

**Studiengang:** Informatik  
**Prüfer:** Prof. Dr. Thomas Ertl  
**Betreuer:** Dipl. Phys. Martin Kraus

**begonnen am:** 12. November 2001

**beendet am:** 12. Mai 2002

**CR-Klassifikation:** I.3.5, J.3, E.1, G.2.2

Studienarbeit Nr. 1830

## **Visualisierung und gezielte Selektion von Isoflächenkomponenten**

Markus Merz

Abteilung für Visualisierung und interaktive Systeme  
Universität Stuttgart  
Breitwiesenstr. 20-22  
D-70565 Stuttgart



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Gliederung . . . . .	3
<b>2</b>	<b>Volumen-Visualisierung</b>	<b>5</b>
2.1	Isoflächen . . . . .	5
2.1.1	Isoflächen aus Konturen . . . . .	5
2.1.2	Opaque Cube/Cuberille . . . . .	6
2.1.3	Marching Cubes, MC . . . . .	6
2.1.4	Marching Tetrahedra, MT . . . . .	7
2.2	Direkte Volumenvisualisierung . . . . .	7
2.2.1	Raycasting . . . . .	7
2.2.2	Projektion . . . . .	8
2.3	Topologie . . . . .	10
<b>3</b>	<b>Der Konturbaum</b>	<b>13</b>
3.1	Beispiele . . . . .	13
3.2	Vorbetrachtungen . . . . .	14
3.3	Datenformat der Volumendaten . . . . .	15
3.4	Datenaufbereitung . . . . .	16
3.5	Virtuelle Tetraederzerlegung . . . . .	16
3.6	Algorithmen . . . . .	17
3.6.1	Baumstruktur . . . . .	17
3.6.2	Durchlaufen des Volumens . . . . .	18
3.6.3	Identifikation des Vertexyps . . . . .	20
3.6.4	Behandlung der Vertexypen . . . . .	21
3.6.5	Vereinfachung des Baumes . . . . .	22
3.6.6	Satzzellenbestimmung . . . . .	23
3.6.7	Berechnung der Volumen . . . . .	24
3.7	Speichern des Konturbaums/Datenformat . . . . .	25
<b>4</b>	<b>Visualisierung des Konturbaums</b>	<b>29</b>
4.1	Laden des Konturbaums und der Daten . . . . .	29
4.2	Visualisierung . . . . .	29
4.2.1	Visualisierung mit Knoten und Kanten . . . . .	30
4.2.2	Visualisierung mit Volumendaten . . . . .	32
4.3	Picking und Isowertselektion . . . . .	33
4.4	Weitere Interaktionsmöglichkeiten . . . . .	34

<b>5</b>	<b>Visualisierung der Isoflächenkomponenten</b>	<b>35</b>
5.1	Saatzellenidentifizierung . . . . .	35
5.2	Marching Tetrahedra . . . . .	35
5.2.1	Schnittpunktinterpolation . . . . .	36
5.2.2	Zellbehandlung . . . . .	36
5.3	Interaktionsmöglichkeiten . . . . .	37
<b>6</b>	<b>Beispiele</b>	<b>39</b>
6.1	Testdatensatz . . . . .	39
6.2	Koordinatenachsen . . . . .	41
6.3	Kopfdatensatz . . . . .	43
<b>7</b>	<b>Erweiterungen und Anwendungen</b>	<b>47</b>
7.1	Erweiterungen . . . . .	47
7.2	Anwendungen . . . . .	50
<b>8</b>	<b>Zusammenfassung</b>	<b>55</b>
<b>A</b>	<b>Programmstruktur</b>	<b>57</b>
A.1	Hauptfenster . . . . .	57
A.1.1	Isowertselektion / Picking Dialog . . . . .	59
A.1.2	Konturbaum Dialog . . . . .	60
A.1.3	Interaktionsmöglichkeiten - Konturbaum . . . . .	60
A.2	Darstellung der Isoflächen . . . . .	61
A.3	Icons . . . . .	62

# 1 Einleitung

## 1.1 Motivation

Das Ziel dieser Studienarbeit ist die Implementierung eines Verfahrens zur Visualisierung von Isoflächenkomponenten und zur indirekten Visualisierung von Volumendaten über die Darstellung eines Konturbaums (Contour Tree).

Ein Konturbaum repräsentiert (im Sinne der Morse Theorie) die Topologie eines Skalarfeldes. Ein kritischer Punkt des Skalarfeldes (Extremum oder Sattelpunkt) wird durch einen Knoten im Konturbaum wiedergespiegelt. Eine Kante entspricht einer zusammenhängenden „Volumenkomponente“. Im Detail repräsentiert eine Kante eine Menge von topologisch gleichen Variationen einer Komponente von Isoflächen zu verschiedenen Isowerten. Die Komponente wird begrenzt durch die Skalarwerte der kritischen Punkte, die an die Kante anschließen.

Zu Beginn erfolgt ein kurzer Einblick in das Gebiet der Volumenvisualisierung, wobei mehrere direkte und indirekte Verfahren kurz angesprochen werden. Ausgehend von einer Einarbeitung in das Themengebiet Konturbaum mit zugehöriger Literaturrecherche wird eine Berechnung des Konturbaums eines 3D Skalarfeldes, gegeben als kartesisches Gitter, durchgeführt. Dabei wird eine Variation des Konturbaums mit zusätzlichen regulären Knoten bestimmt. Diese Variante wird für die folgende Visualisierung dieses Baumes benötigt. Über den Baum ist eine gezielte Selektion eines Isowertes in einer bestimmten Kontur (entspricht einem Teilbaum) möglich. Die zu diesem Isowert und zur Kontur gehörende Isoflächenkomponente kann anschließend über ein einfaches Marching Tetrahedra Verfahren visualisiert werden. Die erzielten Ergebnisse werden durch die Darstellung einiger Beispiele dokumentiert. Zusätzlich werden noch einige Erweiterungsmöglichkeiten und Anwendungen des Konturbaums betrachtet.

## 1.2 Gliederung

Im Folgenden wird kurz die Gliederung dieser Studienarbeit erläutert. Kapitel 1 gibt in der Motivation einen kurzen Überblick über die Aufgabenstellung und die Gliederung der Ausarbeitung. In Kapitel 2 erfolgt eine kurze Einführung in das Gebiet der Volumenvisualisierung. Es werden unter anderem Methoden zur direkten und indirekten Volumenvisualisierung vorgestellt. Der Aufbau des Konturbaums wird komplett in Kapitel 3 behandelt. Dies beinhaltet unter anderem die Vorbereitung der Daten, die Berechnung des Baums und aller notwendigen zusätzlichen Informationen zur Visualisierung, sowie seine Speicherung. In Kapitel 4 folgt anschließend die Beschreibung des Ladens und der Visualisierung des Konturbaums mit einer Vorstellung der Interaktionsmöglichkeiten und in Kapitel 5 werden die Isoflächenkomponenten behandelt (Visualisierung und Interaktion). Danach folgen einige Beschreibungen und Bilder von behandelten Beispieldatensätzen. In Kapitel 7 werden Erweiterungsmöglichkeiten der Arbeit und einige Anwendungen des

Konturbaums angesprochen.

Nach der anschließenden Zusammenfassung folgt ein Anhang zur Beschreibung der Oberfläche und der Bedienung des Programms. Das Abbildungsverzeichnis und die Literaturangaben bilden das Ende dieser Studienarbeit.

## 2 Volumen-Visualisierung

Die Hauptaufgaben der Volumenvisualisierung sind vor allem die Repräsentation, Manipulation und Darstellung von Volumendaten. Anwendungsgebiete für Skalar­daten sind zum Beispiel Bereiche der medizinischen Visualisierung oder die Darstellung von geographischen Daten. Isoflächen/-linien kommen dabei unter anderem bei der Visualisierung von Landschaften (Höhenfelder) oder in der Medizin (Körperteile) zum Einsatz.

Für die Volumenvisualisierung von 3D Skalar­daten bieten sich zwei Sichtweisen, die oft ungenau unterschieden werden:

- 3D-Zellen:  
Werte an Gitterpunkten, dazwischen wird interpoliert
- 3D-Voxel:  
Werte in Gitterzellen (= Voxel) konstant.

Die Verfahren zur Visualisierung können in drei Hauptklassen unterteilt werden.

- Slicing:  
Interpolation von Daten auf 2D Schnittflächen und Verwendung von 2D Skalarfeld Verfahren (Höhenlinien, Rasterbilder, Höhenfelder).
- Isoflächen:  
Hierbei handelt es sich um eine indirekte Technik der Volumenvisualisierung. Die Volumendaten werden auf eine Flächenbeschreibung reduziert. Die Daten werden dann als Fläche gerendert.
- Direkte Volumenvisualisierung (DVR):  
Es erfolgt eine direkte „ganzheitliche“ Darstellung des Volumens ohne Zwischenflächen. Es ist jedoch eine semitransparente Darstellung erforderlich.

### 2.1 Isoflächen

#### 2.1.1 Isoflächen aus Konturen

Zuerst wird in 2D Schnittflächen nach geschlossenen Konturen gesucht. Anschließend verbindet man die Konturen in den 2D Schnitten und zeichnet die sich ergebenden Dreiecke [11]. Abbildung 1 zeigt drei 2D Schnitte mit Konturen und die zu zeichnenden Dreiecke. Jedoch entstehen Schwierigkeiten bei einer großen Anzahl an Konturen pro Schnittfläche und bei starken Variationen zwischen den Schichten.

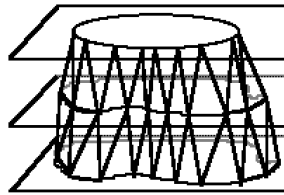


Abbildung 1: Contour Tracking

### 2.1.2 Opaque Cube/Cuberille

Durch eine Binärisierung des Volumens können alle Randflächen gefunden werden. In jedem Voxel, der zum Rand gehört, wird eine Fläche primitiv plaziert. Die schattierten Flächen werden mittels z-Buffer gerendert. Dies ergibt jedoch ein würfeliges Aussehen mit wenig erkennbaren Details (Abbildung 2) [12].

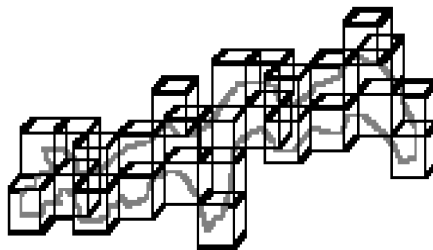


Abbildung 2: Opaque Cube/Cuberille

### 2.1.3 Marching Cubes, MC

Beim Marching Cubes Verfahren von Lorensen und Cline [14] wird das Volumen entlang von Isoflächen durchwandert. Es erfolgt eine Klassifizierung der Knoten (als innerhalb oder außerhalb, wenn Datenwert größer oder kleiner als Isowert) aus der wiederum eine Unterscheidung der einzelnen Fälle für den Voxel erfolgt. Abbildung 3 zeigt alle Fälle (unter Berücksichtigung der Symmetrie reduziert von 256 auf 15) die beim MC auftreten können. Aufgrund der Vielzahl von unterschiedlichen Fällen ist eine Lookup Table sinnvoll. Die Schnittpunkte der Isofläche werden linear auf den Kanten interpoliert. Zusätzlich kann noch die Normale der entstehenden Fläche berechnet werden. Zur Steigerung der Effizienz können bei richtiger Marschrichtung die Ergebnisse der zu letzt behandelten Zelle verwendet werden. Trotz der Fallunterscheidungen können aufgrund von Zweideutigkeiten Löcher in den Flächen entstehen. Dies erfordert eine zusätzliche Behandlung, zum Beispiel durch den Asymptotic Decider von Nielson [15].

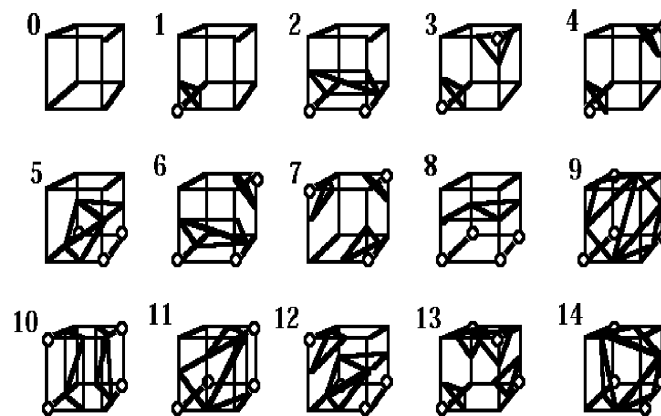


Abbildung 3: 15 Fälle des Marching Cubes

Zur Optimierung des MC wurden mehrere Verbesserungen entwickelt, von denen hier zwei kurz vorgestellt werden.

Beim Discretized Marching Cubes [16] wird der Kantenschnittpunkt nicht durch Interpolation berechnet, sondern es wird einfach der Mittelpunkt verwendet. Es entstehen nur 13 verschiedene Vertizes (12 Kanten und Mittelpunkt). Dadurch ergeben sich keine kleinen Dreiecke und es wird eine effiziente Kodierung ermöglicht. Zusätzlich entstehen keine Zweideutigkeiten. Zur Beschleunigung der Zellsuche können Octrees oder Intervallverfahren zur schnelleren Durchquerung von Bereichen verwendet werden, die den gesuchten Isowert nicht enthalten.

#### 2.1.4 Marching Tetrahedra, MT

Marching Tetrahedra entspricht dem MC auf Tetraedergittern [17]. Zunächst ist bei unstrukturierten Gittern eine Tetraedrisierung notwendig. Dieses Verfahren kommt hier bei der Visualisierung der Isoflächen zur Anwendung und wird in einem späteren Abschnitt erläutert.

## 2.2 Direkte Volumenvisualisierung

Für die direkte Volumenvisualisierung werden kurz einige Verfahren aus den Bereichen Raycasting und Projektion vorgestellt.

### 2.2.1 Raycasting

Hierbei handelt es sich um ein Verfahren im Bildraum. Für jeden zu berechnenden Pixel wird ein Strahl in das Volumen geschickt. Jedes getroffene Objekt wird in die Berechnung

der Pixelwerte miteinbezogen. Farbe und Transparenz der Objekte, sowie ihre gegenseitige Verdeckung müssen dabei berücksichtigt werden.

### Einfache Raycasting Verfahren

Es stehen mehrere einfache Verfahren zur Verfügung, die bei einem Auftreffen auf bestimmte Datenwerte (Maximum, Schwellwert, Mittelwert) eine Zuordnung auf Farbe und Transparenz durchführen. Diese sind unter anderem Maximum Intensity Projection, Integral Intensity Projection (entspricht Röntgen) und das Schwellwert-Verfahren. Schattierung mit oder ohne Berücksichtigung der Normalen liefert die Intensität, Farben können über Transferfunktionen bestimmt werden. Die Komposition semitransparenter Voxel kann mit dem Over-Operator ( $C_{out} = C_{in} * (1 - \alpha) + C * \alpha$ ,  $C \equiv$  Farbe in RGB-Werten,  $\alpha \equiv$  Opazität) erfolgen.

Bei beliebig orientierten Volumen oder achsenparallelen Volumen mit Zentralperspektive wird trilineare Interpolation für die Berechnung der Werte an den Sampling Punkten entlang eines Strahls benötigt (Abbildung 4). Ein möglicher Abstand zwischen den Sampling Punkten ist die halbe Voxelbreite.

Eine Verbesserung für den Punktabstand bietet das Adaptive Ray Sampling von Hanrahan und Danskin [18]. Dabei werden uninteressante Datenbereiche schneller durchquert (Bereiche mit geringer Opazität, kleinem Gradienten).

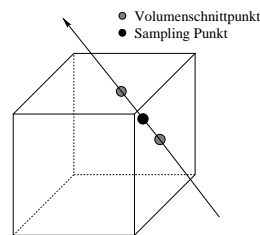


Abbildung 4: Raycasting bei beliebig orientierten Volumen

Template Based Volume Viewing [19] nutzt die Kohärenz von parallelen Strahlen aus. Bei unterschiedlichen Startpunkten wird dasselbe Schnittmuster verwendet. Space-Leaping [20] versucht den leeren (transparenten) Raum möglichst schnell zu traversieren. Dabei können unterschiedliche Varianten zur hierarchischen Speicherung des leeren oder belegten Raumes angewandt werden (Octrees, proximity clouds).

### 2.2.2 Projektion

Dies ist ein Verfahren im Objektraum. Für jeden Voxel im Volumen werden die Pixel bestimmt, auf die er projiziert wird. Auch bei diesem Verfahren müssen Farbe, Transparenz und Verdeckungen beachtet werden. Projektion bietet einige Vorteile gegenüber Raycasting. Diese sind unter anderem eine leichtere Parallelisierbarkeit, eventuell ein besseres Antialiasing und eine bessere Verteilbarkeit der Volumendaten auf Distributed Memory.

### V-Buffer

Es erfolgt die Berechnung einer Bounding Box für jede Datenzelle in der Bildebene. Für jede darin enthaltene Scanline wird die zugehörige Schnittebene bestimmt. Anschließend wird die Ebene in höchstens fünf spans zerlegt (siehe Abbildung 5). Die Pixelwerte können dann mit Gauß-Quadratur mit trilinearärer Interpolation berechnet werden. [21]

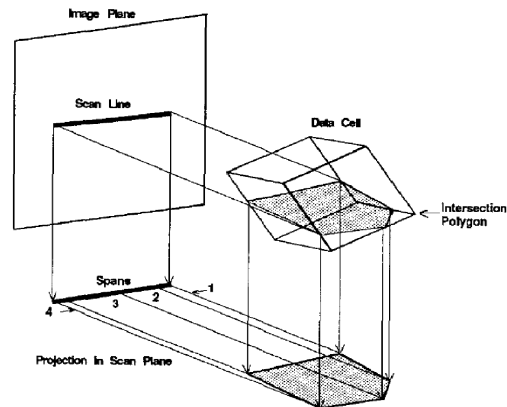


Abbildung 5: V-Buffer: Unterteilung in spans

### Splatting

Der Beitrag eines Voxels zum Bild wird über mehrere Pixel verteilt. Für jeden Voxel wird in einer Tabelle ein Footprint gespeichert. Das Bild entsteht durch Compositing der transformierten Footprints. [22]

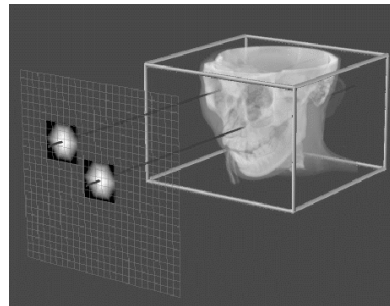


Abbildung 6: Splatting

### Volumenrotation

Das Ziel ist die Transformation eines beliebig orientierten Volumens in das Bildkoordinatensystem. Ein Beispiel für diese Vorgehensweise ist der Shear-Warp von Lacronite/Levoy [24]. Zunächst erfolgt eine Scherung des Volumens. Bei perspektivischer Projektion kann eine Skalierung notwendig sein. Anschließend ergibt Front-to-Back Compositing ein Zwischenbild im Sheared-Object-Space. Als letzter Schritt folgt aus einem Image Warp das Endbild. Abbildung 7 verdeutlicht die Vorgehensweise.

