Auch der Test einer Quadrik auf die Beinahe-Singularität durch die Determinante kippt nicht zugunsten singulärer Bewertungen, was in Abschnitt 3.2.2 gezeigt wurde. Um so weiter die beiden Prozentwerte in Diagramm 4.27 für ein Modell voneinander entfernt liegen, desto mehr sind die singulären Quadriken über das Modell verteilt. Dies ist insbesondere für überabgetastete Modelle wie dem Hasen der Fall. Ähnliches gilt auch für Modelle wie die Dichtewolke, die durch den Marching-Cube-Algorithmus erzeugt wurden, da auch hier ein starres zum Teil sehr feines Gitter des Volumendatensatzes auf ein möglicherweise einfachen gleichmäßigen Datensatz gelegt wird. Das Gelände hat sehr unterschiedlich große Ausdehnungen, da die Höhe meist viel kleiner ist als die Tiefe und Breite. Dies bewirkt auch hier eine leichter vorkommende Beinahe-Singularität gleichmäßig in Bereichen geringer Höhenunterschiede.

Die Wölbung ist ein weiterer besonderer Fall. Sie besitzt eine gleichmäßig leicht gekrümmte Fläche, die lokal durch fast parallele Ebenen gekennzeichnet ist. Diese Fast-Parallelität ergibt für die Quadriken eine Beinahe-Singularität, die durch die ϵ -Wahl des Determinanten-Tests hervorgerufen wird. Wählt man, statt wie in der Implementation 0,0001, für ϵ den Wert 0,000001, so besitzt auch die Wölbung gar keine singulären Bewertungen mehr. Alle Beinahe-Singularitäten auch anderer Modelle ließen sich durch eine entsprechende Wahl von ϵ als regulär zu bewertende Kontraktionen einstufen. Die große Wahl von ϵ schützt das Verfahren vor unschönen Effekten, die in Abschnitt 3.1.4 angesprochen werden. Die einzigen regulären Quadriken im Modell der Wölbung zur Initialisierungszeit liegen auf dem Rand der Wölbung, von dem exakt 1.132 Kanten ausgehen. Dazu gehören nicht nur die 380 Randkanten, sondern auch die in die Wölbung reinreichenden Kanten, da die Randbestrafung auf die Knoten, die auf dem Rand liegen, wirkt.

Das Blatt und das Flugzeug besitzen verhältnismäßig viele singuläre Quadriken, was aus der konstruktiven Herkunft herrührt. Dabei liegt bei diesen Modellen schon eine angepasste Netzdichte vor, die komplexe Regionen mit mehr Dreiecken ausstattet als einfache. Konstruierte Objekte besitzen häufiger exakte Parallelität von Flächen. Das Gegenteil gilt für komplexe Objekte, die aus realen Messungen herrühren, wie gescannte Modelle oder Isoflächen aus gemessenen Volumendaten, mit einer geringen Knotenzahl. Sie besitzen kaum singuläre Quadriken. Dies sind von unseren Modellen die Kuh und der Fußknochen, die im Diagramm nicht eingezeichnet sind, da sie keine singulären Quadriken enthalten und damit auf einer logarithmischen Skala nicht aufgetragen werden können.

4.6 Bestrafungen

Dieser Abschnitt fällt kurz aus, da die Bestrafung schon im Zusammenhang mit anderen Punkten untersucht wurde. Der Einfluss der Bestrafung auf die Geschwindigkeit ist in Abschnitt 4.2.3 untersucht worden. Die Auswirkung auf die globalen Fehler findet der Leser in Abschnitt 4.4.4 beschrieben.

Randbestrafungen werden genau für Kanten an nur einem Dreieck durchgeführt. Dies schützt die Knoten der Ränder vor der Verschiebung vom Rand, erschwert aber auch das möglicherweise gewünschte Schließen von Löchern oder bei Nutzung des Schwellwertes das Zusammenziehen offener Ränder über imaginäre Kanten. In Diagramm 4.28 sind die einzelnen Bestrafungen pro Knoten des Dreiecksnetzes eingetragen. Die Modelle stehen in der Reihenfolge ihrer Knotenzahl im Diagramm. Für die Randbestrafung haben natürlich das Blatt, die Wölbung und das Gelände mit ihren offenen Seiten die größte absolute Anzahl. Durch Löcher im Boden der Vorlage besitzt auch der Hase einige Kanten, die nur an einem Dreieck anliegen. Die Randkanten im Drachenmodell rühren nur von der künstlich hinzugefügten Ebene, auf der das Modell steht. Bei den drei Modellen mit den offenen Seiten, fällt der Anteil von Randkanten im Verhältnis zur Knotenzahl bei größer werdenden Netzen, da die Anzahl der Randkanten nur linear hingegen die Knotenzahl quadratisch mit der Modellausdehnung ansteigt.

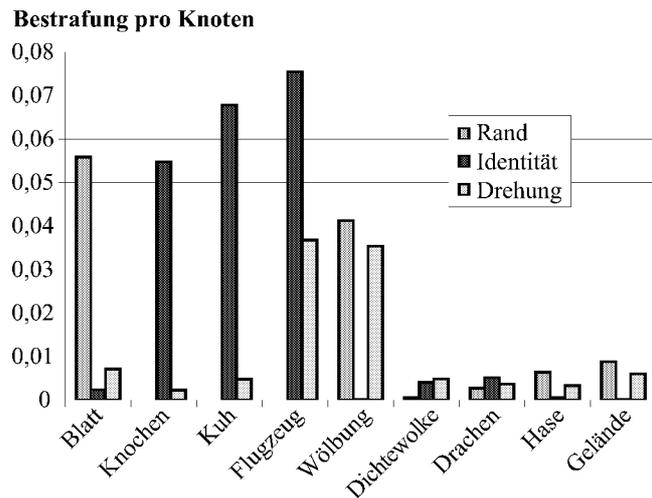


Abb. 4.28: Bestrafungen der Randkanten, der Dreiecksidentität und der Dreiecksdrehung pro Knoten für die untersuchten Modelle in nach Knotenzahl aufsteigender Reihenfolge

Die Bestrafung der Identität findet häufiger in Modellen mit mehreren Zusammenhangskomponenten wie den Knochen und dem Flugzeug und mit vielen ausladenden Details wie bei der Kuh und dem Drachen statt. Diese ausladenden Details kommen zusätzlich bei den Knochen und dem Flugzeug hinzu und begründen auch die Identitäten in der Dichtewolke. Für Mannigfaltigkeiten oder nahezu geschlossene Netze kommt es zum Ende der fortschreitenden Verschmelzung immer zu Dreiecksidentitäten. Modelle mit offenen Seiten und einer hauptsächlich in zwei Dimensionen ausgedehnten Oberfläche, die vermehrt Randbestrafungen aufweisen, haben stattdessen wenig Identitätsbestrafungen. Der Hase hat die ruhigste Oberfläche und dadurch wenig Dreiecksidentitäten.

Drehungen kommen vermehrt in Modellen mit singulären Quadriken vor, da die optimale Knotenverschiebung dadurch verhindert wird. Für die Wölbung verringert sich zum Beispiel die Anzahl an Bestrafungen aufgrund von Dreiecksdrehungen bei Herabsetzen der Schranke für die Beinahe-Singularität, um alle Knotenpositionen optimal zu bestimmen, von 326 auf 200. Aber auch die Form und Verteilung der Dreiecke hat einen Einfluss auf die Häufigkeit einer Dreiecksdrehung. Bei dem Flugzeug dürfte dies ein Grund für die relative Häufigkeit sein, da spitze Dreiecke eher in konstruierten Modellen zu finden sind.

4.7 Adaptive Netzanpassung

Die adaptive Anpassung der Netzdichte aufgrund von Randbedingungen wird in Abschnitt 3.1.7 beschrieben. Hier soll die Auswirkung dieser Netzanpassung an einem real durchgeführten Beispiel erläutert werden. Die Anwendung, in die die Netzanpassung eingebaut wurde, ist SENSITUS, deren Einsatzgebiet in Anhang A.1.3 und deren grafische Oberfläche in A.4 beschrieben wird.

Der Datensatz in der Beispielsitzung ist der Drachen mit seinen 25.418 Knoten im Originalnetz.

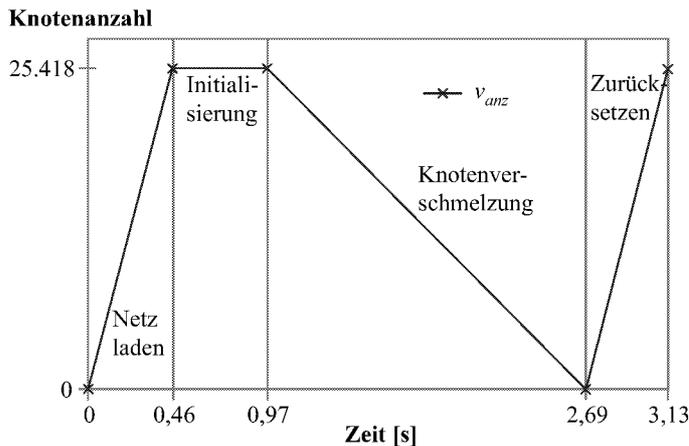


Abb. 4.29: Verlauf der Knotenanzahl des Drachennetzes während des Ladens und der Vereinfachung einschließlich dem Zurücksetzen in den Originalzustand

Diagramm 4.29 zeigt den Verlauf des Netzes anhand seiner Knotenanzahl, den es während des Ladens der SMF-Netzdaten und Vereinfachens einschließlich dem Zurücksetzen nimmt, bevor die adaptive Anpassung starten kann. Die sich in dieser Phase ergebenden Netzdichteschwankungen werden dabei nicht gezeigt, sondern der visuelle Netzstatus nach dem Laden eingefroren, da die fortwährende Neuzeichnung die Verfahrensgeschwindigkeit unnötig verzögern würde. Das Diagramm 4.30 könnte man sich als Fortsetzung des Diagramms 4.29 vorstellen, dabei schließt die Kurve von v_{anz} in Diagramm 4.30 an die in Diagramm 4.29 an.

Die Randbedingung, die es in dieser interaktiven Sitzung einzuhalten galt, war die Bildwiederholrate, die in den 195 Sekunden sieben Mal umgestellt wurde. Die häufige Umstellung der Bedingung ist untypisch, soll aber die Anpassung mehrmals zeigen. Nur zu Beginn und zum Ende erreicht das Netz die volle Knotenanzahl des Originalnetzes. Am Anfang ist die gewünschte Wiederholrate auf 100 fps eingestellt und wird im Laufe der Sitzung auf 50, 150, 50, 200, 50, 100 und 30 gesetzt. Der zu erzielende Wert der Randbedingung r_{goal} ist durch einen dicken waagerechten Strich in Diagramm 4.30 angegeben. Der hinterlegte Bereich um den Zielwert gibt die 25%ige Spanne in beide Richtungen an, die der wirkliche Wert r_{act} von der Zielvorgabe abweichen darf.

Im Diagramm 4.30 zeigen die kleinen Striche auf der Zeitachse an, wann die Überprüfung der Einhaltung der Randbedingung durchgeführt wurde. Diese findet alle 1,5 Sekunden statt und wird entsprechend Abschnitt 4.2.3 in der Geschwindigkeit von knapp 60.000 Detailstufen pro Sekunde durchgeführt. Die Anpassung für dieses Modell in dieser Beispielsitzung geschah somit sehr schnell in Zeiten jeweils unterhalb von 0,1 Sekunden. Die Marker der Kurven zeigen die Stellen an, an denen es für nötig befunden wurde, das Netz anzupassen. Dabei sind für die aktuelle Knotenanzahl v_{anz} bei einer Anpassung zwei Marker für die Knotenanzahl vor und nach der Anpassung eingezeichnet. Die Abstände der Marker voneinander ergeben sich aus den Gleichungen (3.8) und (3.9) im Abschnitt 3.1.7, je nachdem, ob eine Vereinfachung oder Verfeinerung durchzuführen war. Die Kurve von v_{anz} gibt also exakt die Anzahl der Netzknoten wieder, da sich in der Zeit ohne Netzanpassung keine Änderungen der Detailstufe ergeben.

Anders sieht es bei der Kurve des wirklichen Wertes r_{act} der zu berücksichtigenden Randbedingung aus. Diese zeigt nur die zeitliche Abfolge der anzupassenden Werte an. Die alle 1,5 Sekunden ge-

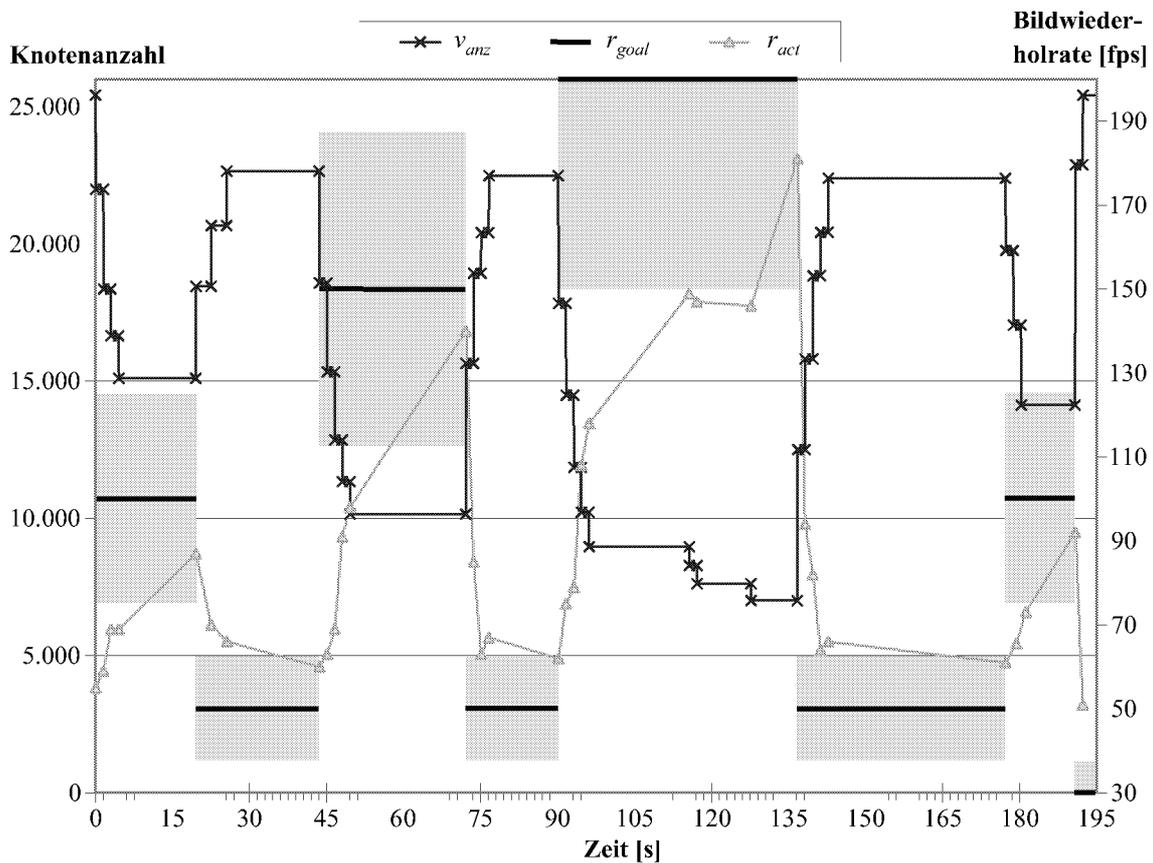


Abb. 4.30: Verlauf der Knotenanzahl des Drachenmodells und der Bildwiederholrate während der adaptiven Netzanpassung unter sich verändernden Vorgaben der Werte der Randbedingung in einer interaktiven SENSITUS-Sitzung

messenen Werte werden für die Kurve nicht herangezogen. Ist kein Marker eingetragen, war keine Netzänderung durchzuführen und die Kurve liegt somit im hinterlegten Schwankungsbereich von r_{goal} . Die Marker von r_{act} sind die Werte, die vor der Netzänderung vorlagen und die adaptive Anpassung veranlassten, die Detailstufe umzustellen. Daher liegen sie alle außerhalb des hinterlegten Bereichs mit Ausnahme des Zeitpunktes der Umstellung der Randbedingung. Zu diesem Zeitpunkt liegt r_{act} am rechten Rand des vorhergehenden Schwankungsbereiches und zeigt damit gut die Erfüllung der Randbedingung vor ihrer Umstellung.

An 34 von 130 Überprüfungen der Randbedingung wurde die Notwendigkeit einer Netzanpassung festgestellt und durchgeführt. Diese Häufigkeit ist, wie zu Beginn gesagt, ungewöhnlich und rührt von der häufigen Umstellung der Randbedingung her. Die Netzanpassungen finden natürlich zu Beginn der Umstellungen statt und brauchen im Schnitt 4 Schritte, bis die Bildwiederholrate das Schwankungsgebiet der Zielvorgabe erreicht. Durch die prozentuale Anpassung von 30% hätte man aber eher einen Wert leicht unterhalb von 3 Schritten bis zum Erreichen der neuen Randbedingung erwartet, denn gemäß Gleichung (3.9) für die Verfeinerung wird die Knotenzahl um $0,3 \cdot v_{anz} \cdot \frac{r_{act} - r_{goal}}{r_{goal}}$ erhöht. Der Faktor $f = \frac{r_{act} - r_{goal}}{r_{goal}}$ liegt dabei durch die Vorgabe des Schwankungsbereichs zwischen

$$r_{act} > 1,25 \cdot r_{goal} \quad \Leftrightarrow \quad f > \frac{1}{4}$$

und, durch die größte Änderung der Zielvorgabe von 200 auf 50 fps,

$$f \leq \frac{200 - 50}{50} = 3.$$

Daraus ergibt sich eine Häufigkeit x der Anpassung von

$$\begin{aligned} v_{anz}(1 + 0,3 \cdot f)^x &= v_{anz}(1 + f) \\ \Leftrightarrow x &= \frac{\log(1 + f)}{\log(1 + 0,3 \cdot f)} \in [2,1, 3,1]. \end{aligned}$$

Ähnliches gilt auch für die Vereinfachung.