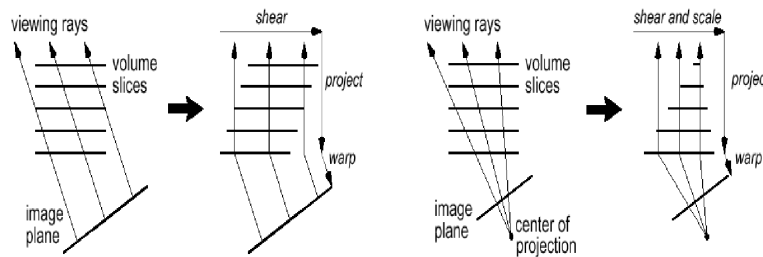


Abbildung 7: Shear Warp für parallele und perspektivische Projektion

### 3D Texture Mapping

Die Visualisierung erfolgt durch Verwendung von OpenGL Texture Mapping und Compositing Hardware [25]. Der Texturspeicher wird als 3D Volumen betrachtet. Es erfolgt eine Berechnung der Schnittpolygone mit dem Texturwürfel. Die Polygone werden Back-to-Front mit Blending gezeichnet. Der Abstand der Polygone entspricht ungefähr dem Abstand der Voxel. Zur Beschleunigung können größere Datensätze in Bricks unterteilt und nach einer Sortierung separat gerendert werden.

## 2.3 Topologie

Aufgrund der schnell wachsenden Größe von Volumendatensätzen treten bei deren Visualisierung häufig Probleme zum Beispiel bei der Auswahl von Transferfunktionen oder der Suche nach dem richtigen Isowert für Flächen auf. Deshalb werden bei der Visualisierung von Volumendaten Informationen über die Topologie der Daten zur schnelleren oder gezielteren Darstellung von Datenbereichen eingesetzt. Im Folgenden werden kurz einige Bereiche erläutert.

### Morse-Theorie

Als hauptsächliche Grundlage für die Entwicklung der Verfahren zur Bestimmung des Reeb-Graphen oder des Konturbaums dient die Morse-Theorie. Deshalb werden hier einige kurze Informationen darüber gegeben. Eine ausführliche Beschreibung kann jedoch wegen ihres Umfangs nicht erfolgen. Diese kann unter anderem in [5] nachgelesen werden.

Die Morse-Theorie untersucht die Beziehung zwischen kritischen Punkten von Funktionen und der Topologie von Mannigfaltigkeiten. Hier sind besonders die Veränderungen in der Topologie von Interesse. Es ist erforderlich, dass kritische Punkte an Vertices des Gitters auftreten. Die restlichen Punkte des Gitters sind reguläre Punkte. Eine Funktion die diese Bedingungen erfüllt ist eine Morse-Funktion. Werte zwischen Vertices werden interpoliert.

### Reeb-Graph

Der Reeb Graph wird in 2D zur Darstellung von geographischen Eigenschaften, Ober-

flächendarstellungen und Isoflächen verwendet. Er repräsentiert die kritischen Punkte der Höhenfunktion einer Fläche. An diesen Punkten (Peak, Pit, Pass) entstehen, enden, gabeln oder vereinigen sich Konturen. Sie stellen die Knoten des Reeb Graphen dar. Abbildung 8 zeigt ein Ellipsoid (a) und einen Torus (b) und die zugehörigen Reeb-Graphen.

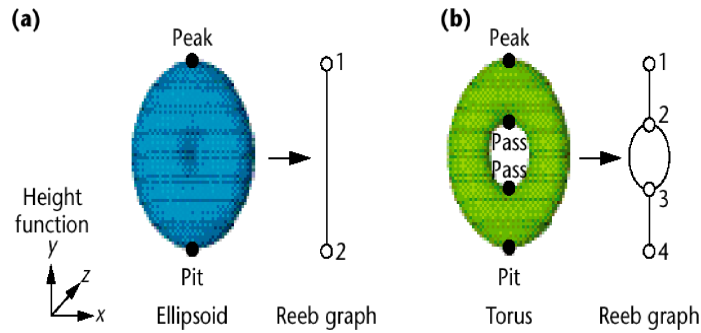


Abbildung 8: Ellipsoid/Torus und zugehörige Reeb Graphen

Für 3D wurde, wie in [6] erwähnt, der Hyper Reeb Graph entwickelt. Er stellt eine Erweiterung des Reeb Graphen für Volumendaten dar. Ein Volumen stellt eine unendliche Anzahl von Isoflächen dar. Durch Verfolgen dieser Flächen wird der Graph aufgebaut. Dessen Knoten sind wieder die kritischen Punkte mit ihren Funktionswerten. Diese Knoten bilden einen linearen gerichteten Graphen. Die Kanten zwischen den Knoten enthalten den zugehörigen Reeb Graphen und die Differenz der Knotenwerte. Abbildung 9 zeigt ein Beispiel für einen Metatorus: Knoten  $v_i$  mit Werten  $f_i$ , Kanten  $e_{i,i+1}$  mit den Knotenwertdifferenzen  $l_{i,i+1}$ .

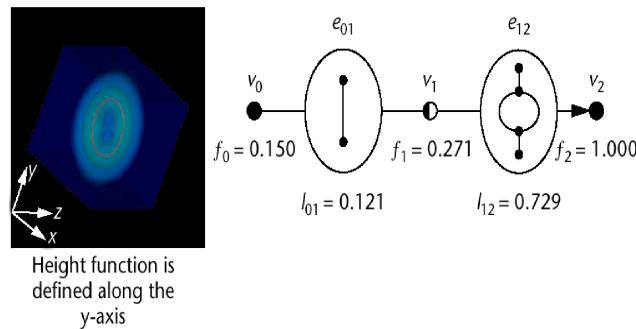


Abbildung 9: Metatorus und zugehöriger Hyper Reeb Graph

### Konturbaum

Eine weitere Möglichkeit zur Darstellung der Topologie von Skalarwerten ist der Konturbaum. Dieser ist verwandt mit dem Reeb Graphen und basiert ebenfalls teilweise auf der Morse Theorie.

Einige grundlegende Bedingungen, die für den Konturbaum angewandt werden, wurden bereits in [2] bewiesen. Diese sind unter anderem: Die ungerichtete Version des aufgebauten Graphen ist für die hier betrachteten Skalarfelder ein Baum, Konturen zerlegen

das Gitter in Regionen, genau zwei kritische Punkte begrenzen eine Region bzw. zwei Konturen begrenzen eine Region.

Das Verfahren von Kreveld [1] benötigt ebenfalls Daten an den Gitterpunkten, Werte dazwischen werden interpoliert. Es wird ausführlich für 2D Daten erklärt, kann aber auch auf 3D angewandt werden. Der Konturbaum wird ebenfalls aus den kritischen Punkten des Volumens aufgebaut. Die Volumenwerte werden von  $+\infty$  nach  $-\infty$  durchwandert. Dabei werden die Veränderungen der Konturen an den kritischen Punkten verfolgt. In 2D wird dieses Verfahren in  $n \log(n)$  Schritten ( $n$  entspricht der Anzahl der Gitterpunkte) durchgeführt, in 3D nur  $n^2$ . Es wird leicht verändert hier angewandt und wird ausführlich im nächsten Abschnitt beschrieben.

In [4] wird der Algorithmus von Kreveld mit einer leichten Veränderung auf 3D angewandt bzw. verbessert und so ein Verfahren, das nur  $n \log(n)$  Schritte benötigt, entwickelt. In drei Durchläufen erfolgt die Erkennung der Vereinigungen, der Gabelungen und die Kombination beider Ergebnisse. Eine weitere Verbesserung der Methode von Kreveld erfolgte von Carr [3]. Ausgehend von Tarasov wurde der Algorithmus auf  $n$  Dimensionen ausgedehnt.

## 3 Der Konturbaum

### 3.1 Beispiele

In diesem Abschnitt soll durch je ein 2D und ein 3D Beispiel ein praktischer Einblick in den Themenbereich Konturbaum gegeben werden. Als Vorlage dient dabei vor allem [1].

Abbildung 10 zeigt ein Beispiel für einen Konturbaum entstanden aus einem 2D Datensatz. Zur Vereinfachung haben alle Vertizes unterschiedliche Funktionswerte. Alle kritischen Punkte der Funktion sind an Vertizes. Alle Konturen, die durch Vertizes gehen, zerlegen die Fläche in Regionen. Konturen durch Minima oder Maxima entsprechen einfach dem Punkt. Jede Region ist begrenzt durch genau zwei Konturen. Jede Kontur repräsentiert einen Knoten im Graphen. Zwischen zwei Knoten im Graph entsteht eine Kante, wenn sie durch eine Region miteinander verbunden sind. Abbildung 10 zeigt links das Datengitter (Konturen gepunktet), rechts der zugehörige Graph.

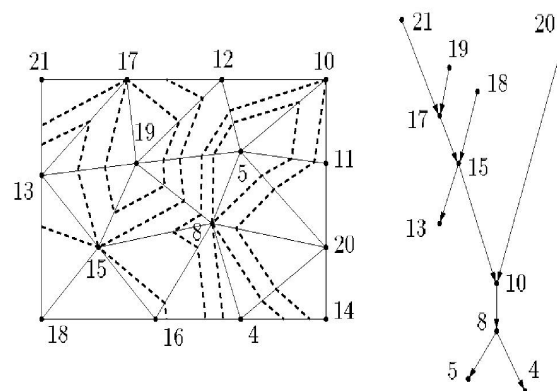


Abbildung 10: Beispiel Konturbaum, 2D

Die Abbildungen 11 und 12 illustrieren ein Beispiel für 3D. Abbildung 11 zeigt entstandene Isoflächen beim Durchlaufen des Volumens von  $+\infty$  nach  $-\infty$ . Zuerst entstehen vier neue Konturen (a), dann vereinigen sich die ersten beiden (b) und anschließend die beiden nächsten Konturen (c). Die entstandenen Konturen vereinigen sich zu einer einzigen (d), die sich wiederum in zwei neue aufteilt (e). Eine Kontur repräsentiert die inneren Flächen, eine die Äußeren. Die innere Kontur verschwindet später, die Äußere geht in das globale Minimum über (f). Abbildung 12 zeigt den zugehörigen Konturbaum. Die Knoten 7, 8, 9 und 10 repräsentieren die zuerst entstandenen Konturen (a), 4, 5 und 6 die darauf folgenden Vereinigungen (b,c,d). In Knoten 3 spaltet sich dann die Kontur wieder auf in die Knoten 1 und 2.

Bei der Bestimmung des Konturbaums ist es auch möglich reguläre Vertizes einer Kontur als Knoten in den Baum mit aufzunehmen. Dabei liegen diese Punkte zwischen den

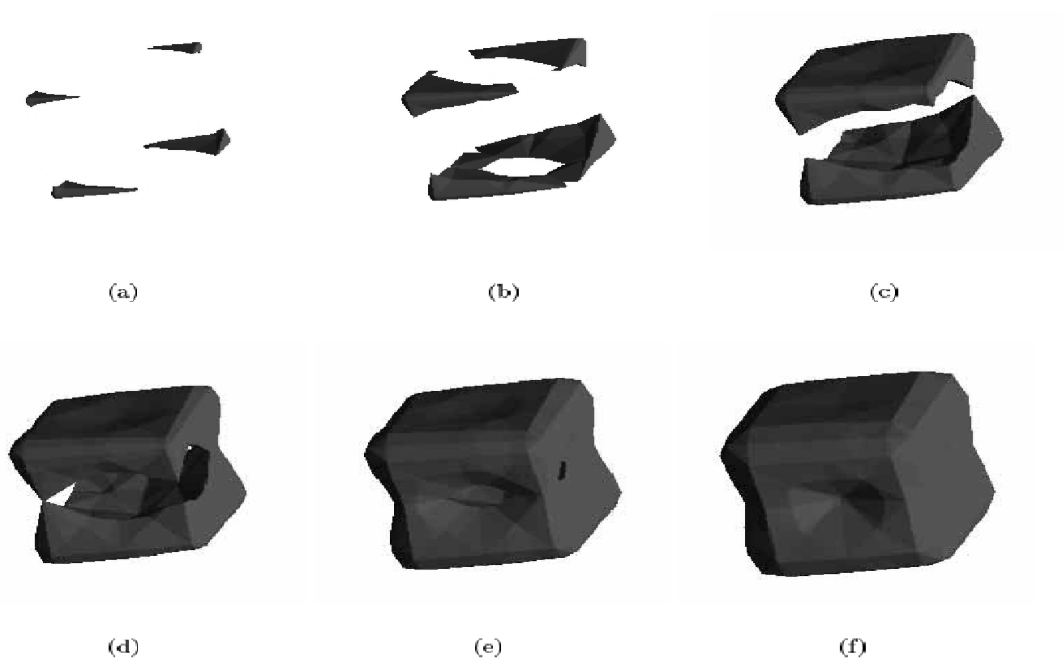


Abbildung 11: Beispiel Konturbaum, 3D, Flächen

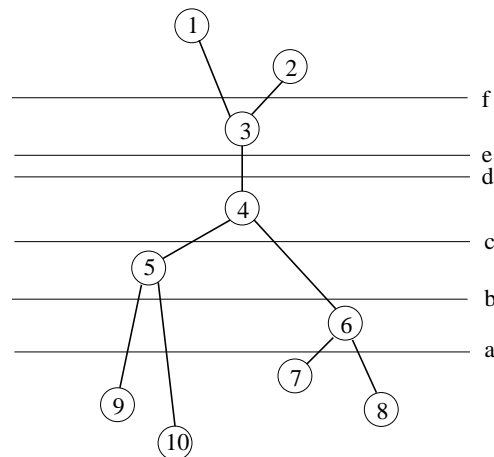


Abbildung 12: Beispiel Konturbaum, 3D, Baum

entsprechenden, die Kontur begrenzenden, kritischen Punkten. Abbildung 13 zeigt den Konturbaum aus Abbildung 12 mit zusätzlich eingefügten regulären Knoten.

### 3.2 Vorbetrachtungen

Für die folgende Bestimmung des Konturbaums müssen einige Bedingungen erfüllt sein. Über dem Datengitter muss eine Funktion für alle Zellen definiert sein. Alle kritischen

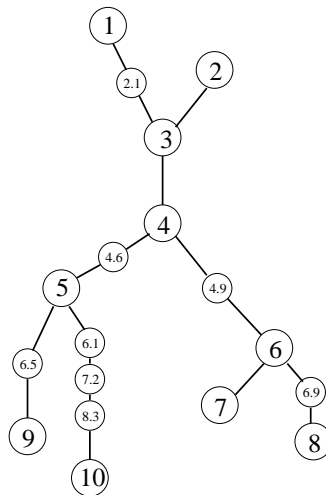


Abbildung 13: Beispiel Konturbaum, 3D, Baum mit regulären Knoten

Punkte müssen an Gitterpunkten liegen und ihr Funktionswert muss bestimmbar sein. Die Definition des Konturbaums kann folgendermaßen betrachtet werden: Finde jede Kontur, die einen kritischen Punkt enthält. Jede dieser Konturen entspricht einem Knoten (Minimum, Maximum oder Sattel) des Konturbaums. Für jede Region, die von zwei Konturen begrenzt wird, wird eine Kante in den Baum zwischen den entsprechenden Knoten eingefügt.

### 3.3 Datenformat der Volumendaten

Die Skalarwerte des 3D Feldes sind auf einem kartesischen Gitter gegeben. Der zu ladende Datensatz wird über eine \*.dat Datei ausgewählt. Diese enthält die wichtigsten Informationen über den Datensatz im ASCII Format. Die Daten selbst sind in einer \*.raw Datei gespeichert.

Die folgende Tabelle zeigt den Aufbau der \*.dat Datei.

Zeile	Beschreibung
ObjectName: *.raw	Dateiname der *.raw Datei
Resolution: x y z	Auflösung des Datensatzes in x-, y-, z- Richtung
SliceThickness: x y z	Abstände der Gitterpunkte in x-, y-, z- Richtung
Format: f	Format der Skalarwerte in der .raw Datei, UCHAR = unsigned char, USHORT = unsigned short, FLOAT = float.

Nach dem Einlesen der \*.dat Datei kann das Anlegen der notwendigen Datenfelder erfolgen. Anschließend werden die Skalarwerte aus der \*.raw Datei eingelesen. Diese sind binär gespeichert und werden je nach Datentyp in unsigned char konvertiert.

### 3.4 Datenaufbereitung

Zur Vermeidung gleicher Datenwerte an benachbarten Vertizes muss das Datenfeld direkt nach dem Laden überarbeitet werden. Hierzu werden immer an jeweils acht Vertizes, die einen Würfel bilden, Werte hinzuaddiert. Da jeder Vertex nur einmal behandelt wird, nur ganze Zahlen an den Vertizes möglich sind und durch die Anordnung der Additionswerte (siehe Abbildung 14) gewährleistet ist, dass zu keinem Nachbarvertex derselbe Wert addiert wird, können an benachbarten Vertizes keine gleichen Werte mehr auftreten.

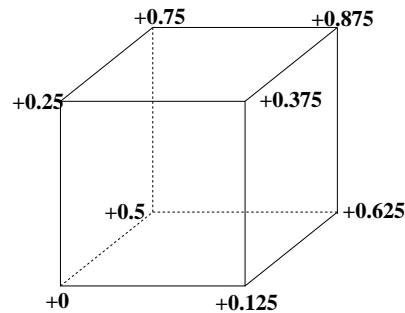


Abbildung 14: Additionswerte an den Vertizes

Für das Durchlaufen des Volumens ist eine Sortierung der Vertizes nach Datenwerten notwendig. Hierfür wird ein simpler Quicksort benutzt. Die sortierten Vertexindizes werden in einem Feld gespeichert. Das Datenfeld wird dabei nicht verändert.

Zu Beginn werden mehrere Datenfelder angelegt, die zur Durchführung des Algorithmus benötigt werden.

Diese weiteren Felder sind:

- Status der Vertizes:  
Speichert ob ein Vertex bereits erfolgreich abgearbeitet wurde oder nicht.
- Status der Tetraederzellen beim Konturbaumaufbau:  
Speichert ob eine Zelle aktiv, bearbeitet oder noch überhaupt nicht betrachtet wurde.
- Status der Tetraederzellen bei der Volumenberechnung:  
Speichert ob eine Zelle beim Marching Tetraedra zur Volumenberechnung bereits bearbeitet wurde.

### 3.5 Virtuelle Tetraederzerlegung

Die Skalarwerte sind auf einem kartesischen Gitter gegeben. Die Berechnung des Konturbaums wird auf einem Tetraedergitter durchgeführt, um lineare Interpolation in den Zellen zu ermöglichen und dadurch kritische Punkte nur an Vertizes auftreten zu lassen. Deshalb

muss das vorhandene Gitter entsprechend zerlegt werden. Diese Zerlegung wird jedoch nicht gespeichert, sondern erfolgt nur virtuell, das heißt sie wird während der Bearbeitung (*on-the-fly*) durchgeführt. Die Berechnung der Tetraedervertizes erfolgt aus den Koordinaten der Würfecken. Für die Tetraedisierung einer Würfelzelle werden hauptsächlich Varianten mit fünf oder sechs Tetraeder verwendet. Um möglichst wenige Tetraeder zu erhalten wird eine Zerlegung in fünf benutzt (siehe Abbildung 15). Dabei muss jedoch bei Nachbarzellen die Ausrichtung der Zerlegung beachtet werden um Kantenüberschneidungen zu verhindern.

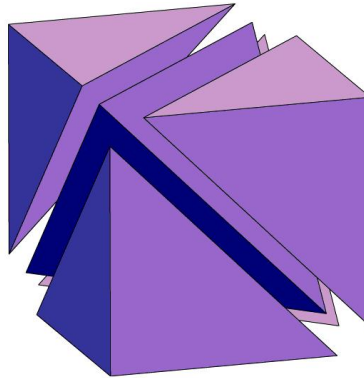


Abbildung 15: Zerlegung eines Würfels in fünf Tetraeder

Für die Vertizes entstehen zwei unterschiedliche Nachbarschaftsverhältnisse mit je sechs bzw. achtzehn Nachbarvertizes.

## 3.6 Algorithmen

### 3.6.1 Baumstruktur

Der Aufbau und die Speicherung des Konturbaums erfolgt über eine dafür entworfene Datenstruktur. Für jeden Knotentyp, Minimum, Maximum, Vereinigung, Gabelung oder regulärer Knoten, kann ein entsprechendes Objekt erzeugt werden. Jedes dieser Objekte speichert den Typ des Knotens, einen eindeutigen Index und eine Konturnummer. Je nach Typ können noch weitere Eigenschaften gespeichert werden, darunter Zeiger auf die jeweiligen Vorgänger und Nachfolger im Baum:

- Minimum:  
Zeiger auf den Nachfolger.
- Maximum:  
Zeiger auf den Vorgänger.
- regulärer Knoten:  
Zeiger auf Vorgänger und Nachfolger.

- **Vereinigung:**  
Zeiger auf den linken und den rechten Vorgänger. Speicherung der zugehörigen linken und rechten Konturnummer. Zeiger auf den Nachfolger. Für dessen Konturnummer ist bereits Platz vorhanden.
- **Gabelung:**  
Zeiger auf den linken und den rechten Nachfolger. Speicherung der zugehörigen linken und rechten Konturnummer. Zeiger auf den Vorgänger. Für dessen Konturnummer ist bereits Platz vorhanden.

Reguläre Knoten haben also Eingangs-/Ausgangsgrad 1/1, Maxima die Grade 1/0, Minima 0/1, Gabelungen 1/2 und Vereinigungen die Grade 2/1. Sattelpunkte mit höheren Eingangs- bzw. Ausgangsgraden können durch Kombination von Knoten erreicht werden (siehe Seite 22). Entsprechend der Grade wird eine Anzahl an Gabelungen und Vereinigungen erzeugt und aneinander angehängt.

Beim Eintragen von Knoten 1 als Vorgänger von Knoten 2 wird gleichzeitig umgekehrt Knoten 2 als Nachfolger von Knoten 1 eingetragen. So kann einfach der Aufbau eines Baumes wie in Abbildung 16 erfolgen.

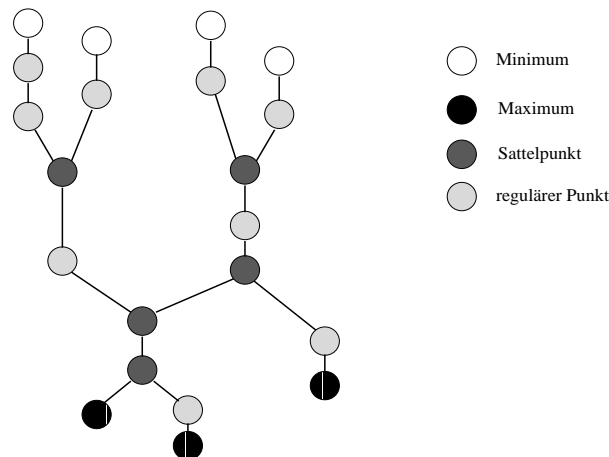


Abbildung 16: Baumstruktur

Es ist dadurch möglich den Konturbaum von einem Knoten aus in jede Richtung zu durchwandern und jeder Knoten kann erreicht werden.

### 3.6.2 Durchlaufen des Volumens

Zur einfacheren Berechnung des Konturbaums erfolgt der Aufbau eines Baums, der nicht nur die kritischen Punkte als Knoten enthält, sondern zunächst alle Gitterpunkte. Zwi-

schen den kritischen Knoten liegen im Baum also alle regulären Punkte bzw. Knoten die zur entsprechenden Kontur gehören.

Beim Aufbau des Baums werden die Funktionswerte von  $-\infty$  nach  $+\infty$  verfolgt, also es erfolgt ein Durchlaufen der Skalarwerte vom Minimum zum Maximum. Das Durchlaufen des Volumens stoppt sobald der Wert eines Gitterpunktes erreicht worden ist. Dies ermöglicht ein Verfolgen der bestehenden Konturen. Für jeden erreichten Gitterpunkt wird ein Knoten im Baum aufgenommen. Der Baum wird so an mehreren Stellen gleichzeitig aufgebaut. Jede offene Stelle, an der der Baum weiterwachsen kann, entspricht einer gerade verfolgten Kontur. Zellen, die den aktuell betrachteten Datenwert enthalten, werden als aktiv bezeichnet. Für jede aktive Zelle wird gespeichert, zu welcher Kontur sie gehört. Für jede aktive Kontur besteht ein Zeiger auf den letzten zugehörigen Knoten im Baum. Die Identifizierung und Handhabung der unterschiedlichen Vertextypen erfolgt in den nächsten beiden Abschnitten. Reguläre Knoten, die am Rand des Volumens vorkommen und noch zu keiner aktiven Kontur zugeordnet werden können, werden gesondert behandelt. Sie gehören zu der aus dem negativen unendlichen kommenden Kontur und werden über ihren Index in den sogenannten Randbereich aufgenommen, um eine Abdeckung dieses Bereichs in der späteren Visualisierung zu gewährleisten.

Abbildung 17 zeigt das Durchlaufen eines Gitters anhand eines Beispiels in 2D. Links das Gitter, rechts der sich im Aufbau befindliche Konturbaum. Aktuell betrachteter Funktionswert: 16, dunkel markiert: bereits abgearbeitetes Gebiet, grau: aktiver Bereich. Aktive Zellen besitzen einen Zeiger auf die zugehörige Kontur bzw. die zugehörige Stelle im Baum.

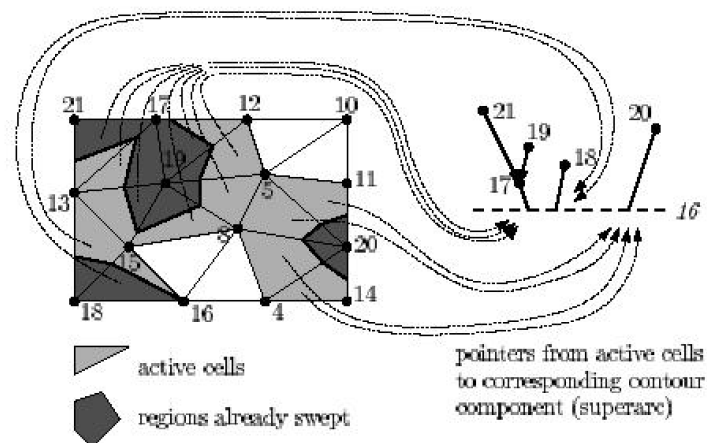


Abbildung 17: Durchlaufen des Volumens

### 3.6.3 Identifikation des Vertextyps

Die Identifikation des Typs eines Vertex wird nach der Methode von Kraus/Ertl [7] durchgeführt. Als kritische Punkte werden lokale Maxima und Minima, sowie Sattelpunkte betrachtet. Die restlichen Punkte sind reguläre Punkte, die ebenfalls zunächst in den Konturbaum mit aufgenommen werden. Alle kritischen Punkte entstehen an Vertexpositionen. Für Randpunkte gilt: Nachbarknoten die außerhalb des Volumens liegen würden, haben Datenwert  $-\infty$ .

Aus der virtuellen Tetraederzerlegung entstehen, wie bereits erwähnt, zwei Nachbarschaftsverhältnisse mit jeweils sechs bzw. achtzehn Nachbarn (siehe Abbildung 18, links: sechs Nachbarn, rechts: achtzehn Nachbarn).

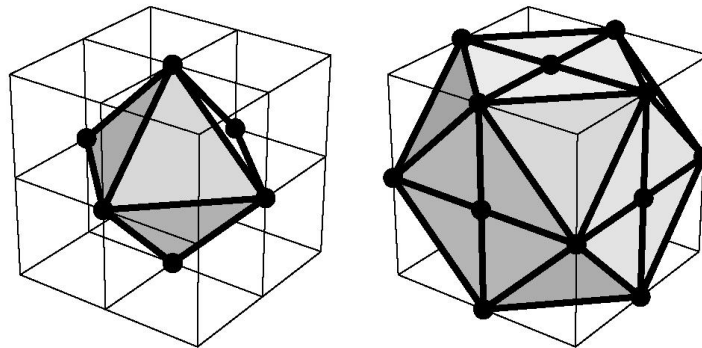


Abbildung 18: Nachbarschaftsverhältnisse

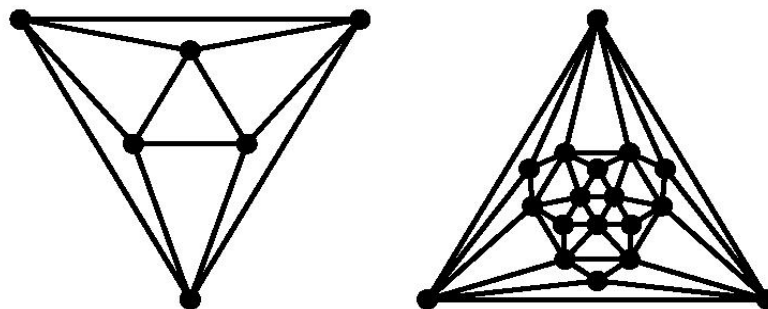


Abbildung 19: Nachbarschaftsgraph

Aus diesen Nachbarschaften ergibt sich jeweils ein Graph mit den Vertices als Knoten und den Kanten zwischen den Vertices als Kanten des Graphen (siehe Abbildung 19, links: sechs Nachbarn, rechts: achtzehn Nachbarn). Jeder Knoten (Nachbarvertex) wird anhand seines Datenwertes mit 0, bei kleinerem Datenwert (als der zu bestimmende Vertex), oder 1, bei größerem Datenwert, klassifiziert. Dies ist eindeutig möglich, da aufgrund der Datenaufbereitung keine Nachbarvertices den gleichen Wert besitzen. Anschließend

wird der Graph nach der Anzahl der zusammenhängenden Regionen mit 0 bzw. 1 Werten durchsucht. Hierdurch erfolgt die Identifikation des Vertextyps.

- Nur eine 0-Region:  
Maximum
- Nur eine 1-Region:  
Minimum
- Je eine 0- und 1-Region:  
regulärer Punkt
- mehr als insgesamt zwei Regionen:  
Sattelpunkt  
Je nach Anzahl der 0- bzw. 1-Regionen kann der spezielle Typ des Sattelpunktes festgestellt werden.

Abbildung 20 zeigt unterschiedliche Möglichkeiten für das Auftreten eines Sattelpunktes in 3D. Eine Kugel um den Sattelpunkt schneidet die von ihm ausgehende Kontur in mindestens zwei unterschiedlichen Komponenten. Es können auch mehrere Fälle gleichzeitig auftreten. Weitere Fälle können am Rand des Volumens vorkommen.

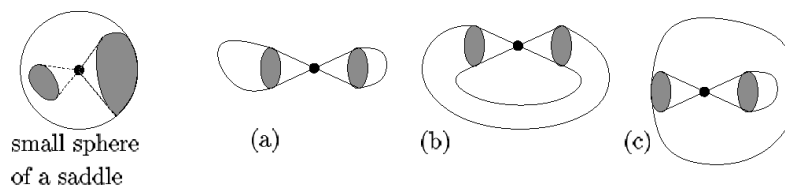


Abbildung 20: Sattelpunkt in 3D

### 3.6.4 Behandlung der Vertextypen

Für jeden Vertex entsteht ein neuer Knoten im Konturbaum. Je nach Typ des Vertex muss eine unterschiedliche Behandlung erfolgen.

#### **Lokales Minimum:**

Es entsteht eine neue Kontur. Ein neuer Minimum-Knoten wird im Konturbaum erzeugt. Dieser wird als letzter Knoten dieser Kontur gespeichert. Alle Tetraederzellen, die an den Vertex angrenzen, werden aktiviert und besitzen die entsprechende Konturnummer.

#### **Lokales Maximum:**

Eine vorhandene Kontur endet. Ein neuer Maximum-Knoten wird im Konturbaum erzeugt. In den umliegenden aktiven Tetraederzellen wird die zugehörige Konturnummer gesucht. Der neue Knoten wird als letzter der Kontur gespeichert und der bisher letzte

Knoten der Kontur zeigt auf ihn. Alle aktiven Knoten dieser Kontur werden gelöscht. Sollten mehrere Konturen in diesem Knoten enden werden entsprechend Vereinigungen erzeugt, die diese zuerst verschmelzen. Die bisher letzten Knoten der Konturen zeigen dann auf die entsprechende Vereinigung. Alle aktiven Knoten dieser Konturen werden dann ebenfalls gelöscht.

#### Regulärer Punkt:

Ein neuer regulärer Knoten wird im Konturbaum erzeugt. In den umliegenden aktiven Tetraederzellen wird die zugehörige Konturnummer gesucht. Der neue Knoten wird als letzter dieser Kontur gespeichert und der bisher letzte Knoten der Kontur zeigt auf ihn. Alle Tetraederzellen, die an den Vertex angrenzen und noch nicht aktiviert sind, werden aktiviert und besitzen die entsprechende Konturnummer.

#### Sattelpunkt:

Zuerst wird der Typ des Sattelpunktes festgestellt, das heißt wie viele Konturen sich vereinigen bzw. entstehen. Entsprechend dieser Anzahl werden dann Gabelungsknoten und Vereinigungsknoten erzeugt. In den umliegenden Tetraederzellen werden wieder die Konturnummern gesucht. Die Knoten werden entsprechend in den Baum eingefügt und die aktiven Zellen der beendeten Konturen gelöscht. Die angrenzenden Zellen werden aktiviert und mit der für sie zugehörigen Konturnummer versehen. Die neu entstandenen Knoten werden untereinander verbunden (siehe Beispiel in Abbildung 21. Der Sattelpunkt besteht aus zwei Vereinigungen und einer Gabelung. Die Konturen 3, 4 und 5 enden, die Konturen 6 und 7 entstehen).

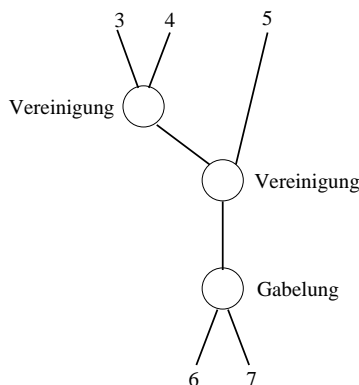


Abbildung 21: Ein Sattelpunkt bestehend aus mehreren Knoten

### 3.6.5 Vereinfachung des Baumes

Bei der Berechnung des Konturbaums werden zunächst alle Vertices des Volumens als Knoten in den Baum aufgenommen. Dies bedeutet, dass für jede Kontur jeder darin enthaltene Vertex im Baum vorkommt. Um aber jeden in einer Kontur enthaltenen Isowert abzudecken und eine Startzelle für die späteren Marching Tetrahedra Verfahren zu erhal-

ten, ist nur ein Weg zwischen zwei kritischen Punkten, bzw. durch eine Kontur, notwendig. Es entstehen also sehr viele reguläre Knoten im Baum, die für seine spätere Nutzung überhaupt nicht notwendig sind. Deshalb wird nach jedem neu entstandenen kritischen Punkt und dem damit verbundenen Ende einer Kontur, ein Verfahren zur Vereinfachung des Baumabschnitts dieser Kontur durchgeführt. Vom neu erzeugten kritischen Knoten wird ein Weg durch die Kontur zurück zum letzten kritischen Knoten des Baums, dem Startpunkt der Kontur, gesucht. In den Nachbarvertizes des letzten Knotens wird nach dem zu der Kontur gehörenden Vertex mit dem kleinsten Datenwert gesucht. Von diesem geht es rekursiv weiter, bis der Startpunkt der Kontur erreicht ist. Die so durchwanderten Vertizes ergeben die verbleibenden Knoten im Baum. Die restlichen regulären Knoten der Kontur werden gelöscht, da sie nun nicht mehr benötigt werden. Der sich ergebende Weg ist nicht immer der kürzeste, aber besitzt doch sehr viel weniger reguläre Knoten. Dies bedeutet eine enorme Vereinfachung des Baums und ermöglicht eine bessere Übersichtlichkeit bei seiner Darstellung.