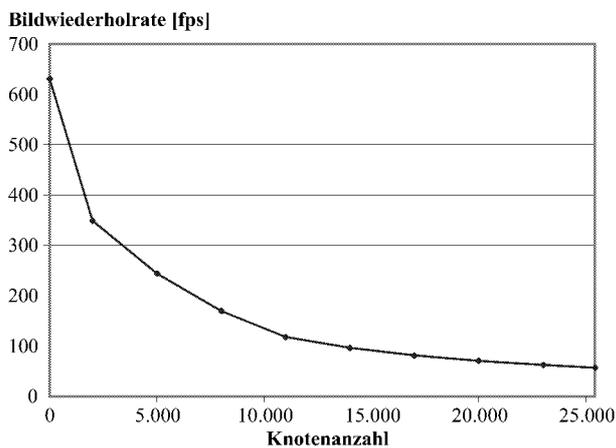


Abb. 4.31: Bildwiederholrate für das Drachenmodell in verschiedenen Detailstufen

Die Differenz könnte auf die kleine Stichprobe an Netzänderungen zurückgeführt werden. Aber auch bei längeren Messreihen ergibt sich ein ähnliches Bild: die Anzahl der Anpassungsschritte liegt im Schnitt bei 4. Zwei Gründe sind daher eher vorstellbar. Zum einen belastet die Netzanpassung selbst das System, so dass in Zeiten von Netzschwankungen die Bildwiederholrate sinkt, und dadurch die Verfeinerung mehr Anläufe braucht. Zum anderen wurde bei den Gleichungen (3.8) und (3.9) davon ausgegangen, dass die Knotenzahl umgekehrt proportional zur Bildwiederholrate ist. Dies ist aber nicht der Fall, da sonst für Knotenzahlen gegen Null die Wiederholrate gegen unendlich streben müsste. Die gemessene Bildwiederholrate für das Drachenmodell in verschiedenen

Detailstufen ist in Diagramm 4.31 zu sehen. Dies ist damit ein weiterer Grund für die schleppendere Anpassung an die Zielvorgabe.

An dem Austrittswert von r_{act} am rechten Rand des erlaubten Schwankungsbereichs ist zu erkennen, dass sich r_{act} im hinterlegten Bereich auf der Seite der Zielvorgabe aufhält, die der vorherigen Einstellung zugewandt ist. Dies führt leichter zu nachträglichen Anpassungen, wie es im fünften hinterlegten Bereich zweimal geschehen ist. Wegen Lastschwankungen des Systems durch eigene Benutzeraktionen oder andere im Hintergrund laufende Prozesse kommt es an solchen Stellen zu Nachbesserungen des wirklichen Wertes der zu berücksichtigenden Randbedingung.

Der zweite, vierte und sechste Wert für r_{goal} ist identisch. Je nachdem, mit welcher Knotenzahl die Netzanpassung das erste Mal die Randbedingung im erlaubten Rahmen erfüllt, stellt sich nicht exakt der gleiche Wert für die Knotenzahl ein. Die Knotenzahlen liegen hier bei 22.664, 22.490 und 22.388. Das gleiche gilt auch für die erste und siebte Anpassung, wo der Unterschied der Knotenzahl mit 15.101 und 14.126 noch gravierender ist. Würde man von verschiedenen Seiten an den gleichen Zielwert versuchen heranzukommen, also zum einen durch eine Verringerung und zum anderen durch eine Erhöhung von r_{goal} , würde der Unterschied noch größer ausfallen.

Am Ende der Sitzung ist der Zielwert mit 30 fps so eingestellt worden, dass das Originalnetz hergestellt werden konnte. Dabei wurde der zu erfüllende Rahmen von r_{goal} nicht erreicht. Da aber keine

Verfeinerung mehr möglich war, wurde die Anpassung ausgesetzt. Ähnliches könnte auch passieren, wenn die Bild- oder Kraftwiederholrate zu hoch eingestellt wird und das Netz komplett eliminiert würde. Auch dann wird die Anpassung eingestellt, obwohl die Zielvorgabe nicht erreicht wurde.

4.8 Probleme des Quadrik-Fehlermaßes

Bisher wurden viele positive Eigenschaften des Verfahrens erwähnt und durch Werte oder Bilder belegt. In diesem Abschnitt sollen aber auch die kritischen Stellen des Verfahrens auf der Basis des Quadrik-Fehlermaßes aufgezeigt werden.

4.8.1 Unruhige offene Ränder

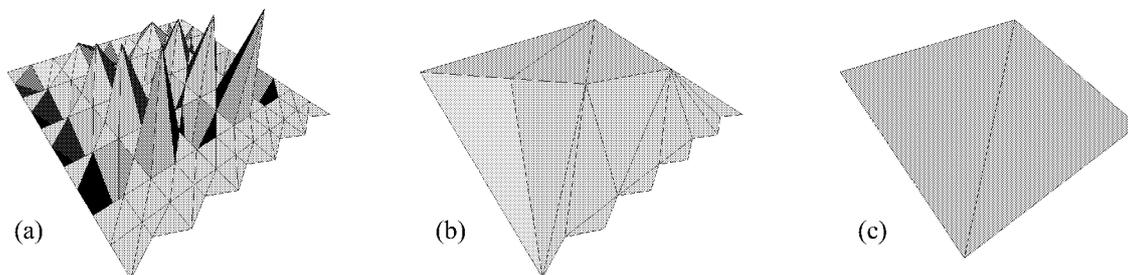


Abb. 4.32: Das Originalmodell aus (a) ist in (b) auf 16 und in (c) auf 4 Knoten vereinfacht worden, ein unruhiger Rand bleibt dabei länger erhalten als starke Unebenheiten innerhalb des Netzes

Bei den Einschränkungen des Verfahrens zur Wahrung von offenen Rändern aus Abschnitt 3.1.5 ist davon gesprochen worden, dass zackige Ränder bei der Bestrafung durch zusätzliche Ebenen ein Problem darstellen. Dazu sei hier ein Beispiel angeben. Abbildung 4.32 zeigt solch ein Modell mit einer zackigen Kante im Vordergrund, bei der es wünschenswert wäre, dass diese Kante zuerst geglättet wird, bevor die starken Unebenheiten in der Oberfläche im Netz verschwinden. Die Glättung findet aber erst nach der kompletten Eliminierung der Unebenheiten ab 16 von 121 Knoten statt.

Durch geringere Gewichtung der zusätzlichen Bestrafungsebenen ließe sich dieses Phänomen zwar reduzieren, aber es würde dazu führen, dass bei sehr feinen Netzen Ränder früher ausfransen, was auch nicht wünschenswert wäre. Es bedürfte also einer adaptiven Anpassung des Gewichtes in Abhängigkeit von der Netzbeschaffenheit. Dies ist aber im Rahmen dieser Diplomarbeit nicht geschehen.

4.8.2 Spitzen

Aufgrund der Natur des Quadrik-Fehlermaßes sind Knoten an sehr spitzen Winkeln problematisch [16]. Abbildung 4.33 zeigt diese Situation im Zweidimensionalen, in der der Knoten v in (a) in einem flachen Gebiet liegt und in (b) eine sehr scharfe Ecke beschreibt. Die anfänglich dem Knoten v zugewiesene Quadrik ist in beiden Fällen aber identisch.

Wandern nun die Geraden der den Knoten v anliegenden Kanten weiter aufeinander zu und lassen in (a) das Gebiet eben werden und in (b) eine Spitze entstehen, so erlaubt das Quadrik-Fehlermaß Punkten sich nahezu ungestraft entlang der die Kanten beschreibenden Geraden zu wandern. Im ersten

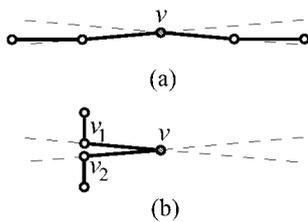


Abb. 4.33: Fast parallele Geraden können Quadriken (a) in flachen Gebieten oder (b) in sehr scharfen Ecken nahezu singular werden lassen

Fall (a) ist dies unproblematisch, da die Punkte in dem flachen Gebiet bleiben (eine mögliche Oberflächenfaltung würde durch die in Abschnitt 3.1.5 beschriebene Abfrage unterdrückt werden), für (b) hingegen können sich Punkte ungeachtet der ursprünglichen Knotenposition von v von dem Originalnetz entfernen. Zum Beispiel würde die Bewertung der Kante (v_1, v) des Netzes in (b) v stark in die Höhe von v_1 und v_2 ziehen, da dies ohne großen Fehler möglich ist. Wandern v_1 und v_2 aufeinander zu, so wird der Fehler immer geringer und das Zusammenfallen der Spitze immer wahrscheinlicher.

Ronfard und Rossignac schlagen in [16] vor, diese scharfen Winkel gesondert zu betrachten oder von der Vereinfachung auszuschließen. Eine weitere Möglichkeit wäre das Hinzufügen von Bestrafungsquadriken an den Spitzenenden wie bei NetZRändern, was die Spitze aber abrunden würde. Die Implementation geht einen für die meisten Fälle sehr viel einfacheren aber effektiven Weg, indem die damit oft einhergehende annähernde Singularität durch eine verhältnismäßig große Schranke bei dem Determinantentest von Seite 58 abgefangen wird und schon bei annähernd singulären Matrizen kein optimaler Verschmelzungsknoten bestimmt, sondern ein expliziter Test ausgewählter Knoten durchgeführt wird.

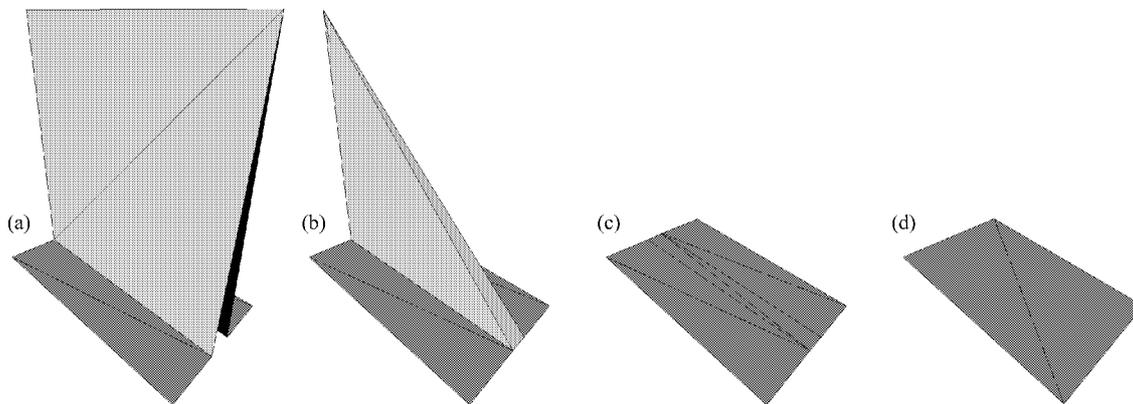


Abb. 4.34: Elimination (a) einer im Original aus 10 Knoten bestehenden Spitze; Verringerung der Knotenanzahl auf (b) 9 (c) 8 und (d) 4

Kommt es wie in Abbildung 4.34 durch die Randbestrafung trotzdem zu regulären Quadriken an Spitzen, sollte dieser seltene Fall explizit über die Analyse von Winkeln der Flächennormalen gesondert behandelt werden, um das Zusammenfallen der Spitze wie in Abbildung 4.34 zu verhindern. Eine leichte Entschärfung des Problems bringt auch schon die im Ausblick 5.2.2 beschriebene Gewichtung der Quadrik mit sich. Eine Implementation dieser Sonderbehandlung ist nicht durchgeführt worden, weil diese Fälle in der Anwendung bisher nicht vorgekommen sind und wegen mannigfaltiger Netztopologie der vom Marching-Cube-Algorithmus erzeugten Isoflächen sehr selten auftreten.

4.8.3 Imaginäre Kanten ohne Randbestrafung

Ist die Randbestrafung deaktiviert, kommt ein weiteres Problem des Quadrik-Fehlermaßes hinzu, wenn eine imaginäre Kante von zwei über einen Schwellwert bestimmten Verschmelzungskandidaten kontrahieren soll und jeweils die Ebenen an den Verschmelzungskandidaten anliegender Dreiecke

ungefähr parallel zueinander sind. Ist der Winkel spitz, in dem sich die Ebenen der beiden Verschmelzungskandidaten schneiden, kann der Verschmelzungspunkt der Kontraktion weit von der ursprünglichen Oberfläche wegliegen.

Abbildung 4.35 zeigt dies im Zweidimensionalen. Verschmelzungskandidaten v_1 und v_2 besitzen jeweils nahezu parallel anliegende Geradenstücke, die ungefähr auf der gestrichelten Geraden liegen. Der Verschmelzungspunkt \bar{v} ist dann der Schnittpunkt dieser zwei Geraden.

Abschließend lässt sich festhalten, dass bei Hinzunahme von Verschmelzungskandidaten über einen Schwellwert die Randbestrafung eingeschaltet werden sollte, um dieses Problem zu beseitigen.

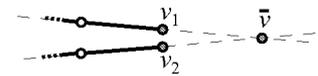


Abb. 4.35: In kantenfremder Verschmelzung zweier flacher Regionen entfernt sich der Kontraktionspunkt ohne Randbestrafung von der Oberfläche

4.8.4 Versetzte Dreiecke an einer Netzspalte

Ein letztes Problem stellen Netzspalten dar, deren angrenzende Knoten besonders weit auseinander liegen, obwohl die Teilflächen einander nah sind. Abbildung 4.36 zeigt ein Beispiel für solch einen Fall. Soll in diesem Beispiel die schmale Spalte durch Angabe eines Schwellwertes geschlossen werden, so ist der Schwellwert sehr hoch zu wählen, um die Lücke zu schließen. Die Problematik eines hohen Schwellwertes ist in Abschnitt 4.4.3 beschrieben und würde auch hier leicht zu Normalendrehungen und unerwünschten Knotenverschmelzungspaaren führen. Solche Probleme treten eher bei konstruierten Modellen auf, da gescannte Objekte meist ein regelmäßig feines Gitter besitzen.

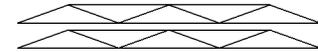


Abb. 4.36: Versetzte Netzspalte

5

Zusammenfassung und Ausblick

5.1 Zusammenfassung

Ziel dieser Arbeit war die Erzeugung einer kontinuierlichen Vereinfachung von Dreiecksnetzen, die im Anschluss der Berechnung eine adaptive Einstellung des Detailgrades des Netzes erlaubt. Benötigt wurde diese adaptive Vereinfachung von der in einer Virtual Reality-Umgebung^G eingesetzten Anwendung SENSITUS, die mittels Krafrückkopplung^G das interaktive Einpassen von exakten Molekülstrukturen in niedrigaufgelöste Aufnahmen von Elektronendichtewolken ermöglicht. Der zur Erzeugung von Isoflächen benutzte Marching-Cube-Algorithmus generiert dabei Netze, die bei großen Volumendaten der Dichtewolke bis zu 200.000 Dreiecke enthalten. Die große Zahl an Dreiecken stellt dabei ein Problem für die flüssige Visualisierung und Kraftausgabe dar.

Für die Netzvereinfachung existieren eine Vielzahl von publizierten Verfahren. Zu Beginn der Arbeit werden einige wichtige repräsentative Methoden vorgestellt und im Anschluss ausführlich miteinander verglichen. Das Vereinfachungsverfahren auf der Basis des Quadrik-Fehlermaßes stellte sich dabei als robust und schnell heraus und erzeugt dabei noch Detailstufen hoher Qualität. Daher dient es als Grundlage für die implementierte Simplifizierung.

Die Vereinfachung mittels des Quadrik-Fehlermaßes bewertet dabei die Qualität einer Detailstufe über den quadratischen Abstand der Netzknoten des simplifizierten Netzes zu Ebenen durch die Dreiecksflächen des Originalmodells. Um die Qualität dieses Maßes zu verstehen, wurde eine allgemeine, motivierende Einführung in geometrische Fehlermaße zwischen zwei Netzen gegeben und zudem ausführlich die Eigenschaften des Quadrik-Fehlermaßes bis hin zum Zusammenhang mit der Oberflächenkrümmung vorgestellt.

Bei der schrittweisen Vorstellung des Vereinfachungsverfahrens sind auch Probleme angesprochen worden, für die teilweise eigene Lösungen erarbeitet wurden. Da als einzige Vereinfachungsoperation eine Verschmelzung von auf einer gemeinsamen Netzkante oder imaginären Kante liegenden Knoten durchgeführt wird, kann durch die kontinuierliche Vereinfachung eine progressive Netzrepräsentation erstellt werden. Diese Repräsentation beinhaltet zum Ende der Vereinfachung, die alle im Netz befindlichen Kanten eliminiert, die gesamte Netzinformation und wird in der Implementation dazu genutzt, um entsprechend vorgegebener Randbedingungen die Detailstufe des Netzes adaptiv einzustellen. Diese Möglichkeit der dynamischen Netzanpassung erlaubt es nun, die Bildwiederholrate so

vorzugeben, dass die Visualisierung den immersiven Eindruck in der Virtual Reality-Umgebung erhält und die Kraftberechnung so häufig durchgeführt werden kann, dass die taktile Wahrnehmung flüssig bleibt.

In den Ergebnissen zeigt sich nochmal anhand von unterschiedlichen Modellen die Generalität, schnelle Geschwindigkeit und gute Qualität des Verfahrens. Die Zeitkomplexität liegt für die Vereinfachung in $O(n \log n)$ und führt auf dem Referenzrechner die komplette Simplifizierung eines Netzes mit rund 200.000 Dreiecken in etwas mehr als 10 Sekunden durch. Die gute Qualität, unterstützt von den eingeführten Bestrafungen und imaginären Kanten, ist sichtbar und wird von den Messungen bestätigt. Dabei erhöht die Wahl des optimalen Verschmelzungspunktes die Qualität erheblich bzw. senkt die zur Qualitätsberechnung herangezogenen globalen Fehler. Dabei fällt insbesondere der im Quadrik-Fehlermaß lokal berücksichtigte quadratische Abstand global sehr gering aus, auch verglichen mit anderen Verfahren. Die aus Gründen von möglichen Systemschwankungen und des visuell weichen Übergangs schrittweise durchgeführte Netzanpassung ermöglicht nach der einmaligen Simplifizierung die adaptive Einstellung der Detailstufe in rund 60.000 Stufen pro Sekunde.

5.2 Ausblick

Das implementierte Verfahren liefert wie ausführlich gezeigt, bereits sehr gute Ergebnisse. Es ist aber an einigen Stellen darauf verwiesen worden, dass bestimmte Aspekte nicht berücksichtigt wurden, da die Anwendung SENSITUS, aufgrund der sich die Notwendigkeit einer Simplifizierung ergab, diese nicht benötigt oder bestimmte Konstellationen bei der Art von Netzen, die der Marching-Cube-Algorithmus erzeugt, nicht vorkommen können.

Diese Einschränkung mag für die eine Anwendung in Ordnung sein. Da aber das entwickelte Modul das unter SENSITUS liegende Grafikpaket SVT um ein adaptiv zu vereinfachendes Dreiecksnetz erweitert, sollte das Modul für andere auf SVT aufsetzende Anwendungen Netze verschiedener Herkunft verarbeiten können. Dass bereits auch andere Netze gut vereinfacht werden können, ist im Ergebnis-Kapitel gezeigt worden. Zudem stellt das verwendete topologietolerante Vereinfachungsverfahren kaum Ansprüche an die Eingabedaten.

Trotzdem würde man sich eine noch allgemeinere Implementation wünschen, die zum Beispiel Materialeigenschaften nicht nur im Netz verwaltet, sondern sie auch bei der Bewertung von Vereinfachungsoperationen berücksichtigt [39, 40]. Des Weiteren wären weitere Bestrafungen optional denkbar, die zum Beispiel zur Topologieerhaltung beitragen können. Am Ende des Ergebnis-Kapitels sind einige Probleme des implementierten Ansatzes angesprochen worden, die zwar für SENSITUS nicht zu berücksichtigen sind, aber für die bessere Tauglichkeit auch für andere Netze einer Antwort in der Implementation bedürfen. Zudem sollte wahlweise auch eine speicher- und nicht geschwindigkeitsoptimierte Vereinfachung durchführbar sein, um ebenso Netze mit Millionen von Dreiecken in gängigen Hauptspeichergrößen simplifizieren zu können.

Natürlich kann das implementierte Modul nicht alle Wünsche erfüllen. Trotzdem sollen im Folgenden drei Erweiterungsideen vorgestellt werden, die teilweise auch für die Anwendung SENSITUS interessant wären.

5.2.1 Selektive Vereinfachung

Aus Gründen der permanenten Änderung der Betrachterposition in einer Virtual Reality-Umgebung^G ist, wie schon in Abschnitt 2.8.3 erwähnt, eine betrachterabhängige Vereinfachung in solch einer Umgebung nicht praktikabel. Trotzdem kann die Idee der betrachterabhängigen Vereinfachung in Form einer selektiven Vereinfachung genutzt werden. Dazu wird das Modell nicht dem sich ständig bewegendem Betrachter angepasst, sondern der Benutzer kann durch interaktive Auswahl Gebiete angeben, die feiner dargestellt werden sollen als das übrige Netz. Diese Auswahl ist in einer Virtual Reality-Umgebung intuitiv direkt durch Anklicken im dreidimensionalen Raum möglich. Aus praktischen Gründen wird dann nicht genau ein Vereinfachungsschritt an dem ausgewählten Punkt durchgeführt, sondern mehrere in der Region um diesen Punkt, wobei die Anzahl der Verfeinerungsschritte mit dem Abstand zum ausgewählten Punkt abfällt.

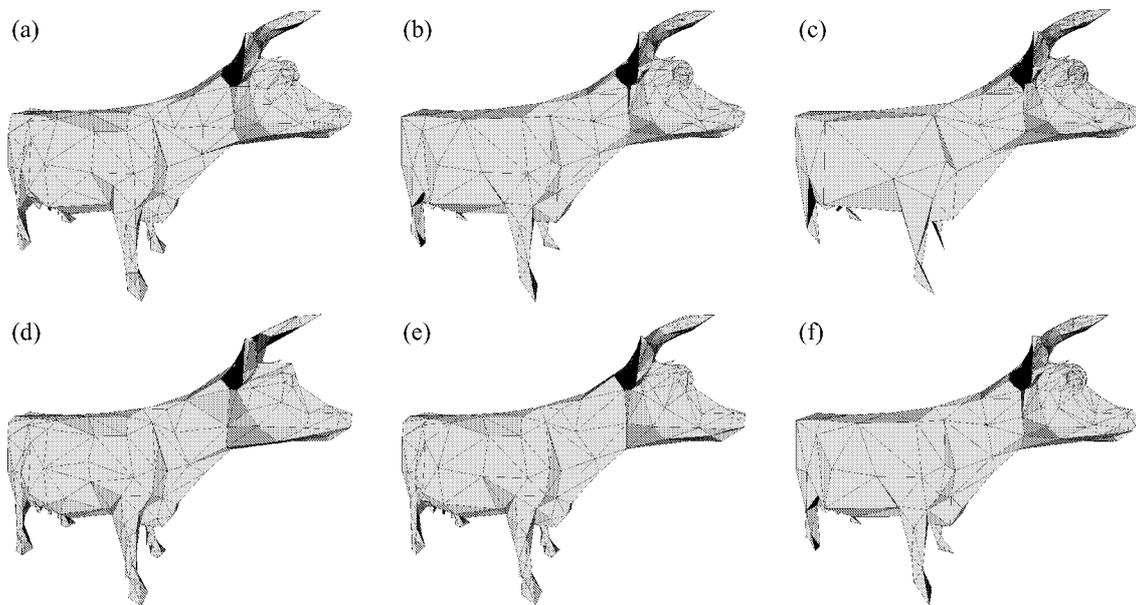


Abb. 5.1: Vereinfachung auf 400 Knoten mit regionaler Gewichtung um eine Hörnerspitze mittels dem Faktor aus Gleichung (5.1); die Parameter sind (a) $a = 1$, $l = 2$, (b) $a = 1$, $l = 4$, (c) $a = 1$, $l = 6$, (d) $a = 10.000$, $l = 4$, (e) $a = 100$, $l = 4$ und (f) $a = 1$, $l = 4$

In Abbildung 5.1 ist die Vereinfachung auf 400 Knoten um die sichtbare Hörnerspitze durch einen Gewichtungsfaktor für die initialen Quadriken für jeden der m Netzknoten in Abhängigkeit vom Abstand zur Spitze v_s , $s \in \mathbb{N}_m$ mit dem Faktor

$$w_i = a + \frac{1}{\|v_i - v_s\|_2^l} \quad i \in \mathbb{N}_m \setminus \{s\}, \quad a \in \mathbb{R}_0^+, \quad l \in \mathbb{Q}^+ \quad (5.1)$$

beeinflusst worden. Für den Knoten v_s sollte mindestens ein Gewicht von

$$w_s = \max_{i \in \mathbb{N}_m \setminus \{s\}} w_i$$

angenommen werden. Durch die Werte a und l kann das Gewicht angepasst werden. Für große a verliert das Gewicht an globaler Bedeutung für die Vereinfachung, für kleine a hingegen bestimmt der Abstand zu Knoten v_s maßgeblich den Quadrik-Fehler. Der Exponent l bestimmt, wie stark und

weit sich das Gewicht um v_s auswirken soll, hat also vornehmlich lokale Bedeutung. Kleine l lassen den Einfluss direkt um v_s sinken, dafür strahlt das Gewicht weiter von v_s aus. Genau andersherum wirkt ein großer Wert für l : das Gewicht für Knoten nahe v_s wird sehr groß, so dass eine Vereinfachung nur schwer möglich ist, dafür fällt der Einfluss des Gewichtes bei der Entfernung von v_s stärker ab.

Die Abbildung zeigt den Einfluss der Parameter a und l auf die Gewichtung. In der oberen Zeile ist a konstant 1 und in deren unteren l konstant 4, dabei wandert oben der Parameter l von 2 in (a) über 4 in (b) bis zu 6 in (c) und unten der Parameter a von (d) 10.000, (e) 100 bis (f) 1. Die Abweichung ist gut sichtbar, wobei (d) fast der Vereinfachung ohne zusätzliche Gewichtung gleichkommt (vergleiche dazu mit der dritten Vereinfachungsstufe aus Abbildung 3.5). Der Einfluss ist modellabhängig und wird stark von der Ausdehnung des Modells beeinflusst, die hier 2 ist.

Für das Beispiel ist eine einfache Möglichkeit der selektiven Vereinfachung gewählt worden. Dazu ist vor der Berechnung der kontinuierlichen Vereinfachung eine Auswahl der länger zu erhaltenden Regionen durchzuführen. Die Möglichkeit ist leicht zu implementieren und schnell zu berechnen. Dafür ist sie aber sehr unflexibel, da für jede Neuwahl von zu erhaltenden Regionen eine Neuberechnung der Vereinfachung nötig ist.

Wesentlich flexibler ist die Nutzung einer betrachterabhängigen Datenstruktur, die es nach der Simplifizierung ohne Neuberechnung erlaubt, regional unterschiedliche Detailgrade anzunehmen. Eine solche Datenstruktur als Knotenbaum HDS ist in Abschnitt 2.7.2 auf Seite 31 vorgestellt worden. Die Implementation ist aber vergleichsweise aufwendig und die Anpassung der Detailstufe nach der selektiven Verfeinerung oder auch Vergrößerung ist zur Laufzeit durchzuführen und bedarf einer längeren Berechnung.

Die regionale Nähe wird bei dieser zweiten Methode nicht wie beim Bestrafungsfaktor über den räumlichen Abstand ermittelt, sondern ergibt sich aus der Verbindung über die Kanten, was die eher gewünschte Vorgehensweise darstellen dürfte. Dadurch ist das Verhalten solch einer betrachterabhängigen Datenstruktur ein anderes als bei der initialen Quadrikgewichtung. Der Einfluss breitet sich also über das Netz und nicht über den Raum aus. Bei der Selektion muss hier daher ein Knoten des Netzes ausgewählt werden. Es darf nicht ein beliebiger Punkt im Raum, wie es bei der ersten Methode möglich wäre, genommen werden.

Einen selektiven Ansatz nutzt auch Gong Li in [61], um in unterschiedlichen Hierarchiestufen des Knotenbaumes Modellierungen am Dreiecksnetz vornehmen zu können (siehe auch Abschnitt 2.8.4).

5.2.2 Gewichtung

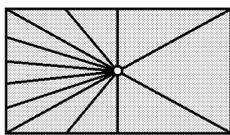


Abb. 5.2: Der linke Teil des Netzes hat trotz gleicher Gesamtgröße für den zentralen Knoten eine dreimal höhere Relevanz als der rechte

Bei dem Fehlermaß über die Volumendifferenz aus Abschnitt 3.2.1 wurde das Quadrik-Fehlermaß durch gewichtete Quadriken verändert. Eine Gewichtung, bei der Volumendifferenzmethode durch das Flächenquadrat der Dreiecke, ist eine sinnvolle Erweiterung des Quadrik-Fehlermaßes, da damit die Strukturempfindlichkeit des Fehlermaßes herabgesetzt wird. An vielen Stellen dieser Arbeit ist auf Vorteile durch eine Gewichtung der Quadriken hingewiesen worden.

Abbildung 5.2 zeigt einen Fall, in dem die Struktur des Dreiecksnetzes sich ungünstig auf die Berechnung der Quadrik auswirkt. Der linke Teil des Netzes hat für den zentralen Knoten ohne Gewichtung eine dreimal höhere Relevanz als der rechte. Dies ist nicht sinnvoll, da zum einen die Dreiecke rechts größer sind

und zum anderen mit dem Knoten einen größeren gemeinsamen Winkel haben. In einem Netz werden bei dem normalen Quadrik-Fehlermaß Bereiche mit feineren Netzunterteilungen stärker gewichtet als mit größeren.

Bei Modellen, die durch 3D-Scanner entstanden sind, spielt dies eine nicht so große Rolle. Bei Isoflächen, die hingegen durch den Marching-Cube-Algorithmus erzeugt wurden, ist zwar die Gebietsaufteilung äquidistant, aber die Dreiecke innerhalb der Zellen können durch die lineare Interpolation der Volumendaten^G große Unterschiede in der Größe aufweisen. Hier wäre es also sinnvoll, kleinere Dreiecke auch geringer zu gewichten. Generell kann man sagen, dass die Form der Oberfläche die Quadrik bestimmen soll und nicht so sehr die Art der Aufteilung in Dreiecke.

Eine Normierung der Gesamtgewichte ist nicht sinnvoll, da sonst Modelle gleicher Struktur aber unterschiedlicher Größe gleiche Quadriken hätten. Sind die unterschiedlich großen Modelle gleichzeitig sichtbar und sollen sie gemeinsam vereinfacht werden, dann wäre es stattdessen sinnvoll, kleinere Modelle geringer zu gewichten, da die Vereinfachung visuell weniger ins Gewicht fällt.

Ein Vorschlag für die Gewichtung der Quadriken über $w_i, i \in \mathbb{N}_k$ gemäß

$$Q(\tilde{v}) = \sum_{i=1}^k w_i D_i^2(\tilde{v})$$

wäre

$$w_i = \frac{\alpha_i}{\pi} \Theta_i.$$

Dabei ist α der Winkel des Dreiecks am zu bewertenden Knoten und Θ die Fläche des Dreiecks. α teilt also die Fläche Θ eines Dreiecks entsprechend den anliegenden Winkeln auf seine drei Knoten auf. Der Winkelfaktor verfälscht dabei nicht die flächenbezogene Gewichtung, weil die Dreiecksfläche über ihre drei Knoten genau einmal voll in die Bewertung einfließt, da die Winkelsumme eines Dreiecks konstant π ist.

5.2.3 Parallelisierung

Bei sehr großen Netzen kann die Wartezeit aus dem interaktiven Zeitfenster geraten. Um bei gleicher Simplifizierungsgüte und heutiger Prozessorgeschwindigkeit schneller zu werden, bleibt nur eine Parallelverarbeitung der Vereinfachung. Der Greedy-Algorithmus ist von seinem Ansatz her nicht parallelisierbar, da eine iterative Kontraktion stattfindet, die Einfluss auf folgende Iterationsstufen des gesamten Netzes hat. Hingegen ist die Initialisierung der Datenstrukturen und Werte mit der Bestimmung der Knotenverschmelzungskandidaten, der initialen Berechnung der Quadriken und der anfänglichen Bewertung der Verschmelzung über den optimalen Verschmelzungspunkt sehr gut zu parallelisieren, da sie nur lokale Netzeigenschaften verwendet.

Bei der rechenaufwendigen Suche nach Punktepaaren, die über einen Schwellwert als Verschmelzungskandidat aufgenommen werden, genügt eine dem Schwellwert entsprechende überlappende Aufteilung des Raumes. In jeder Parzelle kann dann von getrennten Prozessoren nach Kandidaten gesucht werden. Eine Kommunikation zwischen den Parzellen ist nicht nötig. Lediglich bei der am Ende zu erfolgenden Sammlung der Ergebnisse müssen doppelte Einträge wegen der überlappenden Bereiche herausgenommen werden.

Die Berechnung der initialen Quadriken mit möglichen Bestrafungen und der Verschmelzungspunkte für jeden Kandidaten des Originalnetzes zur Bestimmung der Kontraktionskosten kann ebenfalls

räumlich aufgeteilt erfolgen. Hierzu ist noch nicht einmal eine Überlappung der Gebiete erforderlich, da diese Berechnungen lokal geschehen. Eine räumliche Aufteilung ist einer beliebigen Aufteilung vorzuziehen, da so die fundamentale Quadrik einer jeden Ebene nur einmal bestimmt werden muss und für jeden Eckpunkt eingesetzt werden kann.

Der Beschleunigungswert (speed-up factor) der vorgestellten Teilaufgaben des Algorithmus ist wegen ihrer Unabhängigkeit und damit unnötigen Prozessorkommunikation linear zur Anzahl der im Einsatz befindlichen Prozessoren. Dabei ist die Kommunikationszeit für das Verteilen und Einsammeln der Daten und Ergebnisse unberücksichtigt geblieben. Erfolgt die parallele Berechnung direkt auf dem Visualisierungsrechner, handelt es sich bei diesen Rechnern jedoch vornehmlich um Systeme mit gemeinsamen Speicher (shared memory), bei denen keine Datenkommunikation nötig ist. Beim gemeinsamen Speicher ist hingegen auf den schreibenden Zugriff gemeinsamer Daten zu achten.

Sogar die Sortierung der Knotenpaare nach anfallenden Verschmelzungskosten ist parallelisierbar. Dabei ist zum Beispiel mittels des parallelen Quicksort theoretisch eine Laufzeit von $O(\frac{n}{p} \log \frac{n}{p} + \log^2 p + \frac{n}{p} \log p)$ bei p Prozessoren und n zu sortierenden Elementen zu erreichen [85].

Es sind somit die Punkte 1 bis 4 der Verfahrensschritte aus Abschnitt 3.1.8 parallelisierbar. Die darauf folgenden Schritte laufen in voneinander abhängigen iterativer Vorgehensweise ab und sind in der vorgestellten Form nicht parallelisierbar. Die ersten vier Schritte brauchen nach Abschnitt 4.2.3 ungefähr ein Viertel der Rechenzeit der Simplifizierung. Die Gesamtlaufzeit kann somit durch die Parallelisierung leider nicht signifikant gesenkt werden.

Helfen könnte aber eine etwas ungenauere Heuristik, die keinen Greedy-Algorithmus für das gesamte Netz darstellt. Geht man davon aus, dass die Netzaufteilung in Dreiecke einigermaßen gleichmäßig ist, könnte auch für die iterative Knotenverschmelzung eine Gebietsaufteilung zur Parallelisierung erfolgen. Jeder Prozessor simplifiziert dann nur seinen Bereich unabhängig von den anderen Gebieten. Im Randbereich sollte zur Konsistenz der Knotendaten auf gemeinsame Daten zugegriffen werden. Dieser Randbereich müsste sich während der Vereinfachung dem simplifizierten Netz anpassen und sich vergrößern. Bei Systemen mit gemeinsamen Speicher ist darauf zu achten, dass Knoten nur von einem Prozessor insbesondere schreibend benutzt werden.

Für die endgültig Simplifizierung vereint man nun alle Simplifizierungsschritte der einzelnen Parzellen. Dabei kann bei einer gleichmäßigen Netzstruktur einfach Parzelle für Parzelle ein Simplifizierungsschritt in die Gesamtvereinfachung übernommen werden, so dass bei jedem Schritt zyklisch die einzelnen Parzellen zur Verschmelzung durchlaufen werden. Soll die Gesamtvereinfachung genauer sein, müssen die Kosten für die einzelnen Schritte gespeichert werden und diese bei der Vereinigung der partiellen Vereinfachungsfolgen die Reihenfolge in der Gesamtfolge bestimmen. Das Ergebnis wird sich von dem dieser Arbeit zugrundeliegenden Verfahren unterscheiden, da diese Heuristik im Randbereich nicht mit Nachbarschaftszellen kommuniziert, um kostensynchron zu arbeiten. Damit können an diesen Rändern kostspieligere Verschmelzungen vor kostengünstigeren durchgeführt werden. Würde durch die Kommunikation eine Parzelle auf eine Nachbarschaftszelle warten, um im Rand kostensynchron zu vereinfachen, liegt die Vermutung nahe, dass die Effizienz der Parallelisierung stark abfällt. Diese Behauptung bedarf aber noch eines Beweises.