Abbildung 22 zeigt ein Beispiel zur Vereinfachung des Baums. Um eine einfachere aber deutlichere Darstellung zu ermöglichen jedoch nur in 2D. Weiß sind die zwei die Kontur begrenzenden kritischen Punkte, schwarz die Punkte, die zunächst alle in den Baum aufgenommen, aber dann wieder entfernt wurden. Der gewählte Weg in der Kontur entspricht den grauen Punkten.

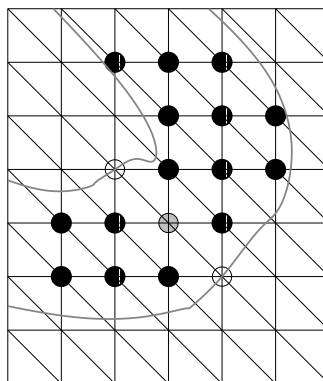


Abbildung 22: Baumvereinfachung, Beispiel in 2D

### 3.6.6 Saatzellenbestimmung

Für das Marching Tetrahedra Verfahren zur Berechnung der überstrichenen Volumen und zum Zeichnen der Isoflächen wird für jede Kante des entstandenen Konturbaums eine Saatzelle benötigt. Diese werden durch ein einmaliges Durchlaufen des Konturbaums nach seiner Berechnung bestimmt. Jede Kante wird durch zwei Knoten begrenzt. Über die Indizes der beiden Knoten wird eine entsprechende Zelle gesucht, die beide Knoten enthält und so als Startzelle benutzt werden kann.

### 3.6.7 Berechnung der Volumen

Um in der Darstellung des Konturbaums größere und damit wichtigere Datenbereiche von Bereichen unterscheiden zu können, die nur durch Rauschen entstanden sind, kann für jeden Knoten ein Volumenwert bestimmt werden. Diese Option kann vor der Berechnung des Baums zusätzlich aktiviert werden. Der zu berechnende Volumenwert repräsentiert das von der jeweiligen Kontur überstrichene Volumen zwischen dem Knoten selbst und seinem direkten Vorgänger. Für die Berechnung dieses Volumens ist ein erneutes zweimaliges Durchlaufen des Baumes notwendig.

Im ersten Durchlauf wird das gesamte Volumen des Teiles der Kontur berechnet, deren Datenwert kleiner als der eines Knotens ist. Dies erfolgt für jeden Knoten im Konturbaum. Die Berechnung wird über ein Marching Tetrahedra Verfahren durchgeführt, dessen Startzelle aus den Saatzellen aus Abschnitt 3.6.6 bestimmt wird. Da die Zerlegung des kartesischen Gitters in Tetraederzellen nur virtuell ist, werden die Nachbarzellen für die Weiterführung des Verfahrens jedesmal neu aus den Koordinaten der Vertizes bestimmt. Für die zu behandelnden Tetraederzellen entstehen unter Berücksichtigung der Symmetrie fünf unterschiedliche Fälle.

**Fall 1:**

Alle Tetraedervertizes besitzen einen kleineren Datenwert. Das Volumen der gesamten Zelle wird zum berechneten Volumen addiert. Die Volumenberechnung muss an jeder angrenzenden Zelle fortgeführt werden (Abbildung 23, Fall 1).

**Fall 2:**

Ein Vertex mit kleinerem Datenwert, drei Vertizes mit größerem Datenwert. Es werden drei Schnittpunkte auf den Kanten berechnet. Das Volumen des sich am Vertex mit kleinerem Datenwert ergebenden Tetraeders wird addiert. Die Volumenberechnung muss in allen Zellen fortgesetzt werden, die an eine Kante mit Schnittpunkt angrenzen (Abbildung 23, Fall 2).

**Fall 3:**

Ein Vertex mit größerem Datenwert, drei Vertizes mit kleinerem Datenwert. Es werden drei Schnittpunkte auf den Kanten berechnet. Das zu addierende Volumen ergibt sich aus dem Volumen des gesamten Tetraeders minus das Volumen des sich ergebenden Tetraeders am Vertex mit größerem Datenwert. Die Volumenberechnung muss an jeder angrenzenden Zelle fortgeführt werden (Abbildung 23, Fall 3).

**Fall 4:**

Je zwei Vertizes mit kleinerem und größerem Datenwert. Es werden vier Schnittpunkte auf den Kanten berechnet. Zur Volumenberechnung wird das sich ergebende Polyeder in Tetraeder zerlegt. Dies erfolgt durch Zerlegung der vorhandenen Vierecke in Dreiecke und anschließender Verbindung eines Vertex mit jedem Dreieck. Die Volumenberechnung muss an allen Zellen fortgesetzt werden, die an eine Kante mit Schnittpunkt oder an die Kante zwischen den Vertizes mit kleinerem Datenwert angrenzen (Abbildung 23, Fall 4).

**Fall 5:**

Alle Tetraedervertizes besitzen einen größeren Datenwert. Die gesamte Zelle muss nicht behandelt werden. Die Volumenberechnung wird an keiner angrenzenden Zelle fortgeführt (Abbildung 23, Fall 5).

Die notwendige Schnittpunktinterpolation auf den Kanten des Tetraeders wird im Abschnitt der Visualisierung der Isoflächenkomponenten behandelt. Es darf selbstverständlich jede Zelle nur einmal bearbeitet werden. Der Zustand der Zelle (bearbeitet/nicht bearbeitet) wird deshalb gespeichert.

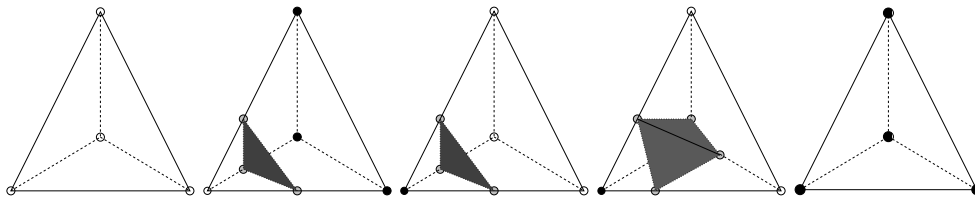


Abbildung 23: Fälle 1-5 des Marching Tetrahedra  
schwarzer Punkt: Wert größer, weißer Punkt: Wert kleiner

**Volumen eines Tetraeders**

Die Berechnung des Volumens eines Tetraeders erfolgt einfach durch die Berechnung der Determinante einer Matrix. A, B, C, D seien die Koordinaten der Eckpunkte des Tetraeders. Aus den Vektoren  $p = \overrightarrow{AB}$  (Vektor von A nach B),  $q = \overrightarrow{AC}$  und  $r = \overrightarrow{AD}$  ergibt sich folgende 3x3 Matrix ( $p_x$  entspricht der x-Komponente des Vektors p):

$$\begin{bmatrix} p_x & p_y & p_z \\ q_x & q_y & q_z \\ r_x & r_y & r_z \end{bmatrix}$$

Das Volumen ist ein Sechstel der Determinante:

$$V = \frac{1}{6}(p_x q_y r_z + p_y q_z r_x + p_z q_x r_y - p_z q_y r_x - p_y q_x r_z - p_x q_z r_y)$$

Der zweite Durchlauf ist zur Berechnung der Volumendifferenz zwischen den Baumknoten notwendig. Hierbei wird einfach von jedem Volumen eines Baumknotens das Volumen seines Vorgängers subtrahiert. Dabei muss beachtet werden, daß ein Sattelknoten unter Umständen mehr als nur einen Vorgänger besitzt. Deshalb müssen hierbei alle Vorgänger bestimmt und ihr Volumenwert subtrahiert werden.

**3.7 Speichern des Konturbaums/Datenformat**

Zur Speicherung des Konturbaums werden alle Knoten durchlaufen und alle vorhandenen Daten wie Knotenindex, Vorgänger bzw. Nachfolger und Konturnummern im ASCII-Format in eine Datei mit Endung .cts (Contour Tree Save File) geschrieben. Zunächst

wird die Anzahl der Knoten jedes Typs gespeichert. Dann werden die Arrays für die einzelnen Knotentypen durchlaufen und die Daten für jeden Knoten geschrieben. Jeder Knoten belegt genau eine Zeile. Zuletzt erfolgt noch die Speicherung der Saatzellen, der Volumenwerte und der speziell zu behandelnden Randknoten. Diese Werte belegen jeweils eine Zeile. Bei den Saatzellen und den Volumendaten sind zusammengehörende Komponenten durch ein Flag voneinander abgetrennt.

Die folgende Tabelle beschreibt den Aufbau der .cts Datei.

Zeile	Beschreibung
Contour Tree Save File	Kopfzeile zur Beschreibung des Dateityps
Max: n1 Min: n2 Simple: n3 Bifurcation: n4 Junction: n5	Anzahl der Maxima Anzahl der Minima Anzahl der regulären Knoten Anzahl der Gabelungsknoten Anzahl der Vereinigungsknoten
Max Nodes: a b c ↓ ... n1 Zeilen ... ↓	Markiert den Beginn der Daten der Maxima Knotenindex a, Vorgängerindex b, Konturnummer c
Min Nodes: a b c ↓ ... n2 Zeilen ... ↓	Markiert den Beginn der Daten der Minima Knotenindex a, Nachfolgerindex b, Konturnummer c
Simple Nodes: a b c d ↓ ... n3 Zeilen ... ↓	Markiert den Beginn der Daten der regulären Knoten Knotenindex a, Nachfolgerindex b, Vorgängerindex c, Konturnummer d
Junction Nodes: a b c d e f g ↓ ... n5 Zeilen ... ↓	Markiert den Beginn der Daten der Vereinigungsknoten Knotenindex a, Nachfolgerindex b, Nachfolgerkonturnummer c, linker Vorgängerindex d, linke Konturnummer e, rechter Vorgängerindex f, rechte Konturnummer g
Bifurcation Nodes: a b c d e f g ↓ ... n4 Zeilen ... ↓	Markiert den Beginn der Daten der Gabelungsknoten Knotenindex a, linker Nachfolgerindex b, linke Konturnummer c, rechter Nachfolgerindex d, rechte Konturnummer e, Vorgängerindex f, Vorgängerkonturnummer g
Seed Cells: -1 a b c d e ... ⇒ alle Zellen in einer Zeile	Markiert den Beginn der Daten der Saatzellen Flag -1, Konturnummer a, linker Index der Kante b, rechter Index der Kante c, Index der Würfelzelle d, Index des Tetraders der Würfelzelle e
Volumes: -1 a b ... ⇒ alle Zellen in einer Zeile	Markiert den Beginn der Volumendaten Flag -1, Knotenindex a, Volumen b
Zeros: a b ... ⇒ Indizes in einer Zeile	Markiert den Beginn der Randknoten Knotenindizes der Randknoten



## 4 Visualisierung des Konturbaums

Für die Visualisierung des Konturbaums stehen zwei unterschiedliche Varianten zur Verfügung, zwischen denen umgeschaltet werden kann. Die erste Variante beinhaltet nur das Zeichnen von Knoten und Kanten als Kreise und Linien mit unterschiedlicher Farbgebung für die Knotentypen (Abschnitt 4.2.1). Die zweite Variante berücksichtigt die berechneten überstrichenen Volumen der Konturen (Abschnitt 4.2.2).

### 4.1 Laden des Konturbaums und der Daten

Die Volumendaten besitzen dasselbe Format und werden auf dieselbe Art eingelesen wie in Abschnitt 3.3 bereits beschrieben. Um auf exakt denselben Daten wie bei der Berechnung des Konturbaums zu arbeiten, erfolgt auch hier eine Datenaufbereitung durch Addieren von Werten an den Vertizes wie in Abschnitt 3.4, obwohl dies zur Visualisierung der Isoflächen und des Baums nicht unbedingt notwendig wäre. Es vereinfacht jedoch zusätzlich die Handhabung der einzelnen Tetraederzellen da keine Nachbarvertizes gleiche Werte haben.

Nach Auswahl der \*.cts Datei, die die Daten des Konturbaums enthält, wird jede einzelne Zeile eingelesen und bearbeitet. Zunächst erfolgt die Überprüfung der Kopfzeile der Datei, um die Richtigkeit des Dateityps festzustellen. Dann folgt das Einlesen und Speichern der Anzahl der Knoten der unterschiedlichen Knotentypen. Anschließend werden die Knoten selbst eingelesen. Nach jedem Erkennen einer Markierungszeile (Max Nodes:, Simple Nodes:, usw.) wird der Typ der nächsten einzulesenden Knoten geändert. Dieser Typ wird in einem Zustand gespeichert. Beim Einlesen einer Zahlenzeile wird ein Knotenobjekt erzeugt und je nach bestehendem Typ-Zustand die 3 bis 7 Eigenschaften des Knotens eingelesen und gespeichert.

Das Einlesen der Saatzellen, Volumendaten und der Randknoten wird ebenfalls über den Zustand erkannt. Dieser wird nach Erkennung der Markierungszeile wieder geändert. Nur wird die folgende Zahlenzeile komplett in ein entsprechendes Array gespeichert. Die in den Zeilen vorkommenden Flags werden als normale Werte mit übernommen, da sie bei der Saatzellen- und Volumenwertsuche für die Visualisierung benötigt werden. Bei unvollständigen Knoten oder einer sonstigen fehlerhaften Zeile erfolgt eine Fehlerausgabe. Das Bearbeiten der Datei wird bei einer Fehlermeldung nicht abgebrochen.

### 4.2 Visualisierung

#### Volumenbestimmung

Für jeden Knoten ist ein berechneter Volumenwert gespeichert, sollte dies bei der Berechnung des Baums aktiviert worden sein. Dieser kann über den Knotenindex nach einem Flag im Volumen-Array gefunden werden und geht beim Zeichnen in der entsprechenden

Variante in die y-Komponente des jeweiligen Knotens mit ein.

#### 4.2.1 Visualisierung mit Knoten und Kanten

In der ersten Art der Visualisierung des Konturbaums werden für Knoten nur Kreise gezeichnet, die durch Kanten (bzw. Linien) verbunden werden. Der Baum wird von einem Startpunkt aus durchwandert. Die Zeichenroutine wird jeweils für den (die) nächsten zu behandelnden Knoten rekursiv aufgerufen. Um beim Durchlaufen des Baumes (Graphen) eine Doppelbehandlung von Knoten zu verhindern, werden abgearbeitete Knoten markiert. Für die unterschiedlichen Knotentypen werden verschiedene Farben verwendet:

Blau	Minimum
Rot	Maximum
Gelb	regulärer Punkt
Grün	Sattelpunkt/Gabelung/Vereinigung

#### Gitter

Zur Beschränkung des Zeichenbereichs werden eine x- und eine y-Achse eingezeichnet. Während die y-Achse in diesem Zeichenmodus keine besondere Bedeutung hat, spiegelt die x-Achse den Datenwert eines Knotens wieder. Min/Max und einige Zwischenwerte werden an der Achse angezeigt.

#### Absolutes Minimum

Das absolute Minimum wird ebenfalls als Knoten gezeichnet. Es erfolgt zunächst nur das Zeichnen des Knotens (noch keine Kante). Ein Kante zum absoluten Minimum entsteht später zu jeder Kontur, die aus der aus dem negativen unendlichen kommenden Kontur entsteht. Diese Kanten werden zur Unterscheidung und aus Gründen der Übersichtlichkeit nur gestrichelt gezeichnet. Da der Algorithmus den Baum komplett durchwandert, kehrt man auf allen Kanten, die an das absolute Minimum anschließen, dahin zurück. Bei einer Rückkehr wird kein neuer Knoten gezeichnet, sondern nur die Kante hinzugefügt. Abbildung 24 zeigt ein absolutes Minimum.

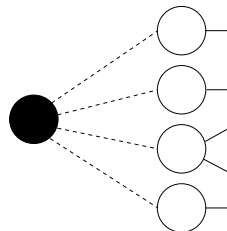


Abbildung 24: Knoten und Kanten: Behandlung des absoluten Minimums

Da der Rand und damit die Kanten zum absoluten Minimum meist keine interessanten Bereiche darstellen, können die Kanten bei der Visualisierung abgeschaltet werden.

### Minimum/Maximum

Der Zeichenalgorithmus startet beim ersten Minimum. Sollte keines vorhanden sein, wird an einem Knoten begonnen, der an das absolute Minimum anschließt. Der Knoten (Kreis) wird an seiner Position gezeichnet. Der neue y-Wert wird bei der Rekursion mitgeliefert, der neue x-Wert ergibt sich aus dem Datenwert des Knotens. Von der neuen Position wird zur Alten eine Kante zurück gezeichnet. An einem Minimum oder Maximum endet die Rekursion, da keine weitere Verfolgung mehr möglich ist.

### regulärer Knoten

Der Knoten wird ebenfalls an seiner neuen Position gezeichnet, die wie zuvor bestimmt wird. Ebenso wird eine Kante zur alten Position zurück gezeichnet. Je nach Richtung, aus der der Knoten erreicht wird, wird sein Nachfolger oder sein Vorgänger als nächster Knoten behandelt. Abbildung 25 zeigt einen regulären Knoten.

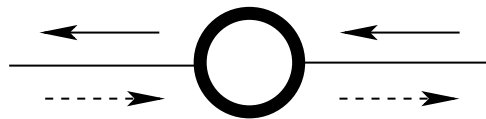


Abbildung 25: Knoten und Kanten: Behandlung eines regulären Knotens

### Gabelung/Vereinigung

Gabelungen und Vereinigungen werden praktisch gleich behandelt. Es sind in beiden Fällen drei Richtungen zu behandeln. Der Knoten wird ebenfalls an seiner neuen Position gezeichnet, die wie zuvor bestimmt wird. Ebenso wird eine Kante zur alten Position zurück gezeichnet. Je nach kommender Richtung werden zwei Rekursionen mit den entsprechenden Knoten gestartet. Abbildung 26 zeigt die drei möglichen Rekursionsrichtungen. Um möglichst wenig Überschneidungen beim Zeichnen des Baums zu bekommen, werden die y-Werte für die nächsten Knoten entsprechend angepasst.

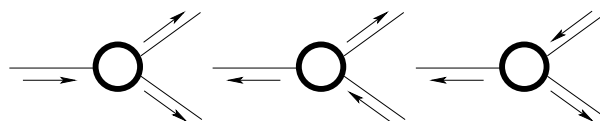


Abbildung 26: Knoten und Kanten: Behandlung einer Vereinigung/Gabelung

### Sattelpunkte

Beim Erreichen eines Sattelpunktes werden alle darin enthaltenen Gabelungen und Vereinigungen in einem Punkt gezeichnet. Die Bestimmung der Position und das Zeichnen der Kante zurück entspricht der vorherigen Vorgehensweise. Alle zugehörigen Kanten laufen in den einem Punkt zusammen. Es ergeben sich jedoch je nach Eingangs- und Ausgangsgrad mehr abgehende Richtungen.