Bisher ist die Parallelisierung nur für ein einzelnes Netz betrachtet worden. Einfacher wird es für Objekte, die in separaten Netzen vorliegen. Hier können die einzelnen Objekte für sich simplifiziert werden und müssen erst zur Darstellung zusammengeführt werden. Eine Arbeit hierzu findet man unter [86].

In [87] wird ein Ansatz zur Parallelisierung des Vereinfachungsverfahrens von Hoppe aus Abschnitt 2.6.2 vorgestellt.

Literaturverzeichnis

- [1] Farin, Gerald: *Kurven und Flächen in Computer Aided Geometric Design*. Vieweg, 1994.
- [2] Seidel, Raimund: *A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons*. *Computational Geometry: Theory and Applications*, 1(1):51–64, 1991.
- [3] Narkhede, Atul und Dinesh Manocha: *Fast polygon triangulation based on Seidel's algorithm*. *Graphics Gems V*, Seite 394–397, 1995.
- [4] Straßer, W., R. Klein, D. Keller und C. Unger: *Graphische Datenverarbeitung II*. Fernuniversität Hagen, FB Informatik, LG Praktische Informatik II, Hagen, 2002. Kurs 1693.
- [5] Hoppe, Hugues: *Progressive Meshes*. *Computer Graphics*, 30(Annual Conference Series):99–108, August 1996. <http://research.microsoft.com/hoppe/>.
- [6] Klein, Rolf: *Algorithmische Geometrie*. Fernuniversität Hagen, FB Informatik, LG Praktische Informatik VI, Hagen, 2002. Kurs 1840.
- [7] Gotsman, C., S. Gumhold und L. Kobbelt: *Simplification and Compression of 3D Meshes*.
- [8] Locher, Franz et al.: *Numerische Mathematik I*. Fernuniversität Hagen, FB Mathematik, LG Numerische Mathematik, Hagen, 2002. Kurs 1271.
- [9] Aspert, N., D. Santa-Cruz und T. Ebrahimi: *MESH: Measuring Errors between Surfaces using the Hausdorff Distance*. In: *Proceedings of the IEEE International Conference on Multimedia and Expo*, Band I, Seite 705 – 708, 2002. <http://mesh.epfl.ch>.
- [10] Cignoni, C. Rocchini P. und R. Scopigno: *Metro: measuring error on simplified surfaces*. Technical Report, Istituto I.E.I.-C.N.R., Pisa, Italy, Januar 1996. Technical Report B4-01-01-96, <http://miles.cnuce.cnr.it/cg/metro.img.html>.
- [11] Csaszar, A.: *General Topology*. Adam Hilger Ltd., Bristol, 1978.
- [12] Soucy, Marc und Denis Laurendeau: *Multiresolution Surface Modeling Based on Hierarchical Triangulation*. *Computer Vision and Image Understanding*, 63(1):1–14, 1996.
- [13] Klein, Reinhard, Gunther Liebich und W. Straßer: *Mesh Reduction with Error Control*. In: *Proceedings of Visualization '96*, Seite 311–318, Oktober 1996.
- [14] Ciampalini, A., P. Cignoni, C. Montani und R. Scopigno: *Multiresolution decimation based on global error*. *The Visual Computer*, 13(5):228–246, 1997.

- [15] Kobbelt, Leif, Swen Campagna und Hans Peter Seidel: *A general framework for mesh decimation*. In: *Proc. Graphics Interface '98*, Seite 43–50, 1998.
- [16] Ronfard, Rémi und Jarek Rossignac: *Full-range approximation of triangulated polyhedra*. Computer Graphics Forum, 15(3), August 1996. Proc. Eurographics '96.
- [17] Luebke, David, Martin Reddy und Jonathan D. Cohen: *Level of Detail for 3D Graphics*. Morgan Kaufmann Publishers, 2003.
- [18] Egerer, Günter, Renate Knecht und Wolfgang E. Nagel: *Algorithmen und Strukturen in C*. Technical Report, Forschungszentrum Jülich.
- [19] Eck, Matthias, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery und Werner Stuetzle: *Multiresolution Analysis of Arbitrary Meshes*. Computer Graphics, 29(Annual Conference Series):173–182, 1995.
- [20] Lounsbery, Michael, Tony D. DeRose und Joe Warren: *Multiresolution Analysis for Surfaces of Arbitrary Topological Type*. Technical Report, Dept. of CS & Eng., U. of Washington, Oktober 1993. TR 93-10-05b, <http://www.cs.washington.edu/research/projects/grail2/www/pub/pub-author.html>.
- [21] Rossignac, Jarek und Paul Borrel: *Multi-resolution 3D approximations for rendering complex scenes*. Technical Report, Yorktown Heights, NY 10598, Februar 1992. IBM Research Report RC 17697. Also appeared in *Modeling in Computer Graphics*, Springer, 1993.
- [22] Heckbert, Paul S. und Michael Garland: *Survey of Polygonal Surface Simplification Algorithms*. Technical Report, 1997.
- [23] Schroeder, William J., Jonathan A. Zarge und William E. Lorensen: *Decimation of triangle meshes*. Computer Graphics (SIGGRAPH '92 Proc.), 26(2):65–70, Juli 1992.
- [24] Cohen, Jonathan, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick Brooks und William Wright: *Simplification Envelopes*. In: *SIGGRAPH '96 Proc.*, Seite 119–128, August 1996. <http://www.cs.unc.edu/geom/envelope.html>.
- [25] Varshney, Amitabh: *Hierarchical Geometric Approximations*. Dissertation, Dept. of CS, U. of North Carolina, Chapel Hill, 1994. TR-050.
- [26] Varshney, Amitabh, Pankaj K. Agarwal, Frederick P. Brooks, Jr., William V. Wright und Hans Weber: *Generating Levels of Detail for Large-Scale Polygonal Models*. Technical Report, Dept. of CS, Duke U., August 1995. CS-1995-20, <http://www.cs.duke.edu/department.html>.
- [27] Popović, Jovan und Hugues Hoppe: *Progressive Simplicial Complexes*. In: *SIGGRAPH 97 Proc.*, Seite 217–224, 1997. <http://research.microsoft.com/hoppe/>.
- [28] Lindstrom, Peter und Greg Turk: *Fast and memory efficient polygonal simplification*. In: *IEEE Visualization*, Seite 279–286, 1998.
- [29] Hoppe, Hugues, Tony DeRose, Tom Duchamp, John McDonald und Werner Stuetzle: *Mesh Optimization*. Computer Graphics, 27(Annual Conference Series):19–26, 1993.
- [30] Boltes, Maik: *Seminarausarbeitung zu Progressive Meshes*. 2002.

- [31] Garland, Michael und Paul S. Heckbert: *Surface Simplification Using Quadric Error Metrics*. Computer Graphics, 31(Annual Conference Series):209–216, 1997.
- [32] Hoppe, Hugues, Tony DeRose, Tom Duchamp, John McDonald und Werner Stuetzle: *Surface reconstruction from unorganized points*. In: *Computer Graphics (SIGGRAPH '92 Proceedings)*, Band 26, Seite 71–78, Juli 1992. <http://research.microsoft.com/hoppe/>.
- [33] Lindstrom, Peter: *Out-of-Core Simplification of Large Polygonal Models*. In: Akeley, Kurt (Herausgeber): *Siggraph 2000, Computer Graphics Proceedings*, Seite 259–262. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
- [34] *Simple Model Format, SMF, version 1.2*, 2003.
<http://basalt.cs.uiuc.edu/garland/class/realtime/dist/SMF.txt>.
- [35] Web3D Consortium: *Virtual Reality Modeling Language (VRML)*, 2003.
<http://www.web3d.org/vrml/vrml.htm>.
- [36] Luebke, David: *Hierarchical structures for dynamic polygonal simplification*. TR 96-006, Department of Computer Science, University of North Carolina at Chapel Hill, 1996.
- [37] Surazhsky, Vitaly und Craig Gotsman: *Explicit Surface Remeshing*. In: *Proceedings of Eurographics Symposium on Geometry Processing*, Seite 17–28, Juni 2003.
- [38] Lee, A., W. Swelden, P. Schröder, L. Cowsar und D. Dobkin: *MAPS - Multiresolution Adaptive Parametrization of Surfaces*. In: *Siggraph 1998, Computer Graphics Proceedings*. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 1998.
- [39] Garland, Michael und Paul S. Heckbert: *Simplifying Surfaces with Color and Texture using Quadric Error Metrics*. In: Ebert, David, Hans Hagen und Holly Rushmeier (Hrsg.): *IEEE Visualization '98*, Seite 263–270, 1998.
- [40] Hoppe, Hugues H.: *New Quadric Metric for Simplifying Meshes with Appearance Attributes*. In: Ebert, David, Markus Gross und Bernd Hamann (Hrsg.): *IEEE Visualization '99*, Seite 59–66, San Francisco, 1999.
- [41] Melax, Stan: *A Simple, Fast, and Effective Polygon Reduction Algorithm*. Game Developer Magazine, 1998.
- [42] Kim, Sun-Jeong, Chang-Hun Kim und David Levin: *Surface simplification using a discrete curvature norm*. Computers and Graphics, 26(Annual Conference Series):657–663, 2002.
- [43] Lindstrom, Peter und Greg Turk: *Image-driven simplification*. ACM Transactions on Graphics, 19(3):204–241, 2000.
- [44] Cohen, Jonathan David: *Appearance-Preserving Simplification Of Polygonal Models*. Dissertation.
- [45] Cignoni, P., C. Montani und R. Scopigno: *A comparison of mesh simplification algorithms*. Technical Report, Istituto I.E.I.-C.N.R., Pisa, Italy, September 1997.
- [46] Lane, T., P. Gladstone, L. Ortiz, J. Boucher, L. Crocker, J. Minguillon, G. Phillips, D. Rossi und G. Weijers: *The independent jpeg group's jpeg software release 6b*, 1998. Software, <ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz>.

- [47] *JPEG image compression FAQ: What is progressive JPEG?*, 2003.
<http://www.faqs.org/faqs/jpeg-faq/part1/section-11.html>.
- [48] Foley, James D., Andries van Dam, Steven K. Feiner und John F. Hughes: *Grundlagen der Computergrafik*. Addison-Wesley, Reading MA, 1994.
- [49] Foley, James D., Andries van Dam, Steven K. Feiner und John F. Hughes: *Computer Graphics: Principles and Practice, 2nd ed.* Addison-Wesley, Reading MA, 1990.
- [50] Straßer, Wolfgang: *Einführung in die Graphische Datenverarbeitung*. Fernuniversität Hagen, FB Informatik, LG Praktische Informatik II, Hagen, 2001. Kurs 1691.
- [51] Straßer, W., R. Klein, D. Keller und C. Unger: *Graphische Datenverarbeitung I*. Fernuniversität Hagen, FB Informatik, LG Praktische Informatik II, Hagen, 2002. Kurs 1692.
- [52] *Geometry Culling in 3D Engines*.
<http://www.gamedev.net/reference/articles/article1212.asp>.
- [53] Luebke, David und Carl Erikson: *View-Dependent Simplification of Arbitrary Polygonal Environments*. *Computer Graphics*, 31(Annual Conference Series):199–208, 1997.
- [54] Xia, Julie C. und Amitabh Varshney: *Dynamic View-Dependent Simplification for Polygonal Models*. In: Yagel, Roni und Gregory M. Nielson (Hrsg.): *IEEE Visualization '96*, Seite 335–344, 1996.
- [55] Xia, J. C., J. El-Sana und A. Varshney: *Adaptive Real-Time Level-of-Detail-Based Rendering for Polygonal Models*. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):171–183, /1997.
- [56] Ugur Gündükbay, Okan Arikan, Bülent Özgüç: *Visualizer: a mesh visualization system using view-dependent refinement*. *Computers and Graphics*, 26:491–503, 2002.
- [57] Hoppe, Hugues: *View-Dependent Refinement of Progressive Meshes*. *Computer Graphics*, 31(Annual Conference Series):189–198, 1997. <http://research.microsoft.com/hoppe/>.
- [58] Luebke, David und Carl Erikson: *View-Dependent Simplification of Arbitrary Polygonal Environments*. In: *SIGGRAPH 97 Proc.*, Seite 199–208, August 1997.
- [59] Delonge, Frank: *Betrachtungsabhängige Visualisierung von Höhenfeldern mittels Wavelet-basierter Methoden zur Detailkontrolle*, 2002.
- [60] Pajarola, R.: *FastMesh: Efficient View-Dependent Meshing*. Seite 22–30.
- [61] Li, Gong und Benjamin Watson: *Semiautomatic simplification*. In: *Symposium on Interactive 3D Graphics*, Seite 43–48, 2001.
- [62] Kobbelt, Leif, Swen Campagna, Jens Vorsatz und Hans-Peter Seidel: *Interactive Multi-Resolution Modeling on Arbitrary Meshes*. *Computer Graphics*, 32(Annual Conference Series):105–114, 1998.
- [63] Zorin, Denis, Peter Schröder und Wim Sweldens: *Interactive Multiresolution Mesh Editing*. *Computer Graphics*, 31(Annual Conference Series):259–268, 1997.

- [64] Rossignac, Jarek: *Edgebreaker: Connectivity Compression for Triangle Meshes*. IEEE Transactions on Visualization and Computer Graphics, 5(1):47–61, /1999.
- [65] Taubin, Gabriel und Jarek Rossignac: *Geometric compression through topological surgery*. ACM Transactions on Graphics, 17(2):84–115, 1998.
- [66] Hoppe, Hugues: *Optimization of Mesh Locality for Transparent Vertex Caching*. In: Rockwood, Alyn (Herausgeber): *Siggraph 1999, Computer Graphics Proceedings*, Seite 269–276, Los Angeles, 1999. Addison Wesley Longman.
- [67] El-Sana, J., E. Azanli und A. Varshney: *Skip-Strips: Maintaining Triangle Strips for View-Dependent Rendering*. In: *IEEE Visualization '99*, Seite 131–138, 1999.
- [68] Ribelles, Belmonte Remolar: *Multiresolution Triangle Strips*.
- [69] Ribelles, J., A. Lopez I. Remolar O. Belmonte M. Chover: *Multiresolution Modelling of Polygonal Surface Meshes Using Triangle Fans*. Lecture Notes in Computer Science N.1953, Seite 367–372, 2000.
- [70] Ribelles, J., A. Lopez, O. Belmonte, I. Remolar und M. Chover: *Multiresolution modeling of arbitrary polygonal surfaces: a characterization*. Computer and Graphics, 26:449–462, 2002.
- [71] Luebke, David P.: *A Developer's Survey of Polygonal Simplification Algorithms*. IEEE Computer Graphics and Applications, Seite 24–35, Mai 2001.
- [72] Agarwal, Pankaj K. und Subhash Suri: *Surface Approximation and Geometric Partitions*. In: *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, Seite 24–33, 1994. (Also available as Duke U. CS tech report, <ftp://ftp.cs.duke.edu/dist/techreport/1994/1994-21.ps.Z>).
- [73] Cormen, Thomas H., Charles E. Leiserson und Ronald L. Rivest: *Introduction to Algorithms*. MIT Press, Cambridge, MA, 1990.
- [74] Erikson, Carl und Dinesh Manocha: *GAPS: general and automatic polygonal simplification*. In: *Symposium on Interactive 3D Graphics*, Seite 79–88, 1999.
- [75] Pumplün, Dieter, et al.: *Lineare Algebra II*. Fernuniversität Hagen, FB Mathematik, LG Numerische Mathematik, Hagen, 2003. Kurs 1103.
- [76] Bronstein, I. N., et al.: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 2000.
- [77] Locher, Franz et al.: *Numerische Mathematik II*. Fernuniversität Hagen, FB Mathematik, LG Numerische Mathematik, Hagen, 2002. Kurs 1372.
- [78] Golub, Gene H. und Charles F. Van Loan: *Matrix Computations*. The Johns Hopkins University Press, 1996.
- [79] Blinn, J.: *The algebraic properties of homogeneous second order surfaces*, 1984.
- [80] Fischer: *Lineare Algebra, Vieweg Studium, Nr.17*. Vieweg, 1998.
- [81] Bronstein, I. N. und K. A. Semendjajew et al.: *Taschenbuch der Mathematik*. B. G. Teubner Verlagsgesellschaft, Verlag Harri Deutsch, 1991.

- [82] Heckbert, Paul S. und Michael Garland: *Optimal triangulation and quadric-based surface simplification*. Computational Geometry, 14(1-3):49–65, 1999.
- [83] Roth, Fabrice: *Algorithmen zur Konstruktion von Isoflächen aus dreidimensionalen Volumendatensätzen*.
- [84] STLport Consulting: *STLport, Standard Library Project is industry standard for multiplatform development*, 2003.
<http://www.stlport.org/>.
- [85] Damaschke: *Parallele Algorithmen*. Fernuniversität Hagen, FB Informatik, LG Theoretische Informatik II, Hagen, 2003. Kurs 1824.
- [86] Schwarze, Tino: *Paralleles, adaptives Level-of-Detail für VR-Simulationen*, 2002.
- [87] Langis, Christian, Gerhard Roth und Frank Dehne: *Mesh simplification in parallel*.
- [88] Zentralinstitut für Angewandte Mathematik (ZAM), Forschungszentrum Jülich.
<http://www.fz-juelich.de/zam/>.
- [89] Birmanns, Stefan, Herwig Zilken und Frank Delonge: *Grafikbibliothek Scientific Visualization Toolkit (SVT)*. Forschungszentrum Jülich.
<http://www.fz-juelich.de/vislab/virtual/svt/>.
- [90] *Virtual Reality Systeme des ZAM*. <http://www.fz-juelich.de/vislab/virtual/>.
- [91] *Interactive Data Language (IDL)*. <http://www.rsinc.com/idl/>.
- [92] *Advanced Visual Systems (AVS)*. <http://www.avs.com/>.
- [93] *Visualization ToolKit (VTK)*. <http://public.kitware.com/VTK/>.
- [94] Schroeder, William J., Ken Martin und Bill Lorensen: *The Visualization Toolkit, An Object-Oriented Approach To 3D Graphics, 2nd edition*. Prentice Hall, 1998. Code at <http://www.cs.rpi.edu:80/~martink/>.
- [95] Schroeder, William J., Lisa S. Avila, Kenneth M. Martin, William A. Hoffman und C. Charles Law: *The Visualization Toolkit User's Guide*. Kitware, Inc., 2001.
- [96] *SENSITUS*. <http://www.sensitus.de/>.
- [97] Birmanns, Stefan: *Haptisches Rendern zum Einpassen von hochaufgelösten Molekülstrukturdaten in niedrigaufgelöste Elektronenmikroskopie-Dichteverteilungen*. Dissertation.
- [98] *PHANTOM* Kraftrückkopplungsgerät der Firma SensAble Corporation.
<http://www.sensable.com/products/phantom.ghost/phantom.asp>.
- [99] Lorensen, W. E. und H. E. Cline: *Marching cubes: A high resolution 3D surface construction algorithm*. In: *Proceedings of SIGGRAPH '87*, Seite 163–169, 1987.
- [100] Silicon Graphics, Inc.: *Standard Template Library Programmer's Guide*, 2003.
<http://www.sgi.com/tech/stl/>.