### 4.2.2 Visualisierung mit Volumendaten

Bei der zweiten zur Verfügung stehenden Art der Visualisierung des Konturbaums werden die für jeden Knoten berechneten Volumenwerte berücksichtigt. Sollten keine Volumenwerte berechnet worden sein, fällt diese Darstellungsmöglichkeit weg. Die Länge des Knotens in y-Richtung wird aus dem für ihn berechnete Volumen bestimmt. Dabei wird der Volumenwert noch durch die Differenz der x-Position (-Wert) des Knotens selbst und der x-Position (-Wert) seines Vorgängerknotens geteilt. Dies verringert die y-Komponente vor allem im Rauschbereich, der aus Übersichtlichkeitsgründen so klein wie möglich gezeichnet werden soll. Zusätzlich erfolgt eine Skalierung der Volumenwerte, die vom Benutzer eingestellt werden kann. Die x-Position wird weiterhin aus dem Datenwert bestimmt. Die Kante zwischen zwei Knoten besteht aus zwei Linien, die jeweils die oberen und unteren Ecken der Knotens miteinander verbinden. Sie bildet also eine Fläche, die von den Volumenwerten der Knoten abhängt. In der Farbgebung wird nur zwischen rot für die Knoten und blau für die Kanten unterschieden.

Die Rekursion beim Durchlaufen des Baumes erfolgt wie im Abschnitt zur Visualisierung mit Knoten und Kanten. Startpunkt ist wieder das erste Minimum oder ein an das absolute Minimum anschließender Knoten. Die Abbildungen 27 bis 30 zeigen die Handhabung der unterschiedlichen Knotentypen.

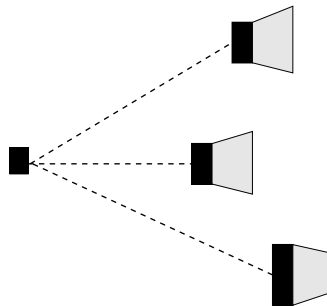


Abbildung 27: Volumen: Behandlung des absoluten Minimums

Bei der Behandlung des absoluten Minimums ist zu beachten, dass die Kante zwischen diesem Knoten und seinen Nachfolgern aus Gründen der Übersichtlichkeit nicht als Fläche sondern wieder nur als gestrichelte Linie gezeichnet wird. Diese Linien sind jedoch wieder abschaltbar.

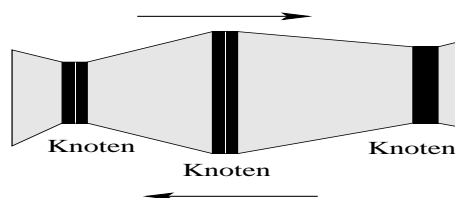


Abbildung 28: Volumen: Behandlung eines regulären Knotens

Gabelungen und Vereinigungen werden wieder gleich behandelt. Bei einer Gabelung wird das Volumen für die beiden abgehenden Kanten aufgeteilt (entsprechend bei Vereinigungen).

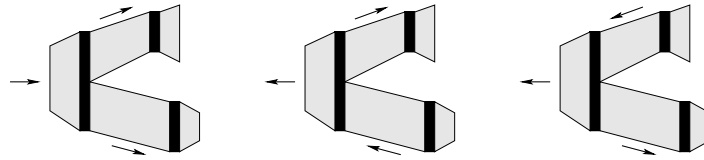


Abbildung 29: Volumen: Behandlung einer Gabelung/Vereinigung

Für Sattelpunkte ist eine feinere Aufteilung des Volumens entsprechend der Anzahl der kommenden und abgehenden Kanten notwendig. Hierfür muss zunächst der Eingangs- und Ausgangsgrad bestimmt werden.

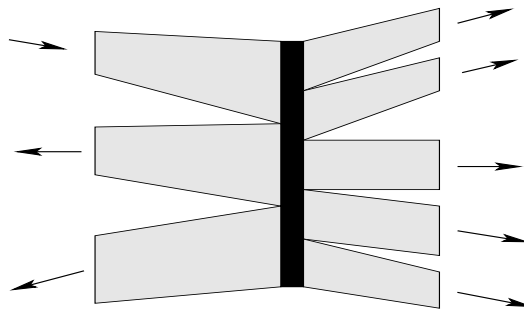


Abbildung 30: Volumen: Behandlung eines Sattelpunktes

### 4.3 Picking und Isowertselektion

Zur gezielten Selektion von Isowerten und Flächenkomponenten können im Konturbaum Kanten oder Knoten selektiert werden. Dafür muss zunächst in den Selektionsmodus umgeschaltet werden. Anschließend kann durch einfaches Klicken auf einen Knoten oder eine Kante diese selektiert werden. Bei einem Knoten wird der Datenwert des Knotens als Isowert selektiert, bei einer Kante kann der Isowert aus einem Bereich ausgewählt werden. Die Grenzen dieses Bereichs sind gegeben durch die Datenwerte der die Kante begrenzenden Knoten.

Das absolute Minimum kann ebenfalls selektiert werden. Wie bei den anderen Minima entsteht jedoch dabei keine Isofläche. Bei der Selektion einer zum absoluten Minimum führenden Kante kann ebenfalls der Isowert zwischen dem kleinsten im Datensatz vorhandenen Wert und dem Wert des zweiten Knotens der Kante eingestellt werden. Die Visualisierung beschränkt sich jedoch auf die Zellen die zum Randbereich gehören. In beiden Zeichenmodi wird die Selektion auf dieselbe Art durchgeführt.

#### 4.4 Weitere Interaktionsmöglichkeiten

Bei der Darstellung des Konturbaums stehen außer der Selektion noch folgende Interaktionsmöglichkeiten zur Verfügung:

- Umschalten zwischen den zwei Arten der Visualisierung des Konturbaums:  
Visualisierung mit Knoten und Kanten und Visualisierung mit Volumendaten
- Kanten zum absoluten Minimum:  
Zeichnen An/Abschalten
- Skalierung der Volumendaten:  
Verkleinerung/Vergrößerung der y-Komponente im Volumen Modus
- Umschalten der Maus-Interaktionen:  
Verändern der Ansicht auf den Konturbaum und Selektion von Knoten oder Kanten für die Visualisierung der Flächen.

Die Ansicht auf den Konturbaums kann verschoben, vergrößert oder verkleinert werden. Eine ausführliche Beschreibung der Bedienung folgt im Anhang Programmstruktur.

## 5 Visualisierung der Isoflächenkomponenten

Die Visualisierung der Isoflächenkomponenten erfolgt über ein Marching Tetrahedra (MT) Verfahren. Dies entspricht dem Marching Cubes (MC) Verfahren abgeändert für unstrukturierte Gitter. MT wird auf Grund der einfacheren Vorgehensweise durch weniger Fälle bei der Flächenpositionierung in den Zellen verwendet. Ausgehend von einer Startzelle wird für den selektierten Isowert die zugehörige Isofläche verfolgt und gezeichnet. Da das Verfahren am Rand der gewählten Kontur abbricht, wird nicht die gesamte Isofläche gezeichnet, sondern nur die entsprechende Komponente. Da die Daten auf einem kartesischen Gitter gegeben sind, erfolgt die Zerlegung der Würfelzellen in Tetraeder wieder virtuell (siehe Seite 16).

Eine ausführliche Beschreibung des Verfahrens kann unter anderem auch in [17] nachgelesen werden.

### 5.1 Saatzellenidentifizierung

Nach der Selektion aus dem Konturbaum stehen entweder ein Knotenindex, bei selektiertem Knoten, oder zwei Knotenindizes, bei selektierter Kante, zur Verfügung. Anhand dieser Indizes kann die benötigte Saatzelle identifiziert werden. Bei nur einem Index stehen mehrere Zellen zur Auswahl, jedoch kann jede verwendet werden, da nur eine Startzelle für das Marching Tetrahedra Verfahren benötigt wird und auch bei Sattelpunkten trotzdem alle angrenzenden Konturen verfolgt werden, da der Isowert direkt auf einem Gitterpunkt liegt.

### 5.2 Marching Tetrahedra

Nach der Saatzellenidentifizierung ist eine Startzelle für das Marching Tetrahedra Verfahren zur Visualisierung der selektierten Isoflächenkomponente vorhanden. Je nach Anzahl der Vertizes mit größerem oder kleinerem Datenwert als der Selektierte, können unter Berücksichtigung der Symmetrie vier Fälle auftreten. Vertizes mit gleichem Datenwert müssen je nach Wert der anderen Tetraedervertizes als größer oder kleiner eingestuft werden, da sonst unter Umständen Nachbarzellen vergessen werden könnten.

#### **Sonderfall absolutes Minimum**

Bei der Selektion des absoluten Minimums oder einer dahin führenden Kante wird keine Rekursion (keine Verfolgung der Fläche) ausgeführt. Da alle zum Rand gehörenden Zellen gespeichert wurden, werden einfach nur alle diese Zellen betrachtet und entsprechend einzeln behandelt.

### 5.2.1 Schnittpunktinterpolation

Um die Teilfläche in einer Zelle zeichnen zu können müssen in jeder Zelle die Schnittpunkte der Isofläche mit den Zellkanten bestimmt werden. Es werden nur Kanten mit einem Vertex mit größerem und einem Vertex mit kleinerem Datenwert geschnitten. Deshalb erfolgt zunächst eine Klassifizierung der Vertizes (größerer oder kleinerer Datenwert). Die Berechnung der Schnittpunkte erfolgt durch lineare Interpolation zwischen den beiden Vertizes der geschnittenen Kante (siehe Abbildung 31).

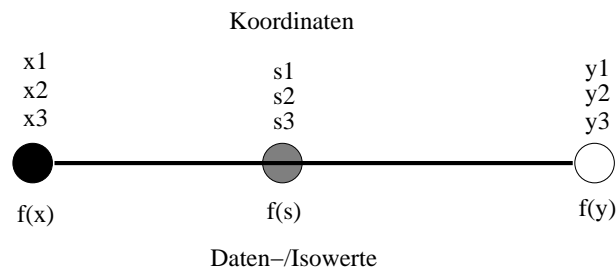


Abbildung 31: Schnittpunktinterpolation  
 $x_1 \equiv x$ -Koordinate von Vertex  $x$ , usw.

$f(s)$  bezeichnet den selektierten Isowert,  $f(x)$  und  $f(y)$  die Werte an den Vertizes, dabei sei  $f(y) < f(x)$ . Die  $x_i$  ( $y_i$ ,  $s_i$ ) sind die Koordinaten der Vertizes (bzw. des Schnittpunktes). Die Schnittpunktkoordinaten werden über folgendes Schema berechnet:

$$w = \frac{f(x) - f(s)}{f(x) - f(y)}$$

$$s_1 = (x_1 - y_1) * w + x_1$$

$$s_2 = (x_2 - y_2) * w + x_2$$

$$s_3 = (x_3 - y_3) * w + x_3$$

### 5.2.2 Zellbehandlung

Nach der Interpolation der Schnittpunkte können die Flächen gezeichnet werden. Da bei Zellen, deren Vertizes nur größere oder kleinere Datenwerte besitzen, keine zu zeichnende Fläche vorhanden ist, reduzieren sich die zu behandelnden Fälle auf zwei (Symmetrie). Es existieren keine aufzulösenden Mehrdeutigkeiten in den beiden Fällen. Die sich ergebenden Flächen können einfach durch Verbinden der Schnittpunkte gezeichnet werden.

#### Fall 1 (Abbildung 32, links):

Ein Vertex mit größerem und drei mit kleinerem Datenwert (bzw. ein Vertex mit kleinerem

und drei mit größerem Datenwert). Es ergeben sich drei Schnittpunkte. Diese ergeben das zu zeichnende Dreieck.

**Fall 2 (Abbildung 32, rechts):**

Je zwei Vertices mit größerem und kleinerem Datenwert. Es ergeben sich vier Schnittpunkte. Diese ergeben ein Viereck, das durch Zerlegung in zwei Dreiecke gezeichnet wird.

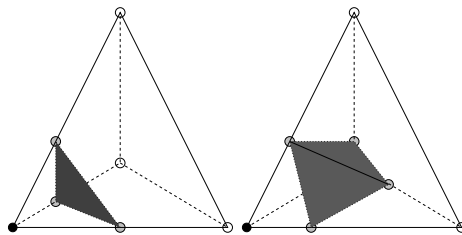


Abbildung 32: Isoflächen beim Marching Tetrahedra

Das Verfahren muss zur Verfolgung der Isofläche in allen Nachbarzellen fortgesetzt werden, die an eine Kante mit Schnittpunkt angrenzen. Jede behandelte Zelle wird markiert, um eine mehrfache Bearbeitung zu verhindern.

**Normalen (Beleuchtung)**

Zur besseren Darstellung der Flächen kann beim Zeichnen eine Beleuchtung eingeschaltet werden. Dies erfordert eine Normalenberechnung für die zu zeichnenden Dreiecke. Bei einem Dreieck mit Eckpunkten A, B und C ergibt sich die Normale aus dem Vektorprodukt der Richtungsvektoren  $x = \overrightarrow{AB}$  und  $y = \overrightarrow{AC}$ . Um die Flächen aus jeder Richtung betrachten zu können, wird Two-Sided-Lighting eingeschaltet. Dadurch muss auch die Richtung der Normalen nicht berücksichtigt werden.

### 5.3 Interaktionsmöglichkeiten

Die Ansicht des Volumens kann entweder gedreht, vergrößert oder verkleinert werden. Zusätzlich besteht die Möglichkeit die Beleuchtung an- bzw. auszuschalten. Eine ausführliche Beschreibung der Bedienung folgt im Abschnitt Programmstruktur.



## 6 Beispiele

Es werden einige Beispieldatensätze mit den berechneten Konturbäumen vorgestellt. Zusätzlich werden einige aus Teilbäumen selektierte Isoflächenkomponenten gezeigt. Die gewählten Beispiele repräsentieren einen manuell erzeugten Testdatensatz sowie zwei größere Anwendungen.

### 6.1 Testdatensatz

Als erstes Beispiel dient ein kleiner, manuell erzeugter Testdatensatz der Größe  $5 \times 5 \times 5$  mit Datenwerten zwischen 2 und 28. Dabei entsteht ein kleinerer Konturbaum, an dem aber Details und auftretende Probleme besser erkennbar sind.

Abbildung 33 zeigt den für diesen Datensatz berechneten Konturbaum im Punkte und Linien Zeichenmodus. Es sind eine Menge an Überschneidungen zu erkennen, die bei größeren Datensätzen zu Problemen bei der Übersichtlichkeit führen können.

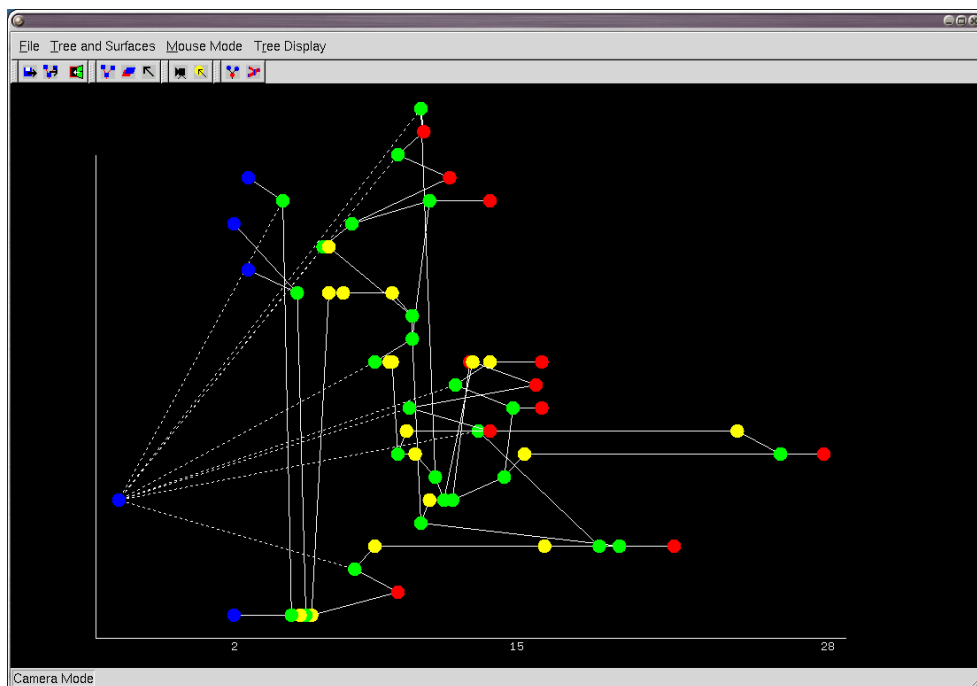


Abbildung 33: Testdatensatz: Konturbaum Punkte und Linien

Abbildung 34 zeigt denselben Konturbaum im Modus mit Volumendaten visualisiert. In diesem Modus treten die Probleme mit Überschneidungen noch stärker in Erscheinung. Die Volumendaten wurden skaliert, um eine zu große Ausdehnung des Baums in y-Richtung zu vermeiden.

Da der Testdatensatz keine besondere Struktur enthält ist eine Verfolgung von Konturen



Abbildung 34: Testdatensatz: Konturbaum mit Volumendaten

anhand von Screenshots nicht sinnvoll. Als Beispiel für eine Isofläche wird deshalb nur eine einzelne Abbildung verwendet (Abbildung 35). Es wurde eine Komponente zwischen zwei Sattelpunkten gewählt, Isowert: 11.

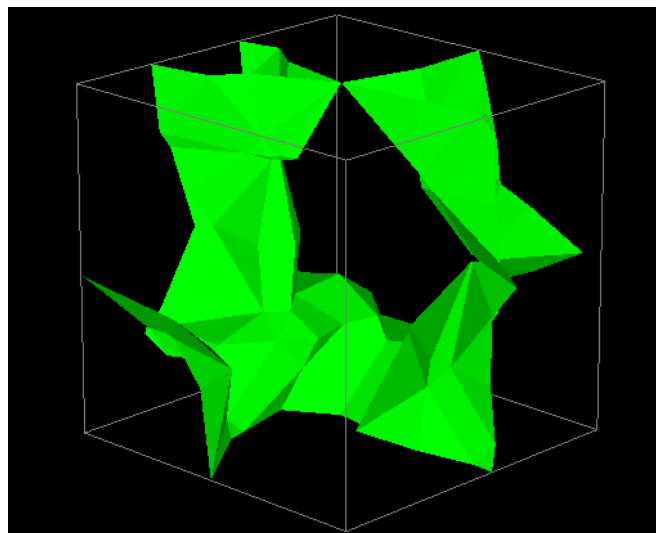


Abbildung 35: Testdatensatz: Isofläche

## 6.2 Koordinatenachsen

Abbildung 36 zeigt als nächsten behandelten Datensatz ein  $64 \times 64 \times 64$  großes Beispiel mit drei Achsen und zwei daran hängenden Kugeln.

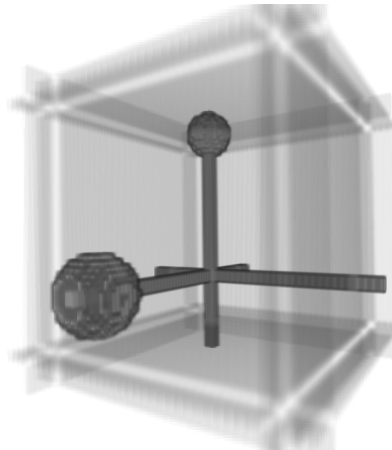


Abbildung 36: Achsendatensatz: Gesamtansicht

Es wurden zwei Konturbäume aus zwei je  $16 \times 16 \times 16$  großen Datenausschnitten berechnet. Diese beschränken sich auf den Bereich des Schnittpunktes der Achsen und den Bereich der großen Kugel. Wie in der Gesamtabbildung 36 zu erkennen ist, befinden sich im Volumen große Bereiche mit gleichen Datenwerten. Dies bedeutet durch die Datenaufbereitung wieder viele zusätzliche Sattelpunkte. Deshalb zeigen die Abbildungen 37 und 38 nur Ausschnitte des jeweiligen Konturbaums. Die Kanten zum absoluten Minimum wurden abgeschaltet, da sie nicht bei der Selektion benötigt wurden und nur die Übersichtlichkeit verschlechtern. Auf eine Berechnung der Volumenwerte wurde an dieser Stelle verzichtet. Die Dauer der Berechnung pro Konturbaum betrug 4:30 Minuten (Intel Pentium 4, 1.8 GHz).

Abbildung 37 zeigt den Ausschnitt, der sich auf den für die Achsen relevanten Teil beschränkt. In Abbildung 38 ist ein großer Ausschnitt des Konturbaums der Kugel zu sehen.

Für die Darstellung der Isoflächen wurde aus jedem Konturbaum ein Isowert selektiert. Abbildung 39 zeigt links die Isofläche von zwei der Achsen an ihrem Schnittpunkt. Rechts in Abbildung 39 ist eine Isoflächendarstellung der Kugel zu sehen. Hierbei ist gut das Loch in der Kugel zu erkennen, das durch die Achse entsteht, die die Kugel durchquert. Diese besitzt einen anderen Isowert als die Kugel und ist in der Selektion nicht enthalten.

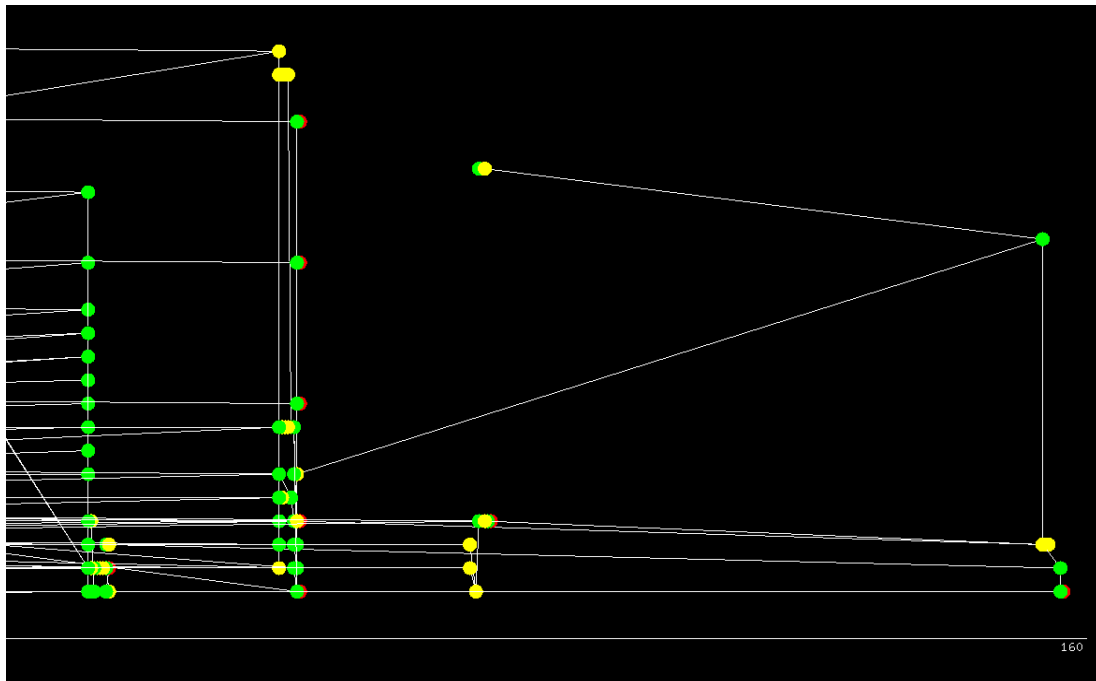


Abbildung 37: Achsendatensatz: Konturbaumausschnitt Achsenschnitt

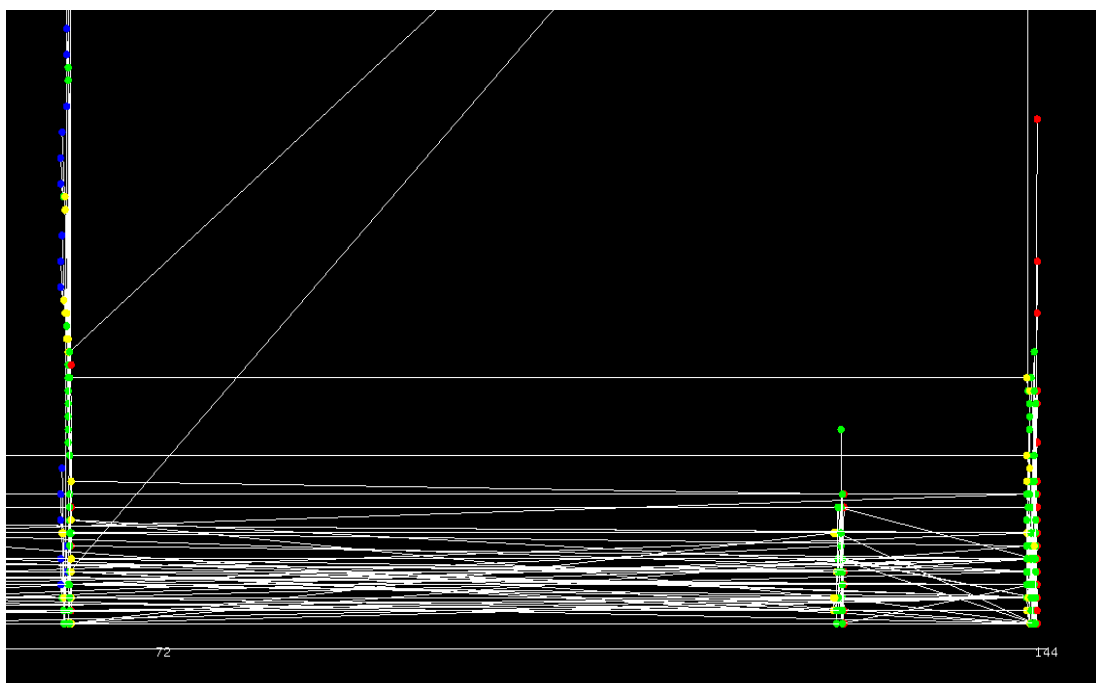


Abbildung 38: Achsendatensatz: Konturbaumausschnitt Kugel

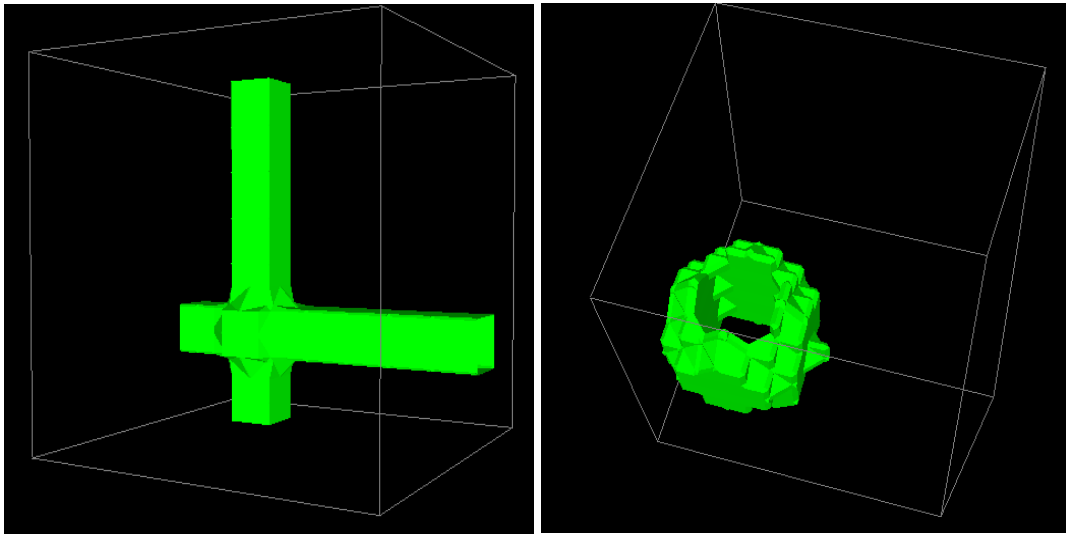


Abbildung 39: Achsendatensatz: Isoflächen

### 6.3 Kopfdatensatz

Als letztes Beispiel wird ein  $64 \times 64 \times 64$  großer Datensatz behandelt, der die Aufnahme eines menschlichen Kopfes darstellt. Abbildung 40 zeigt eine Visualisierung des gesamten Datensatzes.



Abbildung 40: Kopfdatensatz: Gesamtansicht

Es wurde eine Berechnung eines Konturbaums eines  $20 \times 20 \times 20$  Datenausschnitts durchgeführt. Aufgrund der Unübersichtlichkeit bei der Visualisierung im Volumenmodus wurde wieder auf die Berechnung der Volumen verzichtet. Dauer: 11 Minuten (Intel Pentium 4, 1.8 GHz). Der Ausschnitt beschränkt sich auf einen Teil der Kopfschale. Abbildung 41 zeigt den Konturbaum, ohne einen Teil des Randes. Durch die Vielzahl an Überschneidungen wird wieder die Notwendigkeit eines anderen Zeichenalgorithmus deutlich. Abbildung 42 zeigt eine selektierte Isofläche, die den Teil des Kopfrandes darstellt.

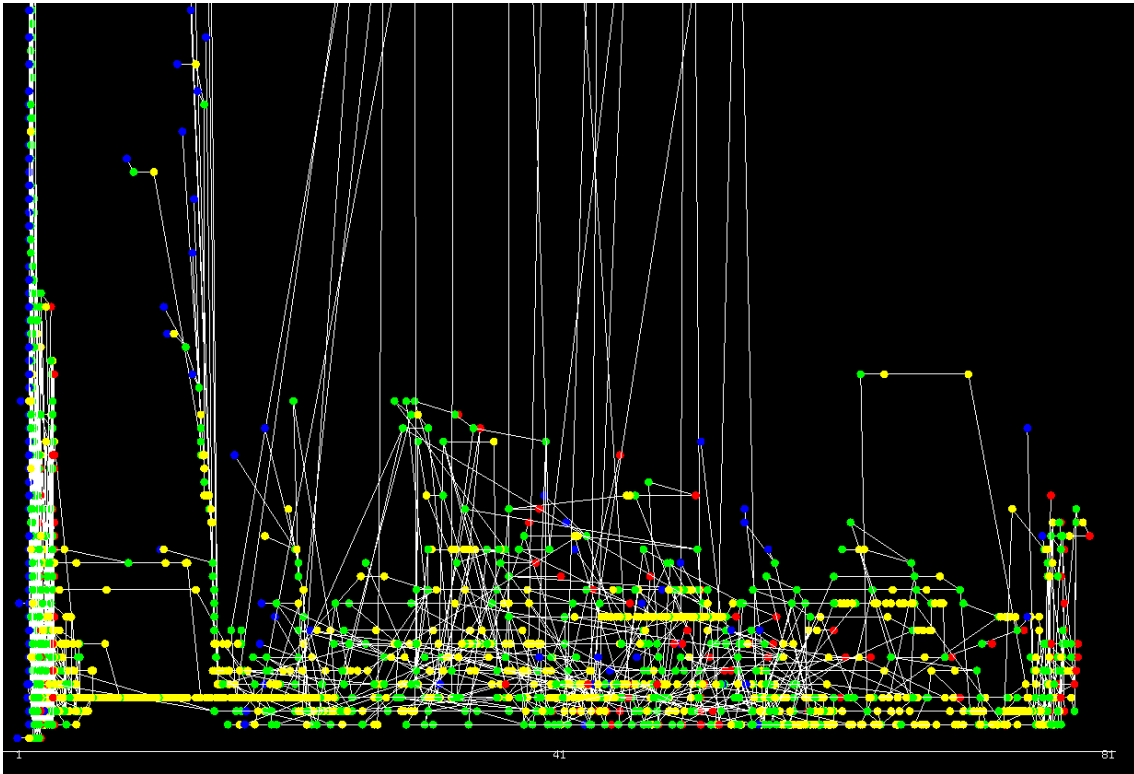


Abbildung 41: Kopfdatensatz 1: Konturbaum

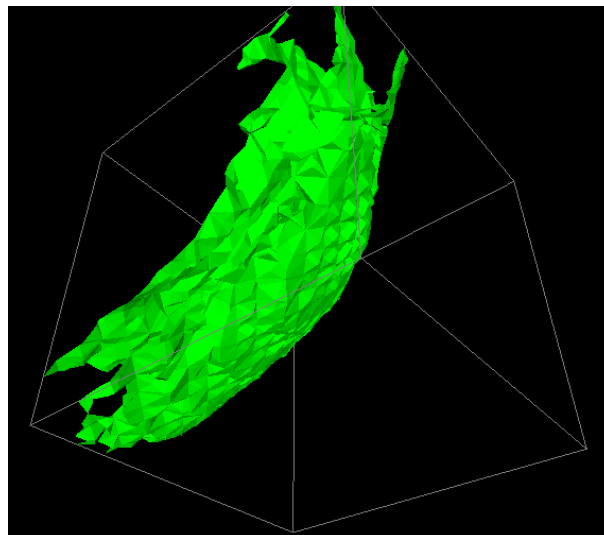


Abbildung 42: Kopfdatensatz 1: Isofläche

Zur Darstellung des Konturbaums im Volumenmodus wurde erneut die Berechnung des Baums eines diesmal 12x12x12 großen Datenausschnitts berechnet, jedoch weiter nach innen verschoben. Die Volumenberechnung wurde aktiviert. Dauer: 10 Minuten (Intel Pentium 4, 1.8 GHz).

Abbildung 43 zeigt zunächst den gesamten Konturbaum wieder im Punkte und Linien Modus visualisiert. Dagegen ist in Abbildung 44 nun einen Ausschnitt des Baums im Volumenmodus zu sehen. Es wurde erneut eine Skalierung (Verkleinerung) der Volumenwerte vorgenommen.

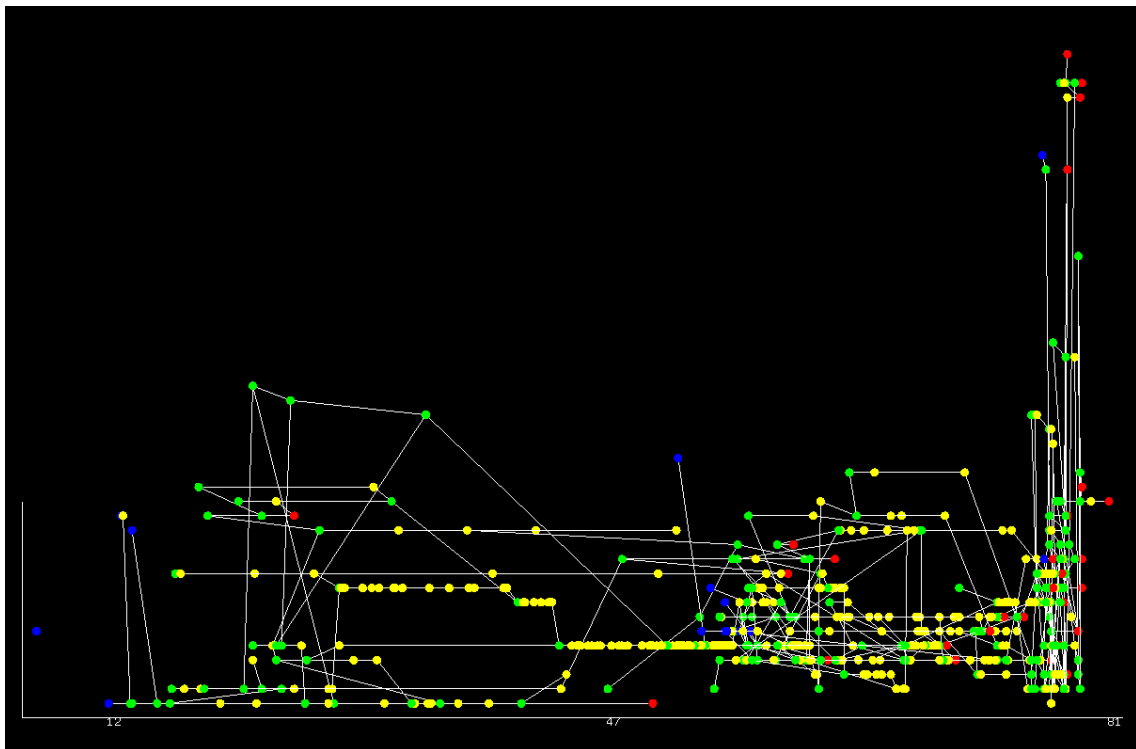


Abbildung 43: Kopfdatensatz 2: Konturbaum Punkte und Linien

Abbildung 45 zeigt noch einmal eine Isofläche, die aus dem Bereich selektiert wurde, der in Abbildung 44 zu sehen ist.