-
- [101] Silicon Graphics, Inc.: *OpenGL, environment for developing portable, interactive 2D and 3D graphics applications*, 2003.
<http://www.opengl.org/>, <http://www.sgi.com/software/opengl/>.
- [102] Trolltech: *Qt, a C++ toolkit for multiplatform GUI and application development*, 2003.
<http://www.trolltech.com/products/qt/>.
- [103] Garland, Michael: *Sample Data in SMF format*, 2000.
<http://graphics.cs.uiuc.edu/~garland/research/quadrics.html>.
- [104] Luebke, David, et al.: *Resources for level of detail and surface simplification techniques in 3D graphics*, 2000. <http://lodbook.com/models/>.
- [105] *Unified Modeling Language (UML)*. <http://www.omg.org/uml/>.
- [106] *Together*, 2002. <http://www.togethersoft.com/>.



Implementationsbeschreibung

A.1 Voraussetzungen

A.1.1 SVT

Am Zentralinstitut für Angewandte Mathematik (ZAM) des Forschungszentrums Jülich [88] ist ein eigenes Grafikpaket Scientific Visualization Toolkit (SVT) [89] entwickelt worden, damit eine kostenlose und einfach anzupassende Programmierumgebung zum Betrieb der Virtual Reality-Systeme^G [90] zur Verfügung steht. Andere existierende Systeme sind entweder sehr kostspielig, schlecht anpassbar oder nicht im Bereich der Virtual Reality einsetzbar, zum Beispiel wegen fehlender Unterstützung von mehreren Displayflächen. Die Szenengraph-basierte C++ Klassenbibliothek SVT ist plattformunabhängig und setzt auf OpenGL auf.

Das dieser Arbeit zugrundeliegende Modul erweitert SVT um die Möglichkeit der automatischen Netzvereinfachung. Bei einem Netz wird dabei immer von einem Dreiecksnetz ausgegangen. Dies stellt aber keine Einschränkung der Benutzbarkeit für beliebige polygonale Netze dar, da jedes polygonale Netz durch Triangulation^G in ein Dreiecksnetz überführt werden kann.

A.1.2 Andere Softwarepakete

Auch andere bekannte Softwarepakete besitzen Erweiterungen zur Netzvereinfachung.

IDL [91] besitzt die Funktionen `mesh_decimate` zur Verringerung der Dreiecksanzahl innerhalb eines Netzes. AVS [92] wartet mit zwei Modulen auf, die beide auf das in Abschnitt 2.5.1 vorgestellte Verfahren der Knotendezimierung von Schroeder aufsetzen: das Modul `surface_reduction` und das die Attribute berücksichtigende Modul `data_surface_reduction`.

Das größtenteils kostenlose Grafikpaket `vtk` [93, 94, 95] bietet die umfangreichste und am besten dokumentierte Sammlung von Klassen zur Netzvereinfachung. `vtkDecimate` beruht auf dem Verfahren in Abschnitt 2.5.1 von Schroeder, welches durch einen globalen Fehler und der Möglichkeit zur Topologieveränderung erweitert wurde. Ähnlich der vorangegangenen Klasse arbeitet

`vtkDecimatePro`, erzeugt zusätzlich aber ein Progressive Mesh (zur Erläuterung siehe 2.6.2). Die Klasse `vtkQuadricDecimation` basiert auf dem im Abschnitt 2.6.3 vorgestellten Verfahren von Garland und Heckbert, welches wie im Aufsatz von Hoppe [40] durch die Hinzunahme von Attributen erweitert wurde. `vtkQuadricClustering` ist die Klasse mit dem schnellsten Vereinfachungsalgorithmus, welche das Verfahren von Lindstrom [33] beinhaltet. Es ist gut für die schnelle Simplifizierung großer Netze geeignet und ermöglicht auch die Bearbeitung von Teilnetzen.

A.1.3 SENSITUS

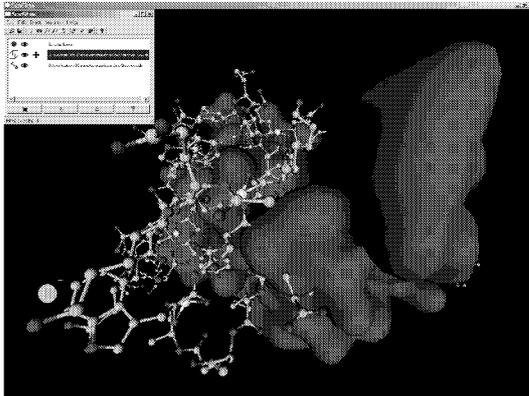


Abb. A.1: Anwendung SENSITUS mit dem hochauflösten Datensatz als CPK Modell und der Isofläche der niedrigauflösten Elektronendichteaufnahmen

Die Notwendigkeit der Simplifizierung ergab sich bei der auf SVT aufsetzenden Anwendung SENSITUS [96, 97], die das interaktive Einpassen von exakten Molekülstrukturen in niedrigaufgelöste Aufnahmen von Elektronendichtewolken mittels Krafrückkopplung^G ermöglicht.

Die Darstellung der Elektronendichtewolke geschieht durch eine Isooberfläche aus den Volumendaten^G der Dichteverteilung. Die Isooberfläche wird mittels Marching-Cube-Algorithmus (Kurzbeschreibung des Algorithmus im folgenden Abschnitt) erzeugt und besitzt somit unabhängig von der Oberflächenbeschaffenheit sehr viele Dreiecke auf einem äquidistanten Gitter.

Die sehr große Anzahl der Dreiecke insbesondere bei großen Molekülen bringt nun Probleme in der schon in der Einführung beschriebenen Interaktivität mit sich. Aber auch für die Krafrückkopplung^G, die den Anwender fühlen lassen soll, wo das Molekül in die niedrigauflöste Dichtewolke hineinpasst, ist eine Auslastung des Rechners allein für die Darstellung schadhaft, da möglichst viel Rechenleistung für die Krafrückkopplung benötigt wird. Die taktile Wahrnehmung besitzt eine zeitlich viel feinere Auflösung als die optische Wahrnehmung und muss statt 25 Mal pro Sekunde 1000 Mal pro Sekunde aktualisiert werden, um einen flüssigen Gefühlseindruck zu vermitteln. Die Krafrückkopplung geschieht mittels eines roboterähnlichen Gerätes namens PHANTOM [98].

A.1.4 Marching-Cube-Algorithmus

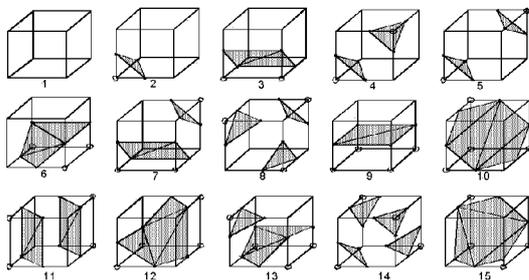


Abb. A.2: Tabelle der 14 Äquivalenzklassen zur Triangulierung und dem Fall, dass die Isooberfläche gar nicht durch das Voxel geht

Der Marching-Cube-Algorithmus ist eines der bekanntesten und meistverwendeten Verfahren zur Volumenvisualisierung, zum Beispiel von Tomographie-Schichtbildern oder Elektronendichteverteilungen. Über einen Schwellwert wird eine Isofläche definiert, die das Verfahren als Dreiecksnetz approximiert.

Jedes Voxel^G des Datensatzes wird dazu iterativ durchlaufen, klassifiziert und über eine Tabelle entsprechend A.2 dessen Triangulierung^G bestimmt. Die genaue Position der Knoten und die Dreiecksnormalen werden durch lineare Interpolation berechnet. Die

256 möglichen Klassifizierungen lassen sich durch Komplementbildung, Spiegelung und Rotation auf 14 Äquivalenzklassen reduzieren [99].

A.1.5 Hilfsmittel

Als Programmiersprache ist die Sprache C++ gewählt worden, die gerade im Grafikbereich durch ihre Objektorientiertheit häufig verwendet wird, da sie einen natürlichen Umgang mit den Grafikobjekten ermöglicht. In vielen Bereichen ist zudem die C++-Standard-Musterklassen-Bibliothek (STL, Standard Template Library) [100] benutzt worden. Es sind verschiedene Compiler unter den Betriebssystemen IRIX (SGI), Linux und Windows eingesetzt worden, um die Software auf den für die Anwendung wichtigen Plattformen zu übersetzen. Die Grafikausgabe setzt auf OpenGL [101] und dem von SVT gelieferten Szenengraphen^G auf. Die grafische Oberfläche ist mit Qt [102] programmiert worden.

Von den oben aufgeführten Betriebssystemen war Windows die vordringlichste Arbeitsplattform, unter der auch alle Beispiele ausgeführt und alle Ergebnisse ermittelt wurden. Als zugrundeliegender Rechner diente ein IBM-Laptop mit folgenden Eckdaten:

<i>Bauart</i>	Laptop
<i>Hersteller</i>	IBM
<i>Modell</i>	ThinkPad A31p
<i>Betriebssystem</i>	Microsoft Windows XP Professional Version 5.1.2600 2002 SP1
<i>Prozessor</i>	Intel Pentium 4 Mobile CPU, 1,8 GHz
<i>Hauptspeicher</i>	512 MB RAM
<i>Platte</i>	IDE ATA/ATAPI-Controller, NTFS-Dateisystem
<i>Grafikkarte</i>	ATI Mobility FireGL 7800, 64 MB

Die neben der eigentlichen Anwendung verwendeten Modelldaten sind von Hand erstellt oder den Quellen [103, 104] entnommen worden.

Die nicht vom Bildschirm abgegriffenen Bilder stammen teilweise aus fremden Publikationen. Die Herkunft ist für diese Fälle immer unter dem Bild angegeben. Eigene Illustrationen sind bei Beispielen mathematischer Herkunft mit Maple erstellt, ansonsten mit Adobe Illustrator gezeichnet worden. Diagramme haben ihren Ursprung in Microsoft Excel.

Diese Publikation ist in \LaTeX gesetzt.

A.1.6 Zeitschriften und Tagungen

Die in dieser Arbeit beschriebene Netzvereinfachung gehört in das Gebiet der Geometrieverarbeitung (geometry processing) oder der Oberflächenmodellierung (shape modeling) und dort zu den Techniken der Mehrfachauflösung (multiresolution, level of detail).

Die zitierten Publikationen entstammen hauptsächlich folgenden Zeitschriften bzw. Kongressberichten:

- Computer Graphics, ACM Siggraph
- Computer & Graphics

- IEEE Computer Graphics
- Computer Graphics Forum
- IEEE Computer Graphics & Applications
- ACM Transactions on Graphics

Die wichtigen Konferenzen, Kongresse, Symposien und Schulen im Bereich der Computergrafik und im Speziellen der Geometrieverarbeitung und Mehrfachauflösung sind:

allgemeine Computergrafik:

- ACM Siggraph
- Winter School of Computer Graphics (WSCG)
- IEEE Visualization
- Eurographics

Geometrieverarbeitung:

- Shape Modeling (SMI)
- Geometry Processing

Mehrfachauflösung:

- Multiresolution in Geometry Modeling (Mingle)

A.2 Modulaufbau und Datenstrukturen

A.2.1 Modulaufbau

Die dieser Arbeit zugrunde liegende Programmierarbeit lässt sich in zwei Teile gliedern. Zum einen ist das Grafikpaket SVT um das Grafikprimitiv Dreiecks-Netz erweitert worden und zum anderen wurde auf der Anwendungsseite SENSITUS die Integration und Steuerung dieses Netzes ermöglicht. Hauptaugenmerk lag dabei auf der in einer VR-Umgebung angepassten Vereinfachung des Netzes.

Folgende Diagramme stellen in Abbildung A.3 zum einen die Klassenstruktur der grafischen Oberflächen zur Steuerung und Visualisierung der Netze im Programm SENSITUS und zum anderen in Abbildung A.4 die Klassenstruktur des Netzes und dessen Vereinfachung in Form von UML-Diagrammen (Unified Modeling Language; standardisierte Sprache zur Beschreibung von Softwaresystemen) [105] dar. Die zentrale Klasse ist `svt.mesh`.

In den einzelnen Klassenblöcken steht ganz oben neben dem Klassennamen gegebenenfalls die Klasse, von der diese abgeleitet wurde. Darunter folgt ein Block mit den Datenelementen (data members) und ein Block mit den Methoden (member functions). Die Pfeile stellen die Beziehungen der Klassen wie Assoziation, Aggregation, Komposition und Vererbung dar. Die Kardinalzahlen an den Pfeilen geben die Anzahl der in Beziehung stehenden Elemente an.

A.2.2 Datenstrukturen

Die zentrale Klasse ist das Dreiecksnetz, das wiederum vier wichtige Klassen enthält. Diese vier Klassen sind das Dreiecksfeld T , das Knotenfeld V , die Liste der imaginären Kanten und Netzkanten E und die PM-Klasse P .

Der Aufbau der Klassen und die Beziehung der Klassen untereinander werden vereinfacht in Abbildung A.5 dargestellt. In der Abbildung sind nur die wichtigsten Klassenelemente aufgenommen worden. Pfeile deuten Verweise oder Verkettungen an. Der hinterlegte Bereich wird nur zum Zeitpunkt der Berechnung der Vereinfachung gebraucht und kann, nachdem die Vereinfachung in P abgelegt wurde, freigegeben werden.

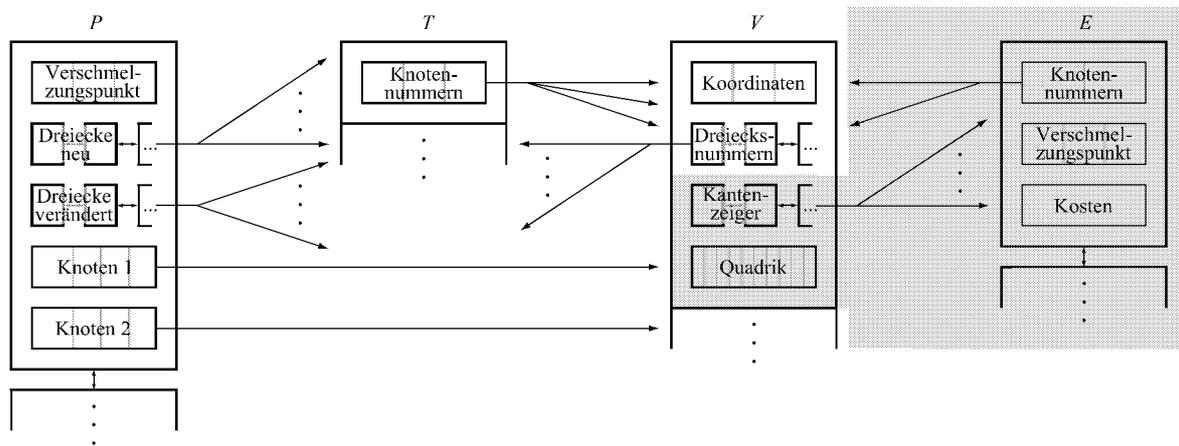


Abb. A.5: Beziehungen der vier wichtigsten Klassen des Dreiecksnetzes

Die PM-Klasse und die Arbeitsweise desselben ist in Abschnitt 3.1.6 beschrieben worden. Die Dreiecks- und Knotenvektoren sind Vektoren, da ein schneller direkter Zugriff auf die Elemente wichtig ist.

Bei der nach Kosten sortierten Liste der Knotenverschmelzungskandidaten E wurde bisher immer von einer Liste gesprochen, da diese entsprechend der Kosten sequentiell abgearbeitet wird und eine Vorstellung als Liste naheliegt. In der Implementation wurde hingegen ein balancierter Suchbaum für die Kantensammlung eingesetzt, da veränderte Kanten neu sortiert in die Sammlung eingefügt werden müssen. Da bei Außerachtlassung imaginärer Kanten ein Knoten durchschnittlich 6 Netzkanten angehört (Erläuterung siehe Abschnitt 2.7.2), sind bei einer Verschmelzung um die 10 Kanten neu zu bewerten und einzusortieren. Eine Liste würde die Geschwindigkeit daher sehr negativ beeinflussen, da sie ein Einsortieren nur in $O(n)$ erlaubt, wenn n die Anzahl der echten und imaginären Kanten angibt, und damit die Abarbeitung der Kantensammlung nur in $O(n^2)$ möglich wäre. Der Suchbaum hingegen lässt ein Einsortieren in $O(\log n)$ zu, so dass die gesamte Abarbeitung in $O(n \log n)$ zu bewerkstelligen ist. Die Datenstruktur der Kantensammlung nutzt in der Implementation die Klasse `multimap` der C++-Standard-Musterklassen-Bibliothek.

A.3 Algorithmen

A.3.1 Abstand zwischen Punkt und Dreiecksnetz

Bei der Bewertung der Netzqualität wird zur Bestimmung der globalen Fehler der Abstand zwischen einem Punkt auf dem einen Netz zum gesamten anderen Netz benötigt. Sei nun M ein Netz und $\phi(M)$ dessen Oberfläche, so ist der Abstand eines Punktes p zu $\phi(M)$ gegeben durch die geringste Entfernung des Punktes zur Netzoberfläche

$$d_p(M) = \min_{w \in \phi(M)} \|p - w\|_2.$$

$\|p - w\|_2$ ist dabei die Euklidische Vektorlänge von $p - w$. Da das Netz aus Dreiecken zusammengesetzt ist, muss das Minimum der Abstände von p zu allen Dreiecken des Netzes bestimmt werden. Hierfür gibt es keine geschlossene Form.