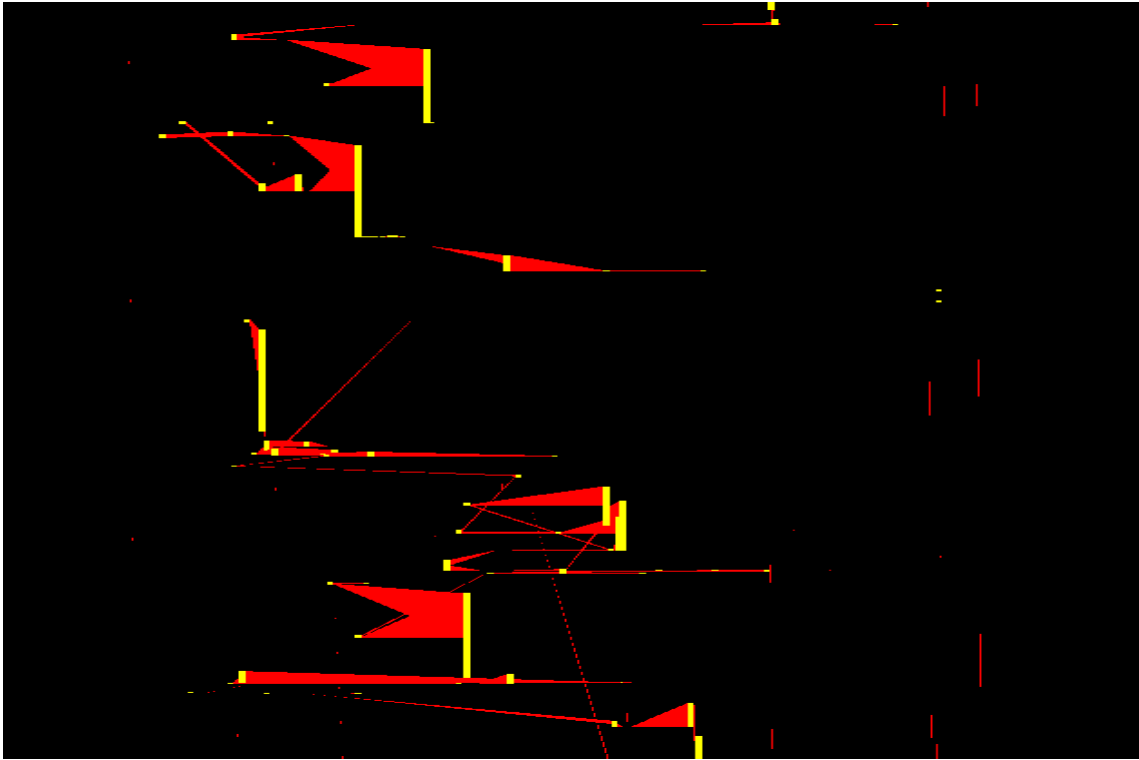


Abbildung 44: Kopfdatensatz 2: Konturbaum Volumenmodus

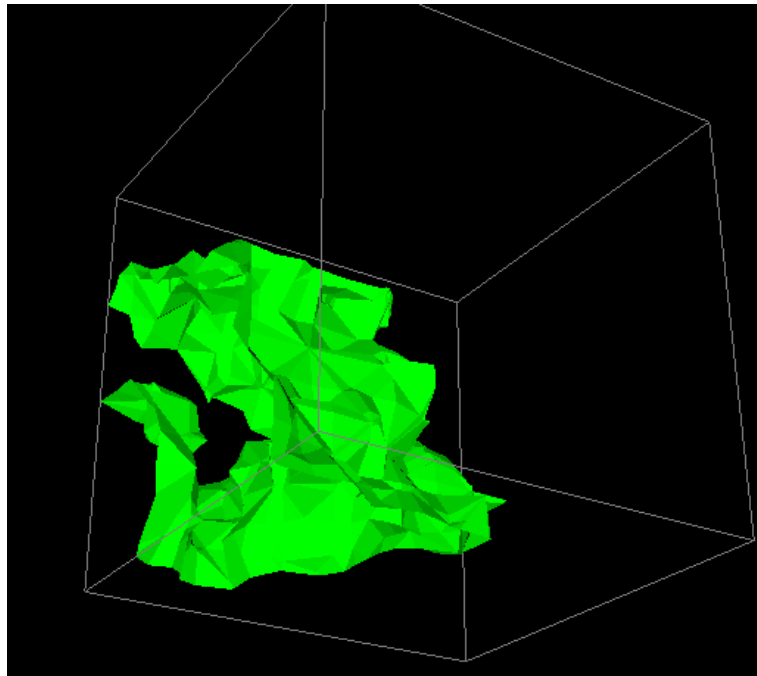


Abbildung 45: Kopfdatensatz 2: Isofläche

## 7 Erweiterungen und Anwendungen

### 7.1 Erweiterungen

#### Direktes Picking

Die Selektion eines Isowertes ist bis jetzt nur durch Picken einer Kante oder eines Knotens und anschließendes Auswählen des Isowertes in einer Dialogbox möglich. Der Rendervorgang der Isofläche wird ebenfalls über die Box gestartet. Um hier eine Verbesserung zu erreichen, kann ein direktes Picking des Isowertes eingeführt werden. Dabei wird durch die selektierte Position auf der Kante der Isowert direkt bestimmt. Hierfür wird aus der x-Position des Mauszeigers der Wert berechnet. Die Bestimmung der Startzelle kann ohne Probleme wie zuvor erfolgen. Der Rendervorgang wird dann sofort nach der Selektion gestartet. Bei der Selektion eines Knotens ist der Isowert schon direkt bestimmt und die Dialogbox kann ebenfalls umgangen werden.

#### Zeichenalgorithmus

Der bisher verwendete Algorithmus zur Visualisierung des Konturbaums verhindert keine Überschneidungen von Kanten, da für die Bestimmung der y-Position keine besonderen Vorkehrungen getroffen wurden um dies komplett zu vermeiden. Es wird lediglich durch die Erhöhung der y-Position an ausgewählten Stellen eine Reduzierung der Anzahl der Überschneidungen erreicht. Zur deutlicheren Darstellung kann für das Zeichnen des Konturbaums ein komplizierterer Algorithmus verwendet werden, von denen hier einige kurz vorgestellt werden.

Dafür bietet [30] einen allgemeinen Einblick in die Möglichkeiten der Visualisierung von und der Navigation in Graphen und kann als Ausgangspunkt bei der Suche nach Zeichenalgorithmen für Graphen benutzt werden. Es erfolgt eine Vorstellung von Algorithmen für das Zeichnen und das Layout von Graphen und Bäumen. Dabei erfolgt hier eine Beschränkung auf 2D Varianten, während [30] darüber hinaus geht.

Abbildung 46 zeigt einen Zeichenalgorithmus für Bäume in einem klassischen Tree-Layout. Er erfüllt einige für die Visualisierung von Bäumen interessante Bedingungen, wie gerade Kanten, keine Überschneidungen und Knoten mit gleicher Tiefe auf gleicher horizontaler Ebene.

HTree Layouts (Abbildung 47, oben links) werden vorzugsweise für Binärbäume verwendet. Bei der radialen Positionierung der Knoten (Abbildung 47, oben rechts) werden die Knoten auf konzentrischen Kreisen platziert. Teilbäume werden dann auf sich nicht überschneidende Sektoren der Kreise aufgeteilt. Das Cone Tree Layout hängt die Teilbäume als Kreise an den jeweiligen Vorgängerknoten an (Abbildung 47, unten links). Ein weiteres Layout, das über die bisherigen traditionellen Layouts hinausgeht, sind die Tree-Maps (Abbildung 47, unten rechts). Der Baum wird repräsentiert durch sich einschließende Rechtecke mit bestimmter Größe. Dies wird zum Beispiel häufig bei der Darstellung der Hierarchie eines Dateisystems benutzt.

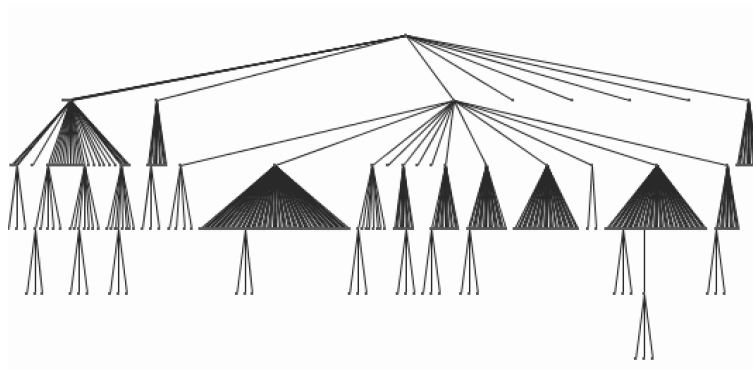


Abbildung 46: Klassisches Tree Layout

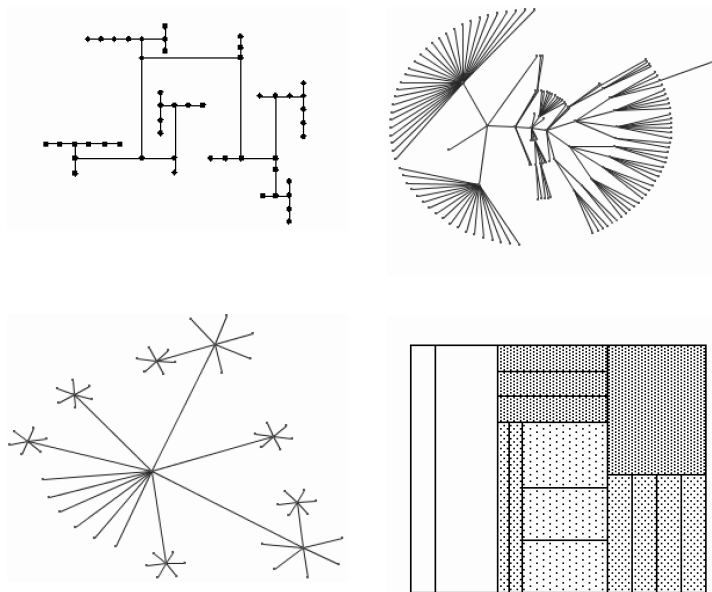


Abbildung 47: Darstellungsmöglichkeiten für Graphen: oben: HTrees, Radial Trees, unten: Cone Trees, Tree-Maps

Ein Beispiel für gerichtete Graphen ist das Sugiyama Layout. Es erfolgt eine Einteilung der Knoten in Schichten und eine Platzierung der Knoten innerhalb dieser Schichten. Eine weitere Möglichkeit für Graphen sind nicht deterministische Vorgehensweisen (Spring Layouts). Da die bisherigen Algorithmen bei großen Graphen Probleme bekommen, ist oft die Vorberechnung eines Spanning Trees empfehlenswert.

### Datenaufbereitung für andere Datenformate

Die verwendete Datenaufbereitung zur Vermeidung von Nachbarn mit gleichen Datenwerten kann nur bei ganzzahligen Werten an Gitterpunkten benutzt werden. Bei anderen Datenformaten, zum Beispiel Floats, muss dies geändert werden, um eine Manipulation der Konturen zu verhindern. Um zu vermeiden, dass durch die Änderung der Datenwerte, die zuvor kleiner als einer ihrer Nachbarn waren, plötzlich größer werden, muss der

kleinste Abstand ( $> 0$ ) zwischen je zwei benachbarten Gitterpunkten bestimmt werden. Anhand dieses Abstandes kann dann ein neues Schema zur Datenänderung entworfen werden. Abbildung 48 zeigt ein mögliches neues Schema.  $min$  ist dabei der bestimmte minimale Abstand. Die Werte an den Würfeckpunkten werden jeweils an den Vertices addiert.

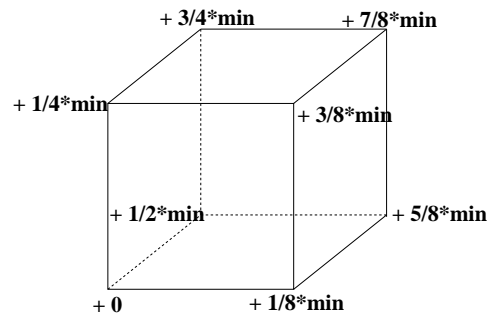


Abbildung 48: Geänderte Datenaufbereitung

### Verfolgung einer Kontur in der Isoflächendarstellung

Als zusätzliche Interaktionsmöglichkeit auf der Ebene der Isoflächendarstellung kann eine Möglichkeit zur Verfolgung einer Flächenkomponente entworfen werden. Nach Selektion einer Isoflächenkomponente kann diese durch sequentielles Erhöhen bzw. Erniedrigen des Isowertes durch Tasteninteraktion verfolgt werden. Die Schrittweite kann dabei veränderbar gemacht werden. Beim Aufspalten der Komponente kann gewählt werden, ob nur eine oder mehrere Konturen weiter verfolgt werden sollen. Durch diese Erweiterung kann die Entwicklung einer Kontur genauer beobachtet werden.

### Selektion von Teilbäumen

Um mehrere Flächen gleichzeitig darstellen zu können, kann eine Möglichkeit zur Selektion eines Teilbaumes oder mehrerer Baumkomponenten hinzugefügt werden. Der Teilbaum bzw. die Komponenten können durch eine Bounding Box oder das Auswählen der einzelnen Komponente nacheinander bestimmt werden. Für jeden darin enthaltenen Knoten bzw. Kante wird eine teilweise transparente Fläche gezeichnet. Für eine Kante kann als Isowert das arithmetische Mittel der Werte der beiden zugehörigen Knoten verwendet werden.

### Zuordnung Teilbaum $\leftrightarrow$ Volumenkomponente

Die angesprochene Selektionsmöglichkeit von Teilbäumen ermöglicht eine weitere Ausnutzung des Konturbaums. Durch ihn kann eine Zuordnung gewisser Stellen des Volumens zu einem Teilbaum erfolgen. Teilvolumen, zum Beispiel das Gehirn in einem Schädel, können so über einen Teilbaum identifiziert und selektiert werden. Der Benutzer wird dadurch bei der Suche nach bestimmten Stellen unterstützt und besitzt durch den Teilbaum die Information, in welchem Teil des Volumens er sich befindet.

### Invarianz des Baums

Der Konturbaum bietet den Vorteil der Invarianz des Baums gegenüber geringfügigen Datenänderungen. Sollte das Datenvolumen an einzelnen Stellen manipuliert werden, zum Beispiel eine Änderung der Meßfehler (Rauschen) oder die Verformung eines Teilvolumens, ergibt sich kein kompletter Neuaufbau des gesamten Konturbaums, sondern nur eine lokale Änderung. Nur im Teilbaum, der sich auf den geänderten Datenbereich bezieht, ergibt sich eine Modifikation der Struktur. Dies kann bei einer Neuberechnung des Baums ausgenutzt werden, in dem nur der Teilbereich der Daten und die zugehörigen Baumkomponenten betrachtet werden.

## 7.2 Anwendungen

### Seed Sets

Ein Saatmenge (Seed Set) ist eine Teilmenge der Gitterzellen, die als Startzellen zur Verfolgung von Konturen verwendet werden können. Eine Saatmenge ist vollständig, wenn alle Konturen mindestens eine Saatzelle enthalten. Diese Menge sollte möglichst klein und trotzdem noch vollständig sein. In [1] werden zwei Methoden zur Bestimmung einer solchen Saatmenge vorgestellt. Als Ausgangspunkt dient der berechnete Konturbaum des Beispiels auf Seite 13.

Die erste Methode berechnet eine minimale Saatmenge in  $O(n^2 \log n)$ . Es wird eine Reduktion auf ein Flussproblem durchgeführt. Zuerst wird ein erweiterter Graph mit einer Quelle und einem Abfluss aufgebaut. Von der Quelle wird zu jedem Maximum und jeder Gabelung ein Segment eingefügt. Ebenso von jedem Minimum und jeder Vereinigung zum Abfluß. Abbildung 49 links zeigt den entstandenen Graph, rechts die Kurznotation des Graphen mit + und - statt Quelle und Abfluss.

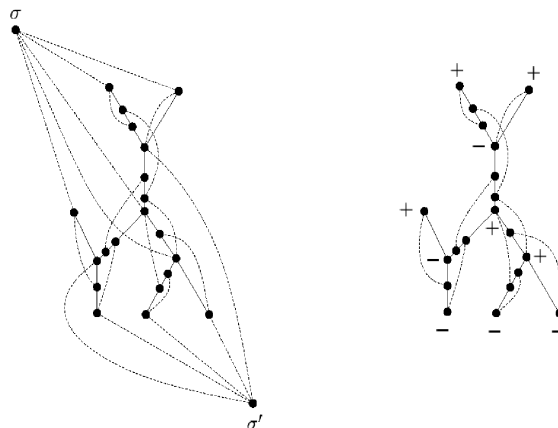


Abbildung 49: Erweiterter Graph mit Quelle und Abfluss

Die Knoten des Graphen sind nach Höhe (Wert) ihrer entsprechenden Gitterpunkte sortiert. Die neuen Segmente und die vorhandenen Kanten des Graphen sind gerichtet. Seg-

mente abwärts von höheren zu niedrigeren Knoten (gepunktet in der Abbildung) und Kanten aufwärts. Die Quelle ist höchster und der Abfluss niedrigster Punkt. Segmente haben Kapazität und Kosten 1, Kanten haben Kapazität  $\infty$  und Kosten 0. Verbindungen zu Quelle und Abfluss haben Kapazität 1 aber Kosten 0. Aus der Graphentheorie ist bekannt: Für jeden Baum entspricht die Anzahl der Maxima und Gabelungen der Anzahl der Minima und Vereinigungen. Man betrachte nun das Flussproblem (Finde einen Fluss von Quelle zu Abfluss mit bestimmten Wert  $f$ ) und dasselbe Problem für einen Fluss mit minimalem Wert ( $f$ ). Eine Lösung der Probleme besteht aus einer Zuweisung eines Flusses für jedes Segment/Kante. Es gilt: Jede Lösung eines Flussproblems mit minimalen Kosten entspricht der einer minimalen Saatmenge. Der Beweis kann ausführlich in [1] nachgelesen werden. Um eine Lösung des Flussproblems zu bestimmen, kann zum Beispiel ein Shortest Path Algorithmus verwendet werden.