Jeder Dreiecksabstand wird daher separat berechnet und anschließend das Minimum bestimmt. Intelligente Verfahren unterteilen jedoch den Raum, um die Abstandsberechnung nicht für alle Dreiecke durchführen zu müssen. Ein nicht entartetes Dreieck t sei durch seine Kanten e_i , $i \in \{1, 2, 3\}$ und Knoten v_i , $i \in \{1, 2, 3\}$ gegeben. Für die Abstandsbestimmung von einem beliebigen Punkt p zum Dreieck t gibt es drei Möglichkeiten:

1. ist p senkrecht auf t projizierbar, so ist die Distanz zwischen Projektionspunkt p_t und p der gesuchte Abstand, ansonsten:
2. ist p senkrecht auf mindestens eine Kante e_i von t projizierbar, so ist das Minimum der Distanzen zwischen den Projektionspunkten p_{e_i} und p der gesuchte Abstand, ansonsten:
3. der gesuchte Abstand ist das Minimum der Distanzen zwischen p und den Dreiecksknoten v_i von t

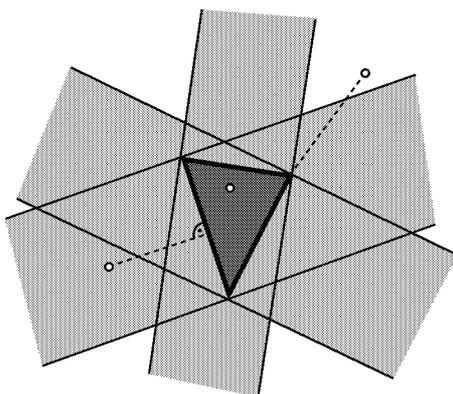


Abb. A.6: Raumaufteilung zur Abstandsbestimmung zwischen Punkt und Dreieck, der dunkle Bereich besitzt als nächsten Projektionspunkt einen Punkt auf dem Dreieck, der helle Bereich einen Punkt auf den Dreieckskanten und in den übrigen Teilen an den Dreiecksknoten

In Abbildung A.6 sind die drei möglichen Fälle der Abstandsvarianten im Zweidimensionalen skizziert. Für den dreidimensionalen Raum sind lediglich die Geraden und Kanten in die dritte Dimension zu Ebenen bzw. Ebenenstücken fortzusetzen. Für beliebige Punkte im Raum der unwahrscheinlichste Fall ist die Lage senkrecht ober- oder unterhalb des Dreiecks, also im dunkel markierten Bereich. Einen größeren Bereich deckt der zweite Fall ab, dessen Einzugsbereich hell markiert ist. Von diesen Raumabschnitten aus ist eine senkrechte Projektion auf die Dreieckskanten möglich. Die drei Raumabschnitte der Kanten schneiden sich in Regionen, von denen aus mehrere orthogonale Kantenprojektionen möglich sind. Den mit Abstand größten Bereich nimmt der dritte und letzte Fall ein. Dieser ist unmarkiert und erlaubt keine senkrechte Projektion auf die Kanten oder das Dreieck.

1. Fall:

Zu Beginn der Berechnung des Abstandes wird das Dreieck t und der Punkt p so verschoben, dass o.B.d.A. der Knoten v_3 im Nullpunkt liegt:

$$\begin{aligned} \underline{v}_i &= v_i - v_3, \quad i \in \{1, 2, 3\} \\ \underline{p} &= p - v_3. \end{aligned}$$

Die orthogonale Projektion \underline{p}_t von \underline{p} in die Ursprungsebene, die nun durch die Knoten \underline{v}_i aufgespannt wird, ist über eine eindeutige Linearkombination der Ortsvektoren der nicht im Nullpunkt liegenden Dreiecksnoten \underline{v}_1 und \underline{v}_2 zu erreichen und lautet mit den Faktoren

$$\alpha_1 = \frac{(v_1^t v_2)(\underline{p}^t v_2) - (v_2^t v_2)(\underline{p}^t v_1)}{(v_1^t v_2)^2 - (v_1^t v_1)(v_2^t v_2)} \quad \text{und} \quad \alpha_2 = \frac{(v_1^t v_2)(\underline{p}^t v_1) - (v_1^t v_1)(\underline{p}^t v_2)}{(v_1^t v_2)^2 - (v_1^t v_1)(v_2^t v_2)} ;$$

$$\underline{p}_t = \alpha_1 \underline{v}_1 + \alpha_2 \underline{v}_2.$$

Anhand der Wertigkeit von α_1 und α_2 ist nun die Lage von \underline{p}_t zu ermitteln. Da ein Dreieck die konvexe Hülle^G seiner Knoten darstellt, muss für die Lage von \underline{p}_t auf dem Dreieck einschließlich Rand

$$\alpha_1 + \alpha_2 \leq 1 \quad \wedge \quad \alpha_1, \alpha_2 \geq 0$$

gelten. Ist dies der Fall, lautet der minimale Abstand des Punktes p von Dreieck t

$$\|\underline{p} - \underline{p}_t\|_2.$$

2. Fall:

Für alle anderen Fälle muss das Minimum der Abstände zu den möglichen Kantenprojektionen bestimmt werden.

Die Orthogonalprojektion der Vektoren $p - v_i$, $i \in \{1, 2, 3\}$ auf die Richtungsvektoren der Kanten liefert die Faktoren

$$\alpha'_1 = \frac{(v_3 - v_2)^t (p - v_2)}{(v_3 - v_2)^t (v_3 - v_2)}, \quad \alpha'_2 = \frac{(v_1 - v_3)^t (p - v_3)}{(v_1 - v_3)^t (v_1 - v_3)}, \quad \alpha'_3 = \frac{(v_2 - v_1)^t (p - v_1)}{(v_2 - v_1)^t (v_2 - v_1)}.$$

Für

$$0 \leq \alpha'_i \leq 1, \quad i \in \{1, 2, 3\}$$

lauten die Kantenprojektionspunkte

$$p_{e_1} = \alpha'_1 (v_3 - v_2), \quad p_{e_2} = \alpha'_2 (v_1 - v_3), \quad p_{e_3} = \alpha'_3 (v_2 - v_1).$$

Wenn mindestens einer der Faktoren den Bedingungen genügt, dann ist der minimale Abstand des Punktes p von Dreieck t

$$\min_{i \in \{1, 2, 3\}} (\|p - p_{e_i}\|_2).$$

3. Fall:

Falls alle orthogonalen Projektionen nicht auf dem Dreieck oder seinen Kanten landen, ist der minimale Abstand durch den geringsten Abstand zu den Dreiecksnoten gegeben:

$$\min_{i \in \{1, 2, 3\}} (\|p - v_i\|_2).$$

A.4 Grafische Oberfläche

An dieser Stelle soll knapp die grafische Oberfläche der Anwendung SENSITUS (siehe Abschnitt A.1.3) mit den für die Visualisierung von Dreiecksnetzen und deren Vereinfachung wichtigen Fenstern beschrieben werden, da diese als Erweiterung der Anwendung im Rahmen der Arbeit implementiert wurden.

Abbildung A.7 zeigt folgende Fenster:

1. Konsole als hinterstes Bild zur Ausgabe von Informationen
2. Visualisierungsfenster mit einem Kuhmodell und einer Isofläche einer Elektronendichtewolke
3. Hauptfenster der Anwendung SENSITUS mit den zu visualisierenden Objekten im Szenengraph^G
4. Eigenschaftsfenster zur Einstellung von Parametern für die Oberfläche
5. Fenster mit Optionen für die Netzvereinfachung

Kurzbeschreibung der einzelnen Fenster:

1. *Konsole:*

Die Konsole dient der Ausgabe von Informationen während des Programmlaufs, zum Beispiel über den Fortschritt einer Vereinfachung.

2. *Visualisierungsfenster:*

Im Visualisierungsfenster werden die einzelnen geladenen Objekte dargestellt. In der Abbildung sind dies eine Kuh mit flacher Schattierung^G und einem Drahtmodell unter Angabe von Dreiecks- und Knotennummern, sowie einer Isofläche einer Elektronendichtewolke mit Gouraud-Schattierung^G.

3. *Hauptfenster:*

Das Hauptfenster der Anwendung SENSITUS verwaltet die geladenen Objekte und bindet sie in den Szenengraph^G ein. Das erste Symbol vor den Objektknoten gibt Auskunft über die Art der Objektdaten, die Weltkugel steht für den gesamten Szenengraph im globalen Weltkoordinatensystem, die 8 Dreiecke dienen als Symbol für ein Dreiecksnetz und der Würfel für Volumendaten^G. Das Auge neben den Symbolen zeigt an, dass das jeweilige Objekt angezeigt wird, und das Pfeilkreuz, dass das ausgewählte Objekt bei der Interaktion im Visualisierungsfenster manipuliert werden kann. Bei einem Doppelklick auf den Dateinamen des Objektes öffnet sich ein Fenster mit Eigenschaften des Knotens. Am unteren Rand des Hauptfensters ist die aktuelle Bild- und Kraftwiederholrate eingeblendet.

4. *Eigenschaftsfenster:*

Im Eigenschaftsfenster besteht die Möglichkeit, Parameter des Objektes einzustellen oder Informationen über das Objekt abzurufen. Im Info-Reiter sind die Ausdehnung des Originalmodells und der Vereinfachung, sowie Anzahl an Kanten und Dreiecken beider Modelle abzulesen. Des Weiteren wird die Anzahl der verschiedenen Kantentypen und der Zusammenhangskomponenten angegeben.

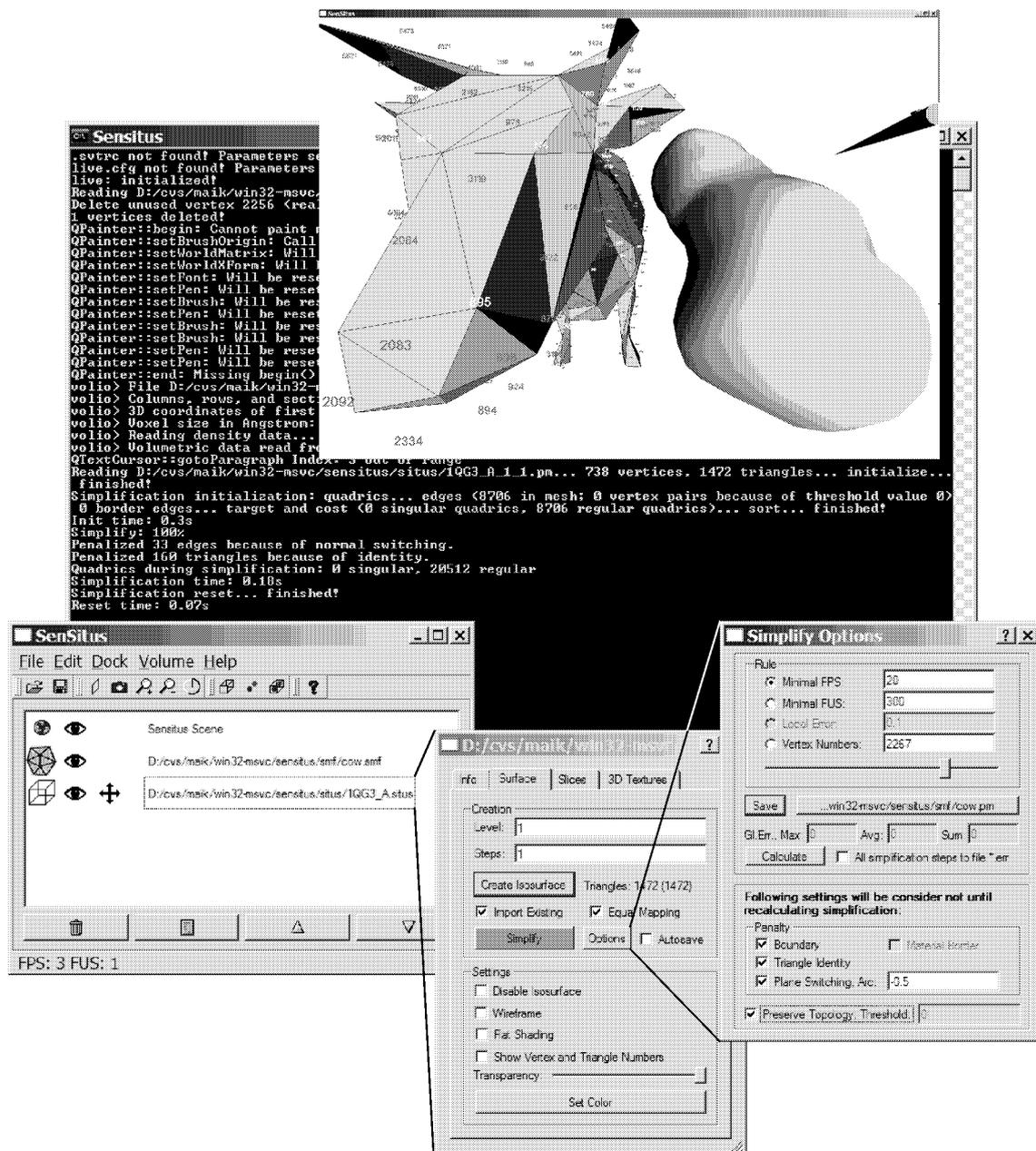


Abb. A.7: Grafische Oberfläche der Anwendung SENSITUS mit den für die Visualisierung wichtigen Fenstern

Der abgebildete Surface-Reiter stellt die Oberfläche des Objektes ein. Für den Volumendatensatz wird zur Erzeugung der Isofläche mittels des Marching-Cube-Algorithmus der Wert, der an der Isofläche im Datensatz herrschen soll, und die Schrittweite durch das Volumen, angegeben. Der Marching-Cube-Algorithmus erzeugt gemäß Abschnitt A.1.4 für jedes Voxel^GDreiecke, bringt diese aber in keinen Zusammenhang und erzeugt somit Knoten und Kanten doppelt. Damit identische Primitive auch im Netz durch nur ein Primitiv vertreten werden, kann eingestellt werden ("Equal Mapping"), dass dem Netz hinzugefügte Objekte überprüft werden, ob sie bereits im Netz vorliegen. Ist das Netz bereits einmal vereinfacht worden und wurde entsprechend der Einstellung für Isoflächenwert und Schritt-

weite dieses vereinfachte Netz gesichert, so kann ausgewählt werden (“Import Existing”), dass statt der Erzeugung eines neuen Netzes durch den Marching-Cube-Algorithmus ein entsprechendes bereits vereinfachtes Netz geladen wird. Ein letzter Auswahlschalter (“Autosave”) gibt an, ob berechnete Vereinfachungen automatisch neben dem Volumendatensatz zur Wiederverwendung auf der Platte abgelegt werden sollen. Der Simplify-Knopf berechnet eine Netzvereinfachung, ist er grün, liegt bereits eine Vereinfachung vor, ansonsten ist er rot. Im unteren Teil des Surface-Reiters sind Einstellungen für die Oberfläche zu machen, die das Aussehen verändern: Drahtmodell, Umschalten der standardmäßigen Gouraud- auf flache Schattierung, Anzeige von Knoten- und Dreiecksnummern, Transparenz und Farbe. Die Knoten- und Dreiecksnummern sind insbesondere bei der genauen Netzanalyse hilfreich.

5. Fenster der Vereinfachungsoptionen:

Durch Drücken des Option-Knopfes des Eigenschaftsfensters kommt ein Fenster zur Einstellung von Vereinfachungsoptionen hoch. Bei Vorliegen einer Vereinfachung kann über die Angabe der Regel für die adaptive Netzanpassung im oberen Bereich vorgegeben werden, ob sich die Detailstufe an die Bild- oder Kraftwiederholrate anpassen oder eine fest vorgegebene Knotennummer erfüllen soll (siehe Abschnitt 3.1.7).

Unter den Regeln besteht die Möglichkeit, einen Dateinamen für das vereinfachte Netz auszusuchen und dieses zu sichern (siehe Abschnitt 3.1.6).

Die globalen Fehler können für die aktuelle Detailstufe auf die Konsole ausgegeben oder die Fehler aller Stufen in eine Datei geschrieben werden (siehe Abschnitt 4.4).

Der untere Teil der Vereinfachungsoptionen ermöglicht Einstellungen künftiger Simplifizierungsläufe. Bestrafungen des Randes, der Dreiecksidentität und der Normalendrehung können wahlweise ein- oder ausgeschaltet werden. Für die Normalendrehung kann eingestellt werden, ab welchem Winkel in Form des Kosinus des von den Dreiecksnormalen eingeschlossenen Winkels eine Bestrafung erfolgen soll. Dieser Wert wird aber, wenn für die komplette Vereinfachung nötig, automatisch schrittweise hochgesetzt (siehe Abschnitt 3.1.5).

Als letzter Punkt kann die Hinzunahme von imaginären Kanten aktiviert werden. Der Schwellwert sollte aber mit Bedacht klein gewählt werden, da sonst die Anzahl imaginärer Kanten und damit die zusätzliche Rechenzeit drastisch ansteigt (siehe Abschnitt 3.1.2).



Beweise

B.1 Semidefinitheit der Untermatrix A des Quadrik-Fehlermaßes

Die Matrix A des Quadrik-Fehlermaßes ist positiv semidefinit. Hier soll diese Semidefinitheit anschaulich über die an der Matrix A beteiligten Matrizen gezeigt werden. Die Matrix A beinhaltet für das Quadrik-Fehlermaß alle Elemente der Quadrik, die zu einer Gesamtpotenz von zwei gehören. Ein Erzeugendensystem für reelle symmetrische 3×3 -Matrizen bildet zum Beispiel

$$\left\{ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\}.$$

Diese Matrizen sind durch folgende Ebenen

$$E_1 : x + y = 0, \quad E_2 : x + z = 0, \quad E_3 : y + z = 0, \quad E_4 : x = 0, \quad E_5 : y = 0, \quad E_6 : z = 0$$

als Quadratabstand zu erzeugen. Da diese Ebenen über einen Faktor zwar durch die Quadrierung positive Linearkombinationen der Matrizen des Erzeugendensystems erzeugen können, aber negative Matrixelemente nicht entstehen können, sind zur erschöpfenden Erzeugung aller symmetrischen Matrizen A , die bei dem Quadrik-Fehlermaß anfallen können, noch drei zusätzliche Matrizen für ein erweitertes Erzeugendensystem nötig:

$$\left\{ \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix} \right\}.$$

Die Matrizen sind Erzeugnisse aus den Quadratabständen der Ebenen

$$E_7 : x - y = 0, \quad E_8 : x - z = 0, \quad E_9 : y - z = 0.$$

Da die Hauptdiagonalelemente einer quadratischen Form wie hier nie negativ sind, sind Matrizen mit negativen Hauptdiagonalelementen nicht dem erweiterten Erzeugendensystem hinzuzufügen.

Da nun jede mögliche Matrix A einer Fehlerquadratik durch Linearkombination der Quadratabstände der Ebenen E_1 bis E_9 erzeugt werden kann, gilt wegen der Konstruktion über Quadrate, dass A positiv semidefinit ist, also

$$v^t A v \geq 0.$$



Beispiele

C.1 Minima lokaler Fehlermaße

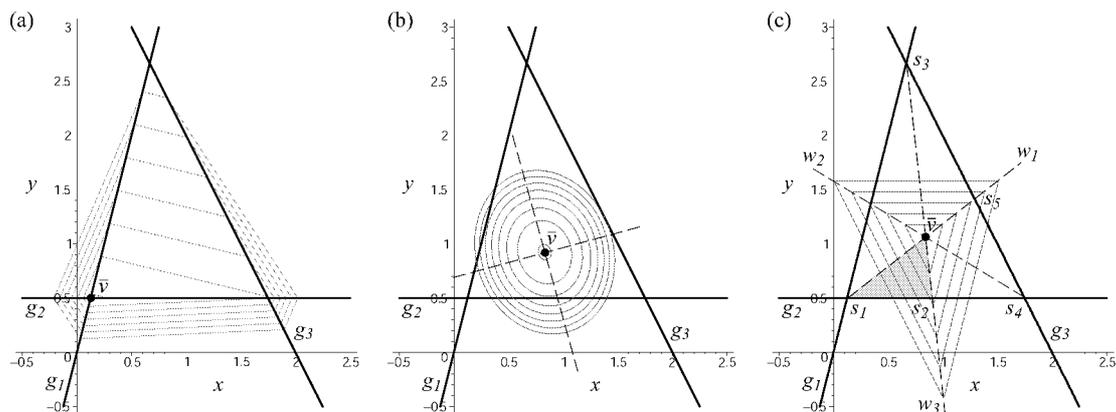


Abb. C.1: Geraden und Isolinien der lokalen Fehlermaße (a) E_{sum}^l , (b) E_{avg}^l und (c) E_{max}^l

Das Quadrik-Fehlermaß benutzt für die Bewertung einer Kontraktion die Summe der quadratischen Abstände des Kontraktionspunktes zu seinen korrespondierenden Flächen (siehe Abschnitt 3.1.3). Anhand eines Beispiels im Zweidimensionalen gemäß Abbildung C.1 soll eine Berechnung dieser Quadratsumme und der anderen zwei in Abschnitt 2.1.2 aufgeführten lokalen Fehlermaße neben der dortigen allgemeinen Analyse durchgeführt werden.

Gegeben sind folgende drei Geraden in der Hesseschen Normalenform:

$$g_1 : \frac{\sqrt{17}}{17}(4x_1 - x_2) = 0, \quad g_2 : x_2 - \frac{1}{2} = 0, \quad g_3 : \frac{\sqrt{5}}{5}(2x_1 + x_2 - 4) = 0,$$

mit $x = (x_1, x_2) \in \mathbb{R}^2$. Die Abstände eines Punktes x von den Geraden sind entsprechend:

$$D_1(x) = \left| \frac{\sqrt{17}}{17}(4x_1 - x_2) \right|, \quad D_2(x) = \left| x_2 - \frac{1}{2} \right|, \quad D_3(x) = \left| \frac{\sqrt{5}}{5}(2x_1 + x_2 - 4) \right|.$$

Die Geraden sind in sofern generell als dass sie nicht parallel zueinander sind und durch ihre drei Schnittpunkte ein Dreieck beschrieben wird, das nicht rechtwinklig ist und keine gleich langen Seiten aufweist.

Für Simplifizierungsverfahren ist lediglich der Winkel zwischen den Geraden ungewöhnlich groß, da zwei aufeinanderfolgende scharfe Ecken bzw. im Dreidimensionalen Kanten selten sind.

Abbildung C.1 zeigt die Geraden und Isolinien der hier zur Wiederholung aufgeführten lokalen Fehlermaße

$$\begin{aligned} \text{(a)} \quad E_{sum}^l(x, G) &= \sum_{i=1}^3 D_i(x), \\ \text{(b)} \quad E_{avg}^l(x, G) &= \sum_{i=1}^3 D_i^2(x), \\ \text{(c)} \quad E_{max}^l(x, G) &= \max_{i \in \mathbb{N}_3} D_i(x). \end{aligned}$$

$G = \{g_1, g_2, g_3\}$ ist dabei die Menge der Geraden.

Die Minima der Maße lauten:

(a) *lokales Summen-Fehlermaß:*

$$E_{sum}^l(x, G) = \left| \frac{\sqrt{17}}{17}(4x_1 - x_2) \right| + \left| x_2 - \frac{1}{2} \right| + \left| \frac{\sqrt{5}}{5}(2x_1 + x_2 - 4) \right|.$$

Dieses Maß ist nicht geschlossen zu differenzieren, so dass zur Eliminierung der Beträge Fallunterscheidungen nötig sind. Es gibt laut Zeichnung sieben außer auf den Rändern disjunkte Gebiete, die durch die Geraden begrenzt werden. Statt alle Fälle vorzurechnen, soll exemplarisch das Minimum im von den Geraden eingeschlossenen Dreieck bestimmt werden. Die Begrenzung erfolgt durch die Ungleichungen

$$\frac{\sqrt{17}}{17}(4x_1 - x_2) \geq 0 \quad \wedge \quad x_2 - \frac{1}{2} \geq 0 \quad \wedge \quad \frac{\sqrt{5}}{5}(2x_1 + x_2 - 4) \leq 0.$$

Die Beträge lösen sich somit wie folgt auf:

$$\begin{aligned} E_{sum}^l(x, G) &= \frac{\sqrt{17}}{17}(4x_1 - x_2) + x_2 - \frac{1}{2} - \frac{\sqrt{5}}{5}(2x_1 + x_2 - 4) \\ &= \underbrace{\frac{20\sqrt{17} - 34\sqrt{5}}{85}}_{>0} x_1 + \underbrace{\frac{85 - 5\sqrt{17} - 17\sqrt{5}}{85}}_{>0} x_2 + \frac{8\sqrt{5} - 5}{10}. \end{aligned}$$

E_{sum}^l besitzt also kein Minimum, da es streng monoton steigend in x_1 und x_2 ist. Da der Punkt

$$\bar{v} = \begin{pmatrix} 1/8 \\ 1/2 \end{pmatrix} = \begin{pmatrix} 0,125 \\ 0,5 \end{pmatrix}$$

auf dem Dreieck liegt und es in x_1 - und x_2 -Richtung keinen kleineren Wert auf dem Dreieck gibt, wird das Minimum am Punkt \bar{v} angenommen. Der minimale Wert lautet dort

$$E_{sum}^l(\bar{v}, G) = \frac{13\sqrt{5}}{20} \approx 1,453.$$

Diese Untersuchung müsste für alle restlichen sechs Fälle ebenso durchgeführt werden, wovon in drei Fällen an der gleichen Stelle das gleiche Minimum erzielt würde, das wegen seines kleinsten Gesamtwertes auch das Gesamtminimum darstellt.

(b) *lokales Quadratsummen-Fehlermaß oder auch Quadrik-Fehlermaß:*

$$\begin{aligned} E_{avg}^l(x, G) &= \left| \frac{\sqrt{17}}{17} (4x_1 - x_2) \right|^2 + \left| x_2 - \frac{1}{2} \right|^2 + \left| \frac{\sqrt{5}}{5} (2x_1 + x_2 - 4) \right|^2 \\ &= \frac{148}{85} x_1^2 + \frac{107}{85} x_2^2 - \frac{16}{5} x_1 - \frac{13}{5} x_2 + \frac{28}{85} x_1 x_2 + \frac{65}{4}. \end{aligned}$$

Die aufwendige Fallunterscheidung entfällt und die partiellen Ableitungen ergeben durch Nullsetzen die Stelle, an der das Minimum angenommen wird:

$$\begin{aligned} \frac{dE_{avg}^l(x, G)}{dx_1} &= \frac{296}{85} x_1 + \frac{28}{85} x_2 - \frac{16}{5} \stackrel{!}{=} 0 \\ \frac{dE_{avg}^l(x, G)}{dx_2} &= \frac{28}{85} x_1 + \frac{214}{85} x_2 - \frac{13}{5} \stackrel{!}{=} 0. \end{aligned}$$

Durch Lösen des Gleichungssystems ergibt sich der Punkt

$$\bar{v} = \begin{pmatrix} 153/184 \\ 85/92 \end{pmatrix} \approx \begin{pmatrix} 0,832 \\ 0,924 \end{pmatrix},$$

an dem das Minimum

$$E_{avg}^l(\bar{v}, G) = \frac{169}{184} \approx 0,918$$

angenommen wird. Es handelt sich dabei um das einzige Minimum, da E_{avg}^l für x_1 und x_2 streng monoton ist und für beide Unbekannte gegen Unendlich strebt:

$$\begin{aligned} \lim_{x_1 \rightarrow +\infty} E_{avg}^l((x_1, x_2)^t, G) &= \lim_{x_1 \rightarrow -\infty} E_{avg}^l((x_1, x_2)^t, G) = +\infty, \\ \lim_{x_2 \rightarrow +\infty} E_{avg}^l((x_1, x_2)^t, G) &= \lim_{x_2 \rightarrow -\infty} E_{avg}^l((x_1, x_2)^t, G) = +\infty. \end{aligned}$$

(c) *lokales Maximum-Fehlermaß:*

$$E_{max}^l(x, G) = \max\left(\left| \frac{\sqrt{17}}{17} (4x_1 - x_2) \right|, \left| x_2 - \frac{1}{2} \right|, \left| \frac{\sqrt{5}}{5} (2x_1 + x_2 - 4) \right|\right).$$

Für dieses Maß ist wie bei dem lokalen Summen-Maß wieder eine Fallunterscheidung für die Eliminierung der Beträge nötig. Innerhalb dieser Bereiche ist aber eine zusätzliche Fallunterscheidung bezüglich der maximalen Komponente nötig, was die Behandlung des Maximum-Maßes sehr aufwendig macht.

Wie beim Summen-Maß beschränken wir uns exemplarisch auf einen Fall, der das Minimum beinhaltet:

$$\begin{aligned} \frac{\sqrt{17}}{17} (4x_1 - x_2) \geq 0 \quad \wedge \quad x_2 - \frac{1}{2} \geq 0 \quad \wedge \quad \frac{\sqrt{5}}{5} (2x_1 + x_2 - 4) \leq 0 \\ \wedge \quad D_3 \geq D_1 \geq D_2. \end{aligned}$$

Dieser Fall beschreibt das Dreieck $t = \{s_1, \bar{v}, s_2\}$ mit s_1 als Schnittpunkt zwischen g_1 und g_2 und s_2 als Schnittpunkt zwischen g_2 und der Winkelhalbierenden des eingeschlossenen Winkels von g_1 und g_3 .

Durch die Ungleichungen ist nur die Komponente

$$E_{max}^l(x, G) = \frac{\sqrt{5}}{5}(2x_1 + x_2 - 4)$$

zu bewerten. Diese wird kleiner, wenn x_1 und x_2 steigen. Innerhalb des Dreiecks t ist somit auf dem Geradenstück $\{\bar{v}, s_1\}$ das Minimum zu suchen. Da E_{max}^l in beiden Komponenten streng monoton fallend ist, brauchen auf dem Geradenstück nur die Endpunkte untersucht werden.

Zur Bestimmung von \bar{v} und s_2 sind einige Berechnungen nötig. Die Schnittpunkte der Geraden lauten:

$$s_1 = \left(\frac{1}{8}, \frac{1}{2}\right)^t, \quad s_3 = \left(\frac{2}{3}, \frac{8}{3}\right)^t, \quad s_4 = \left(\frac{7}{4}, \frac{1}{2}\right)^t.$$

Die Schnittpunkte der Winkelhalbierenden mit den Geraden liegen zwischen den Schnittpunkten der Geraden:

$$s_2 = \frac{1}{2}(s_1 + s_4) = \left(\frac{15}{16}, \frac{1}{2}\right)^t, \quad s_5 = \frac{1}{2}(s_3 + s_4) = \left(\frac{29}{24}, \frac{19}{12}\right)^t.$$

Die Winkelhalbierenden w_1 und w_2 lauten nun

$$w_1 : x = s_1 + \lambda(s_5 - s_1), \quad \lambda \in \mathbb{R},$$

$$w_2 : x = s_3 + \mu(s_2 - s_3), \quad \mu \in \mathbb{R}.$$

Der Schnittpunkt der Winkelhalbierenden ist nach Gleichsetzen

$$\bar{v} = \begin{pmatrix} 2219/2568 \\ 349/321 \end{pmatrix} \approx \begin{pmatrix} 0,864 \\ 1,087 \end{pmatrix}.$$

Nun geben sich für die Endpunkte folgende Fehlerwerte:

$$E_{max}^l(\bar{v}, G) = \frac{507\sqrt{5}}{2140} \approx 0,530,$$

$$E_{max}^l(s_2, G) = \frac{13\sqrt{5}}{40} \approx 0,727.$$

Da $E_{max}^l(\bar{v}, G) < E_{max}^l(s_2, G)$ ist, wird an der Stelle \bar{v} das Minimum des lokalen Maximum-Fehlermaßes für dieses Gebiet angenommen.

Die gleiche Untersuchung müsste auch mit den restlichen 17 Fällen gemacht werden. In deren Bereichen wird E_{max}^l außer am Punkt \bar{v} größer als $E_{max}^l(\bar{v}, G)$ sein, so dass am Punkt \bar{v} das Minimum des lokalen Maximum-Maßes vorliegt.

Auch an dieser exemplarischen Berechnung der Minima verschiedener Fehlermaße ist zu erkennen, dass das Quadrik-Fehlermaß eine geschlossene kompakte Art der Minimumsberechnung erlaubt, wohingegen andere Maße aufwendige Fallunterscheidungen benötigen.

C.2 Konstruktion einer Ellipse als Summe der Quadratabstände zu zwei nicht parallelen Geraden

Es soll hier die in dieser Arbeit oft bemühte elliptische Isolinie als Summe der Quadratabstände von Geraden konstruiert werden. Die Ellipse ist gesucht, für die folgende Gleichung erfüllt ist:

$$D_1^2(x) + D_2^2(x) = c,$$

also die Summe der Quadratabstände identisch c ist. $c \in \mathbb{R}$ ist dabei beliebig aber fest. $D_1^2(x)$ ist der Abstand des Punktes $x \in \mathbb{R}^2$ von g_1 und $D_2^2(x)$ der Abstand von g_2 .

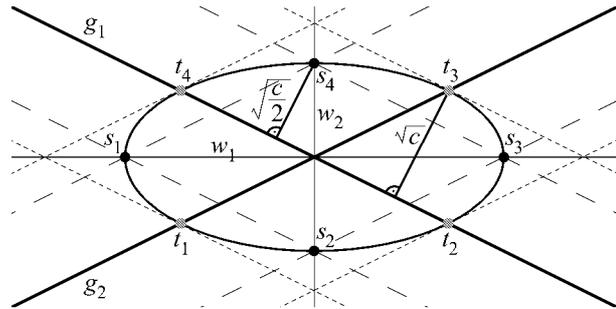


Abb. C.2: Beispiel der Konstruktion einer Ellipse als Summe der Quadratabstände zu den Geraden g_1 und g_2

Die Konstruktion läuft folgendermaßen ab. Für beide Geraden werden zwei parallele Geradenpaare eingezeichnet, die im Abstand von $\sqrt{\frac{c}{2}}$ und \sqrt{c} verlaufen. Die Geradenpaare beider Geraden im Abstand von $\sqrt{\frac{c}{2}}$ schneiden sich in den Scheitelpunkten s_1, s_2, s_3 und s_4 , da dort der Abstand von beiden Geraden gleich groß ist:

$$D_1^2(s_i) + D_2^2(s_i) = \left(\sqrt{\frac{c}{2}}\right)^2 + \left(\sqrt{\frac{c}{2}}\right)^2 = c \quad \forall i \in \mathbb{N}_4.$$

Durch die gegenüberliegenden Scheitelpunkte verlaufen die Hauptachsen w_1 und w_2 der Ellipse, die gleichzeitig die Winkelhalbierenden der von den Geraden eingeschlossenen Winkel sind. Durch die Scheitelpunkte ist die Ellipse festgeschrieben.

Zum besseren Zeichnen sind aber leicht vier weitere Punkte auf der Ellipse zu ermitteln. Die Geradenpaare im Abstand von \sqrt{c} stellen eine Begrenzung der Ellipse in dem Sinne dar, als dass die Ellipse diese Geraden nicht überschreiten kann, da auf den Geradenpaaren bereits der Abstand c als Quadrat von \sqrt{c} erreicht ist. Wo die Geradenpaare die jeweils andere Geraden g_1 oder g_2 schneiden, ist der Abstand von einer Geraden identisch Null. Da an diesen Punkten t_1, t_2, t_3 und t_4 der gewünschte Abstand

$$\begin{aligned} D_1^2(t_i) + D_2^2(t_i) &= (\sqrt{c})^2 + 0^2 = c, \quad i \in \{1, 3\} \\ D_1^2(t_i) + D_2^2(t_i) &= 0^2 + (\sqrt{c})^2 = c, \quad i \in \{2, 4\} \end{aligned}$$

erreicht wird, sind $t_i, i \in \mathbb{N}_4$ weitere vier Punkte auf der Ellipse. An den Punkten t_i stellen die Geradenpaare Tangenten an die Ellipse dar, da sie sie dort berühren und die Ellipse die Geradenpaare nicht überschreiten darf.

Diese Konstruktionsmethode lässt sich nicht auf mehrere Geraden erweitern.

C.3 Berechnung des Quadrik-Fehlermaßes

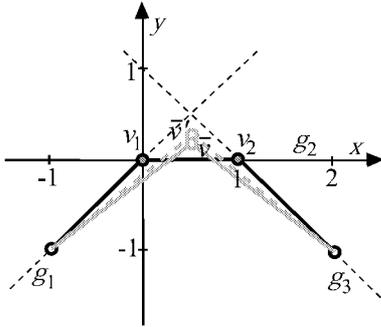


Abb. C.3: Beispiel zur Kostenbestimmung im Zweidimensionalen über die Summe der Quadratabstände und über das Quadrik-Fehlermaß

Anhand eines simplen zweidimensionalen Beispiels entsprechend Abbildung C.3, soll hier die Berechnung des Quadrik-Fehlermaßes für eine Kantenkontraktion und die Ermittlung des optimalen Verschmelzungspunktes durchgeführt werden.

Die normierten Geradennormalenformen lauten

$$g_1: \frac{\sqrt{2}}{2} \begin{pmatrix} 1 \\ -1 \end{pmatrix}^t \begin{pmatrix} x \\ y \end{pmatrix} = 0,$$

$$g_2: \begin{pmatrix} 0 \\ 1 \end{pmatrix}^t \begin{pmatrix} x \\ y \end{pmatrix} = 0 \quad (\text{x-Achse}),$$

$$g_3: \frac{\sqrt{2}}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix}^t \begin{pmatrix} x \\ y \end{pmatrix} - \frac{\sqrt{2}}{2} = 0.$$

Die Quadratabstände eines Punktes $\tilde{v} = (x, y, 1)^t$ zu den Geraden sind

$$D_1^2(\tilde{v}) = \tilde{v}^t \begin{pmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \\ 0 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \end{pmatrix} \tilde{v} = \tilde{v}^t \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \tilde{v},$$

$$D_2^2(\tilde{v}) = \tilde{v}^t \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \tilde{v} = \tilde{v}^t \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tilde{v},$$

$$D_3^2(\tilde{v}) = \tilde{v}^t \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{pmatrix} \tilde{v} = \tilde{v}^t \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \tilde{v}.$$

Daraus ergeben sich die Quadriken an den Knoten v_1 und v_2 gemäß Gleichung (3.4)

$$Q_1 = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{3}{2} & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$Q_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{3}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

Für die auf Basis des Quadrik-Fehlermaßes zu bewertende Kantenkontraktion $(v_1, v_2) \rightarrow \bar{v}$ ist

$$Q(\bar{v}) = \bar{v}^t (Q_1 + Q_2) \bar{v} = \bar{v}^t \begin{pmatrix} 1 & 0 & -\frac{1}{2} \\ 0 & 3 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \bar{v}$$

zu minimieren. Hierzu ist entsprechend Gleichung (3.7) folgendes Gleichungssystem zu lösen:

$$\begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix} \bar{v} + \begin{pmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} = 0$$

$$\Leftrightarrow \bar{v} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{6} \end{pmatrix}$$

Die durch den Verschmelzungspunkt \bar{v} erzeugte Vereinfachung ist in Abbildung C.3 durch die hellen durchgezogenen Linien gekennzeichnet. Durch die Doppelbewertung der Geraden g_2 über beide Quadriken Q_1 und Q_2 rückt der Verschmelzungspunkt näher an g_2 heran als bei der einfachen Summe der Quadratabstände

$$\sum_{i=1}^3 D_i^2(\tilde{v}) = \tilde{v}^t \begin{pmatrix} 1 & 0 & -\frac{1}{2} \\ 0 & 2 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \tilde{v},$$

mit dem Minimum bei

$$\bar{v}' = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{4} \end{pmatrix}.$$

Dieses Minimum würde in Abbildung C.3 die hellen gestrichelten Linien als Vereinfachung zur Folge haben.

C.4 Anzahl von zusätzlichen Knotenpaaren aufgrund eines Schwellwertes

In Abschnitt 3.1.2 wird von einem quadratischen bis kubischen Knotenpaaranstieg gesprochen, je nachdem wie die Knoten angeordnet sind. Als Beispiel soll hier für ein flächiges Objekt, also ein Objekt mit quadratischem Knotenpaaranstieg, ein äquidistantes Punktegitter im Zweidimensionalen untersucht werden, das entsprechend Abbildung C.4 an jeder ganzzahligen Position einen Knoten besitzt.

Die Anzahl der Knotenpaare zu einem Knoten im Ursprung steigt nun folgendermaßen. Alle Knoten $(x, y) \in \mathbb{N}^2$, $|x| \leq r, |y| \leq r$ liegen in der durch den Schwellwert $r \in \mathbb{R}_0^+$ gegebenen Kreisscheibe mit dem gleichen Radius, für die $\sqrt{x^2 + y^2} \leq r \Leftrightarrow |y| \leq \sqrt{r^2 - x^2}$ gilt. Die abgeschlossene Kreisscheibe (“ \leq ”) hier unterscheidet sich zur offenen Kreisscheibe im eigentlichen Verfahren (“ $<$ ”), was die Berechnung etwas vereinfacht, die Größenordnung des Ergebnisses aber beibehält. Somit liegen

$$c = \sum_{x=-\lceil r \rceil}^{\lfloor r \rfloor} (2\lfloor \sqrt{r^2 - x^2} \rfloor + 1)$$

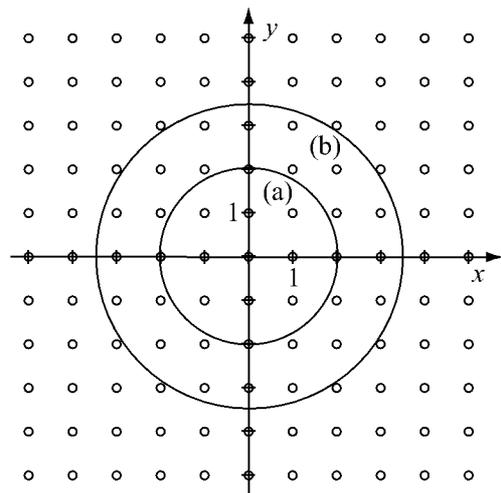


Abb. C.4: Netzbeispiel mit Knoten auf einem ganzzahligen äquidistanten Gitter zur Bestimmung der Knotenpaaranzahl in Abhängigkeit des Schwellwertes. Kreisscheibe (a) umfasst $\sum_{x=-\lfloor 2 \rfloor}^{\lfloor 2 \rfloor} (2\lfloor \sqrt{2^2 - x^2} \rfloor + 1) - 1 = 12$ Knotenpaare und (b) $\sum_{x=-\lfloor 3,5 \rfloor}^{\lfloor 3,5 \rfloor} (3,5\lfloor \sqrt{3,5^2 - x^2} \rfloor + 1) - 1 = 36$ Knotenpaare, die vom Ursprung ausgehen.

Knoten in der Kreisscheibe. $c - 1$ Knotenpaare wären aufgrund des Schwellwertes pro Knoten die Folge. Eine obere Schranke

$$c - 1 < \sum_{x=\lceil -r \rceil}^{\lfloor r \rfloor} (2\sqrt{r^2} + 1) = (2\lfloor r \rfloor + 1)(2\lceil r \rceil + 1) \leq (2r + 1)^2$$

und eine untere Schranke

$$\begin{aligned} c - 1 &\geq \sum_{x=\lceil -r \rceil}^{\lfloor r \rfloor} (\sqrt{r^2 - x^2} - 1) - 1 \\ &\geq \sum_{x=\lceil -r \rceil}^{\lfloor r \rfloor} ((r - x) - 1) - 1 && \text{(C.1)} \\ &= (2\lfloor r \rfloor + 1)(r - 1) - 1 \geq 2r^2 - 3r \end{aligned}$$

zeigen, dass die Anzahl der Knoten um einen Knoten bei Erhöhung des Schwellwertes r quadratisch ansteigt. Die Ungleichung (C.1) gilt, da für $|x| \leq r$

$$r - x \leq \sqrt{r^2 - x^2} \Leftrightarrow (r - x)^2 \leq r^2 - x^2 \Leftrightarrow x^2 \leq rx$$

korrekt ist.

Wird das Knotengitter ins Dreidimensionale fortgesetzt, ergibt sich für die Anzahl der Knotenpaare pro Knoten entsprechend

$$\sum_{x=\lceil -r \rceil}^{\lfloor r \rfloor} \sum_{y=\lceil -\sqrt{r^2 - x^2} \rceil}^{\lfloor \sqrt{r^2 - x^2} \rfloor} (2\lfloor \sqrt{r^2 - y^2} \rfloor + 1) - 1$$

und der Anstieg ist damit kubisch in Bezug auf den Schwellwert r .



Glossar

Konvexe Menge Eine Teilmenge wird konvex genannt, wenn sie alle Punkte auf der Verbindungsstrecke zwischen zwei Punkten dieser Menge enthält.

Konvexe Hülle Die konvexe Hülle einer Teilmenge ist die kleinste konvexe Menge, die die Teilmenge beinhaltet.

Polygon Ein Polygon ist ein geschlossenes ebenes Vieleck mit geradlinigen Begrenzungsstrecken als Seiten. Diese eingeschränkte Definition wird oft auch als einfaches, ebenes, geschlossenes Polygon bezeichnet.

Polygonnetz Das Polygonnetz ist eine Flächenbeschreibung durch eine Menge von Polygonen, die sich nur in ihren Eckpunkten oder Kanten schneiden dürfen. Bei einem Dreiecksnetz sind keine Polygone mit mehr als drei Eckpunkten zulässig.

Polyeder Ein Polyeder ist ein zusammenhängendes Objekt, dessen Oberfläche durch ein Polygonnetz beschrieben wird. Jede Kante im Netz gehört dabei zu genau zwei Polygonen.

Volumendaten Mit Volumendaten bezeichnet man eine Menge von skalaren Daten, welche an diskreten Punkten im Raum definiert sind. Sie werden auch 3D-Skalarfelder genannt. Die Daten entstehen vornehmlich bei Messungen auf einem regulären Gitter (zum Beispiel Computertomografie-Aufnahmen) und teilen dabei den Raum in Zellen ein. Diese Zellen werden bei Volumendaten aber auch bei anderen raumaufteilenden Strukturen, äquivalent zu Pixeln in 2D-Bildern, **Voxel** genannt.

Quadtree Der Quadtree ist eine Datenstruktur, die ein Rechteck rekursiv in vier Quadranten einteilt. Die einzelnen Quadranten werden dabei fortlaufend in vier gleich große Gebiete zerlegt, wobei die Häufigkeit der rekursiven Zerlegung je nach lokalen Gegebenheiten schwanken darf.

Octree Der Octree ist eine Datenstruktur als dreidimensionale Verallgemeinerung des Quadtree durch rekursive Zerlegung eines Würfels in acht gleich große Oktanten.

Triangulation Die Triangulation erstellt aus einer Punktwolke oder einem Polygonnetz ein Dreiecksnetz. Die Dreiecksseiten dürfen dabei nur von Knoten zu Knoten gehen und sich dabei nicht schneiden.

Voronoi-Diagramm Eine **Voronoi-Region** eines Elementes einer Menge ist das Gebiet, das keinem Element der Menge näher ist als dem Element selbst. Elemente können dabei neben Punkten zum Beispiel auch Kantenstücke oder Dreiecke sein. Im Zweidimensionalen sind **Voronoi-Kanten** die Gebiete, die zu zwei Elementen gleich weit entfernt liegen, und **Voronoi-Knoten** die Gebiete, die über mindestens drei gleich weit entfernte Elemente verfügen. Im Dreidimensionalen sind für Voronoi-Kanten genau drei und für Voronoi-Knoten mindestens vier Elemente nötig, die die gleiche Entfernung voneinander haben. Die **Voronoi-Ebenen** entsprechen im Dreidimensionalen den Voronoi-Kanten im Zweidimensionalen mit genau zwei Elementen, deren Distanz zu dieser Voronoi-Ebene identisch ist. Alle Gebiete, die nicht genau ein nächstes Element haben, bilden zusammen das Voronoi-Diagramm [6].

Delaunay-Triangulation Die Delaunay-Triangulation ist eine Triangulation einer Punktmenge, deren Dreiecksseiten genau die Punkte verbinden, die einen Kreis im Zweidimensionalen bzw. Kugel im Dreidimensionalen besitzen, die nur die jeweiligen Kantenendpunkte beinhalten. Dies entspricht Kanten, die im Zweidimensionalen genau dort zwischen zwei Punkten gezogen werden, die eine gemeinsame Voronoi-Kante besitzen, und im Dreidimensionalen dort, wo eine gemeinsame Voronoi-Ebene vorliegt. Die Delaunay-Triangulation ist damit eine zum Voronoi-Diagramm duale Struktur [6].

Visualisierungspipeline Der mehrstufige Prozess für die Realisierung von dreidimensionalen Darstellungen wird Visualisierungspipeline, Grafik-Pipeline, Bildgenerierungspipeline oder Rendering-Pipeline genannt.

Beleuchtungsmodell Beleuchtungsmodelle beschreiben die Zusammenhänge zwischen Raumgeometrie, Lichtquellen und Oberflächenbeschaffenheit zur Berechnung der Leuchtdichte in einem Punkt der darzustellenden Szene. Es wird zwischen lokalen und globalen Beleuchtungsverfahren unterschieden. Lokale Beleuchtungsmodelle erfassen nur direkte Beleuchtung von Lichtquellen und simulieren die globale Beleuchtung durch einen konstanten ambienten Lichtanteil. Sie sind für die schnelle Visualisierung gedacht und berechnen die Beleuchtung meist nur an wenigen diskreten Netzpunkten. Die Beleuchtung der Flächen wird dann über die Schattierung erzeugt. Im Gegensatz zu den lokalen Beleuchtungsverfahren steht bei den globalen Verfahren eine möglichst realistische Wiedergabe einer Szene im Vordergrund. Dies wird durch möglichst exakte Modellierung von physikalischen Vorgängen erreicht. Die bekanntesten Verfahren hierfür sind das Raytracing- und Radiosity-Verfahren.

Schattierung Schattierung wird in lokalen Beleuchtungsmodellen die Berechnung der Lichtausbreitung auf der Oberfläche genannt. Die bekanntesten Verfahren sind die **konstante** oder **flache Schattierung** (flat shading), die **interpolierende Schattierung** gemäß **Gouraud** (Gouraud-Shading) und gemäß **Phong** (Phong-Shading). Die konstante Schattierung berechnet nach einem lokalen Beleuchtungsmodell genau eine Leuchtdichte für eine Fläche und schattiert die gesamte Fläche mit dem identischen Wert. Die Gouraud-Schattierung berechnet die Leuchtdichte an Netzknoten und interpoliert die Dichte im ersten Schritt entlang der Netzkanten und im zweiten Schritt entlang der Rasterzeilen in der Fläche von Kante zu Kante. Die Berechnung ist langsamer als die konstante, aber schneller als die Phong-Schattierung. Das visuelle Ergebnis ist gleichmäßig diffus. Die Phong-Schattierung erfasst besser spiegelnde Effekte, da

die Normalenvektoren statt der Leuchtdichte interpoliert und zur Berechnung der Leuchtdichte an jedem Punkt herangezogen werden.

Textur Texturen sind Funktionen, die Punkte einer Textur-Ebene im Zweidimensionalen bzw. eines Textur-Raumes im Dreidimensionalen auf Farben oder Grauwerte abbilden.

Szenengraph Ein Szenengraph ist eine hierarchische Beschreibung einer dreidimensionalen Szene. Komplexe Objekte werden mit Hilfe von Kindknoten in einem Graphen beschrieben. Komplexere Objekte können so aus einfacheren Objekten zusammengesetzt werden.

Virtual Reality-System Ein Virtual Reality-System bietet eine vom Computer simulierte Wirklichkeit bzw. künstliche Welt, in die man sich mit Hilfe der entsprechenden technischen Ausrüstung scheinbar hineinversetzen kann.

Kraftrückkopplung Die Kraftrückkopplung dient insbesondere in einem Virtual Reality-System dazu, einen **taktilen Sinneseindruck** zu vermitteln. Der Vorgang wird **haptisches Rendern** genannt und kann aktiv Kraft weitergeben oder passiv Kraft spüren lassen.

Tiefenpuffer-Algorithmus Der Tiefenpuffer-Algorithmus oder **z-Buffer-Algorithmus** (z buffering) dient zur Ermittlung sichtbarer Flächen mit Bildgenauigkeit. Hierzu wird ein z-Buffer verwaltet, der für jedes Pixel den z-Wert des in diesem Punkt dem Betrachter am nächsten liegenden Objekts enthält.

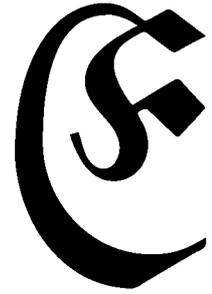
Rendern Als Rendern bezeichnet man den Vorgang der Überführung eines Szenegraphen virtueller Objekte in ein gerastertes meist zweidimensionales Bild. Diesen Vorgang nennt man auch Bildgenerierung oder Bilderzeugung durch Rasterisierung.

Display-Liste Die Display-Liste oder auch Darstellungsliste ist eine von OpenGL unterstützte Datenstruktur, die das Zeichnen von einmal bestimmten grafischen Objekten bei wiederkehrender Darstellung beschleunigt, für ihre Berechnung aber zusätzlich Zeit beansprucht.

NP-hart P ist die Menge aller Probleme, die man mit deterministischen Algorithmen in Polynomialzeit lösen kann, NP-Probleme mit nichtdeterministischen Algorithmen. P ist eine Teilmenge von NP. Es gibt eine Reihe von Problemen in NP, von denen nicht bekannt ist, ob sie auch in P liegen. Einige von diesen Problemen hängen in dem folgenden Sinn miteinander zusammen: wenn man auch nur von einem dieser Probleme zeigen könnte, dass es in P liegt, wären die anderen auch alle in P. Probleme mit dieser Eigenschaft bezeichnet man als NP-hart.

Wavelet Wavelets sind Funktionen, die sowohl im Ortsraum als auch im Wellenzahlraum nur in einem endlichen Intervall signifikant von Null abweichen und gegen Unendlich rasch gegen Null streben. Der Mittelwert über ihren gesamten Definitionsbereich ist Null.

Wavelet-Repräsentation Die Wavelet-Repräsentation bietet in der Multiskalen-Analyse eine hierarchische Zerlegung einer Funktion in eine Basis aus Wavelet- und Skalenfunktionen. Dabei werden Wavelets verwendet, deren Skalierung von einer Grundskalierung bis hin zu einer beliebigen kleinen Skalierung reicht, während die Skalenfunktionen die Beiträge der Wavelets oberhalb der Grundskalierung zusammenfassen.



Danksagung

Die vorliegende Arbeit wurde als Diplomarbeit am Lehrstuhl für Numerischen Mathematik der Fern-Universität Hagen erstellt.

An dieser Stelle möchte ich all jenen danken, die durch ihre fachliche und persönliche Unterstützung zum Gelingen dieser Arbeit beigetragen haben.

Besonderer Dank gebührt Herrn Prof. Dr. Franz Locher und Herrn Dr. Herwig Zilken, die mit viel Interesse diese Arbeit betreut haben. Zudem möchte ich mich bei Herrn Prof. Dr. Friedel Hoßfeld für die Übernahme des Zweitgutachtens und die technischen Möglichkeiten, die ich am Zentralinstitut für Angewandte Mathematik des Forschungszentrums Jülich nutzen durfte, bedanken.

Für das Korrekturlesen ist Herrn Dr. Rainer von Seggern, Herrn Dr. Bernd Mohr und Frau Tatjana Eitrich herzlich zu danken.

Weiter bedanken möchte ich mich bei Herrn Dr. Stefan Birmanns für die Schaffung der hervorragenden programmiertechnischen Basis, auf der die Implementation des Verfahrens aufsetzt.

Zu guter Letzt möchte ich meiner Familie danken, die in der Zeit der Verfassung der Diplomarbeit oft auf ihren Ehemann bzw. Vater verzichten musste.

Forschungszentrum Jülich
in der Helmholtz-Gemeinschaft



Jül-4157
Januar 2005
ISSN 0944-2952