In der zweiten Methode wird der Aufwand auf  $O(n \log^2 n)$  in jeder Dimension reduziert. Dafür erhält man nicht die minimale Saatmenge, aber wird dabei nicht größer als doppelt so groß wie die minimale Menge. Es erfolgt zunächst wieder die Beschränkung auf vier Typen für die kritischen Punkte (Min, Max, Vereinigung, Gabelung). Diese Beschränkung kann jedoch später wieder beseitigt werden. Ausgangspunkt ist wieder der zuvor bestimmte Konturbaum. Jeder Knoten im Baum erhält zwei zusätzliche Integers, deren Zuweisung durch eine Durchnummerierung der Knoten von links nach rechts (erster Wert) und dann von rechts nach links (zweiter Wert) erfolgt. Für zwei Knoten  $u$  und  $v$  gilt dann:  $u$  ist Vorgänger von  $v$ , genau dann wenn beide Integer von  $u$  kleiner sind als die von  $v$ . Dies ermöglicht eine leichtere und schnellere Vorgängerbestimmung. Es ergibt sich der Baum mit der Nummerierung aus Abbildung 50.

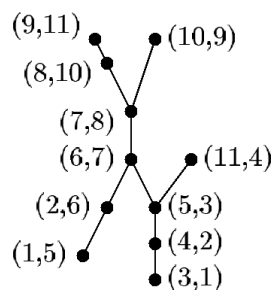


Abbildung 50: Baum mit Durchnummerierung der Knoten

Dann erfolgt wieder das Einfügen der Segmente. Man erhält den Graphen  $G$ . Über ein erneutes Durchlaufen des Volumens und eine simultane Verfolgung im Graph wird die Saatmenge bestimmt. Bei jedem aktuell bearbeiteten Knoten: Werden der Pfeil zum Knoten hin und der vom Knoten weg bereits durch die vorhandene Saatmenge abgedeckt, dann ist keine neue Saatzelle notwendig, wenn nicht, dann wird eine neue Saatzelle hinzugefügt, die einen möglichst großen Bereich abdeckt. Die genaue Handhabung der einzelnen Knotentypen und die erforderlichen Datenstrukturen können wieder in [1] nachgelesen werden.

### Topologiedarstellung

In [29] erfolgt ebenfalls eine Auseinandersetzung mit dem Problem der Herausarbeitung von für den Benutzer interessanten Datenbereichen aus Skalarfeldern. Hierfür werden die kritischen Punkte (lokale Maxima/Minima, Sattelpunkte) des Skalarfeldes bestimmt und diese durch bestimmte kritische Kurven verbunden (siehe Abbildung 51, Konturen (gepunktet) eines Skalarfeldes mit kritischen Punkten und den sie verbindenden Kurven).

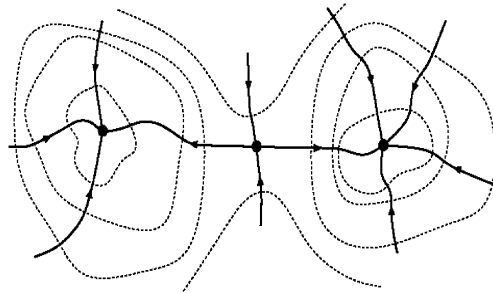


Abbildung 51: Kritische Punkte verbunden durch kritische Kurven

Kritische Kurven sind Kurven, die an allen Stellen tangential zum Gradientenfeld des Skalarfeldes verlaufen. Diese Kurven unterteilen das Feld in Bereiche, in denen sich der Gradient einheitlich verhält. Die Analyse der Topologie erfolgt in drei Schritten: Identifizierung der kritischen Punkte, Klassifizierung der kritischen Punkte und Berechnung der kritischen Kurven. Zunächst muss jedoch ein Modell für ein Skalarfeld mit stetigem Gradienten erstellt werden, um die kritischen Kurven überhaupt sinnvoll verfolgen zu können. Dabei sollen kritische Punkte nicht verloren gehen und möglichst wenig Punkte neu entstehen.

### Contour Spectrum

Im Bereich der Visualisierung mit Isoflächen ist vor allem die Suche nach dem für den Benutzer „richtigen“ Isowert ein aufwändiges Problem. In [28] wird eine Benutzeroberfläche vorgestellt, die den Anwender bei der Suche nach für ihn interessanten Volumenbereichen/Isoflächen unterstützt. Dabei werden nach der Auswahl eines Isowertes charakteristische Daten der zugehörigen Isofläche berechnet und interaktiv angezeigt. Diese Daten sind:

*Länge (2D) bzw. Fläche (3D) einer Kontur:*

In einem 2D Skalarfeld, zerlegt in Dreiecke, wird die Länge einer in einem Dreieck verlaufenden Kontur über eine B-Spline Funktion berechnet. Ebenso kann in 3D die Fläche einer Kontur in einem Dreieck bestimmt werden.

*Fläche (2D) bzw. Volumen (3D):*

Nach der Berechnung der Länge (Fläche) einer Kontur kann durch Integration auch die eingeschlossene Fläche (das Volumen) bestimmt werden.

*Gradient:*

Da der Gradient einer Kontur im Allgemeinen nicht konstant ist, wird eine Art der Gradientenbestimmung für eine Kontur eingeführt. In jedem Dreieck ist zuvor die Länge der Kontur berechnet worden. Ebenso kann ein Gradient für dieses Konturstück interpoliert werden. Um den Beitrag dieses Gradienten für die gesamte Kontur zu erhalten, wird dieser einfach mit der Länge des Konturstücks multipliziert. Die Summe über alle Konturteile ergibt den gesuchten konstanten Gradienten. Dies kann genauso in Tetraederzellen (3D) erfolgen.

*Konturbaum:*

Da die bisherigen Daten keine Informationen über die Struktur der Daten enthalten, wird auch hier der Konturbaum zur Darstellung der Topologie des Volumens benutzt. Die Berechnung des Baums erfolgt wie in [1].

**Level Sets**

Level Set Methoden segmentieren Strukturen in Volumendaten. Es wird eine Veränderung von Isoflächen ermöglicht, um interessante Bereiche zu extrahieren.

In [26] werden die zwei Hauptabschnitte eines Frameworks zur Benutzung von Level Sets beschrieben (Abbildung 52).

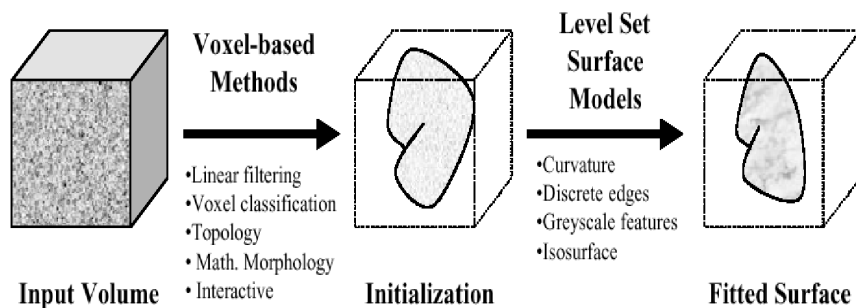


Abbildung 52: Framework für Level Sets

Der erste Abschnitt ist die Initialisierung um eine Startmöglichkeit für die Oberflächenanpassung zu erhalten. Die Initialisierung kann automatisch über eine lineare Filterung, Voxel Klassifizierung, topologische/logische Voxeloperationen (z.B. Vereinigung oder Schnitt von Voxelgruppen) und morphologische Filterung (durch eine partielle Differentialgleichung) erfolgen. Wenn keine automatische Lösung akzeptabel ist, kann diese auch manuell erzeugt werden. Dies erfolgt über ein Constructive Solid Geometry (CSG) Modell in Verbindung mit einem Marching Cubes Gitter. Nach einer Konvertierung in ein binäres Volumen wird eine Isofläche generiert, die das Startmodell für den nächsten Abschnitt ergibt. Die Oberflächenänderung des Level Sets ist der zweite Abschnitt. Die Initialisierung sollte nahe an der gesuchten Lösung sein. Es erfolgt eine letzte Anpassung der Fläche an die Lösung durch unterschiedliche Operatoren.

Im medizinischen Bereich verwendet [27] level-set surface segmentation zur Erkennung

von Verschiebungen der Gehirnmasse zur Unterstützung von bildgesteuerten Operationen. Dies geschieht in mehreren Schritten. Zunächst erfolgt als erster Schritt die Berechnung der neuen Position der beobachteten Gehirnmasse. Hierfür wird eine level-set Methode verwendet. Dann werden die 3D Koordinaten der sichtbaren Oberfläche durch lasergesteuerte Flächenabtastung bestimmt. Anschließend wird die Verschiebung der Oberfläche abgeschätzt.

## 8 Zusammenfassung

Diese Studienarbeit stellt ein Verfahren zur direkten Visualisierung von Volumendaten vor. Dabei wurde zunächst ein allgemeiner Einblick in den Bereich der Volumenvisualisierung gegeben und einige Verfahren zur direkten und indirekten Volumenvisualisierung vorgestellt. Zusätzlich wurde unter dem Gesichtspunkt der Visualisierung die Topologie der Volumendaten betrachtet.

Erster Hauptpunkt ist jedoch die Entwicklung eines Verfahrens zur Berechnung eines Konturbaums (Contour Tree) aus der Topologie eines 3D-Skalarfeldes. Hierfür wurde ein bereits bekanntes Verfahren leicht verändert verwendet. Bei der Bestimmung des Konturbaums wurden bereits die notwendigen Voraussetzungen für das zweite Ziel, die Visualisierung von Isoflächenkomponenten, berücksichtigt. Aus einer Darstellung des Baumes heraus, wurde eine Möglichkeit zur gezielten Selektion von Komponenten einer Fläche zu einem bestimmten Isowert ermöglicht. Diese Flächenkomponente kann dann getrennt visualisiert und betrachtet werden.

Zusätzlich wird anschließend noch ein Ausblick auf die Erweiterungsmöglichkeiten dieser Arbeit gegeben und einige Anwendungsgebiete des Konturbaums und der Topologie eines Skalarfeldes vorgestellt.



## A Programmstruktur

### A.1 Hauptfenster

Das Hauptfenster stellt sämtliche Menüpunkte zur Handhabung der Dialoge und zur Interaktion mit dem Konturbaum zur Verfügung. Alle Menüpunkte können auch über ein entsprechendes Icon direkt aktiviert werden. Die Beschreibung der Icons folgt im Abschnitt *Icons*. Die Abbildungen 53 bis 56 zeigen alle vorhandenen Menüs.

Das Menü *File*

*Load Data*: Ermöglicht das Laden der Volumendaten für die Visualisierung des Konturbaums.

*Load Contour Tree*: Ermöglicht das Laden des Konturbaums. Wird erst aktiv (aufrufbar), wenn die Volumendaten bereits geladen wurden.

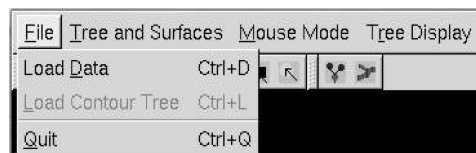


Abbildung 53: File Menü

Das Menü *Tree and Surfaces*

*Contour Tree Dialog*: Öffnet den Dialog zur Steuerung der Berechnung des Konturbaums für einen Volumendatensatz. Beschreibung des Dialogs in Abschnitt A.1.2.

*Show Surface Dialog*: Öffnet das Fenster zur Darstellung der Isoflächen. Beschreibung des Fensters in Abschnitt A.2.

*Show Picking Dialog*: Öffnet den Dialog zur Isowertselektion im Konturbaum. Beschreibung des Dialogs in Abschnitt A.1.1.

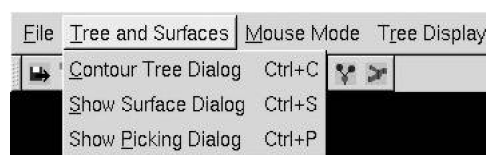


Abbildung 54: Tree and Surfaces Menü

Das Menü *Mouse Mode*

Ermöglicht ein Umschalten zwischen den beiden vorhandenen Modi der Maus Interaktion für den Konturbaum. Der aktuelle Status wird am unteren linken Rand des Fensters angezeigt.

*Adjust Camera*: Ermöglicht eine Veränderung der Ansicht auf den Konturbaum.

*Pick Objects*: Ermöglicht die Selektion von Knoten oder Kanten für die Visualisierung der Flächen.



Abbildung 55: Mouse Mode Menü

Das Menü *Tree Display*

Umschalten zwischen den zwei Arten der Visualisierung des Konturbaums.

*Points and Lines*: Visualisierung mit Knoten und Kanten.

*Volumes*: Visualisierung mit Volumendaten.

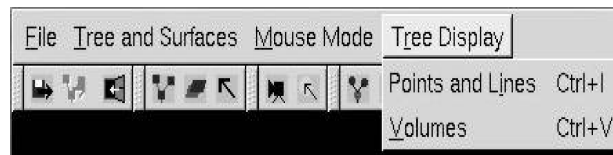


Abbildung 56: Tree Display Menü

Abbildung 57 zeigt einen kompletten Screenshot des Hauptfensters mit Visualisierung eines Konturbaums im Knoten und Kanten Modus.

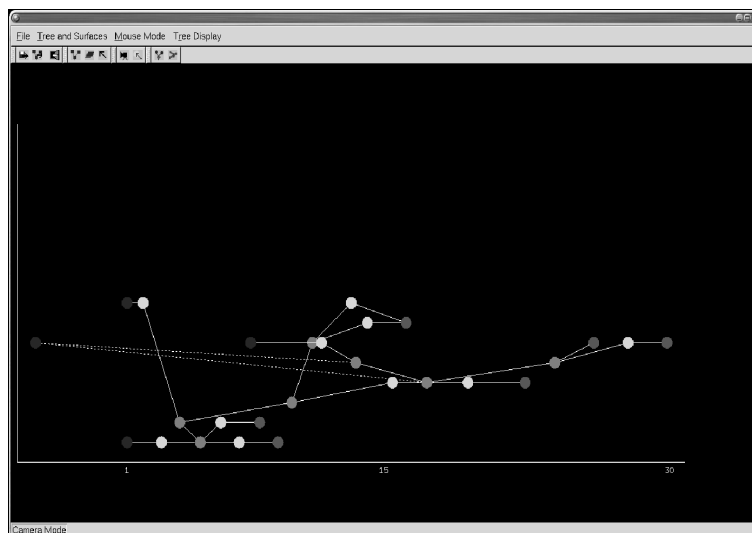


Abbildung 57: Beispiel für Visualisierung mit Knoten und Kanten

Ein Screenshot des Hauptfenster mit einem Konturbaum, der im Volumen Modus gezeichnet wurde, ist in Abbildung 58 dargestellt.

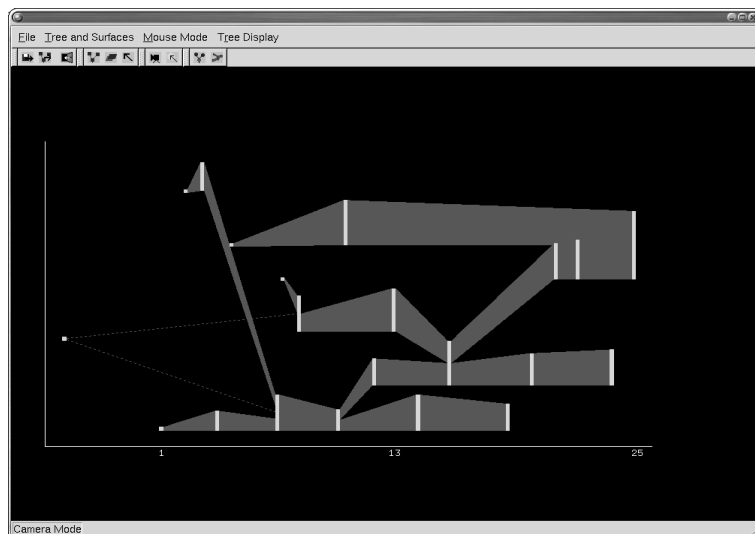


Abbildung 58: Beispiel für Visualisierung mit Volumendaten

### A.1.1 Isowertselektion / Picking Dialog

Zur gezielten Selektion von Isowerten und Flächenkomponenten können im Konturbaum Kanten oder Knoten selektiert werden. Dafür muss zunächst in den Selektionsmodus umgeschaltet werden. Anschließend kann durch einfaches Klicken auf einen Knoten oder eine Kante diese selektiert werden. Die gewählten Werte werden in einem eigenen Dialogfenster angezeigt. Bei einem Knoten wird der Typ und der entsprechende Datenwert des Knotens angezeigt (Abbildung 59, links). Dieser Wert ist hier nicht veränderbar. Nach Selektion einer Kante kann ein Isowert aus einem Bereich ausgewählt werden (Abbildung 59, rechts). Die Grenzen dieses Bereichs sind gegeben durch die Datenwerte der die Kante begrenzenden Knoten und werden in zwei Feldern angezeigt. Der Wert kann entweder über einen Scrollbar ganzzahlig oder durch Eingabe in das Textfeld mit Nachkommastellen gewählt werden.

Durch Drücken des Render-Buttons werden die gewählten Werte übernommen und die Isoflächenkomponente wird gezeichnet. Der Close-Button schließt den Dialog.

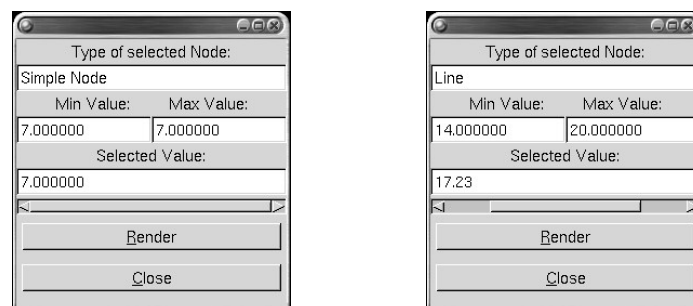


Abbildung 59: Dialoge zur Selektion

### A.1.2 Konturbaum Dialog

Die Berechnung des Konturbaums wird über einen eigenen Dialog gesteuert.

Über den Button *Load Data* wird die die Skalarwerten enthaltende Datei ausgewählt und geladen. Die Beschreibung des Datenformats erfolgte bereits in einem vorherigen Abschnitt. Erst nach dem Laden kann die Berechnung des Konturbaums gestartet werden.

Der Button *Compute Contour Tree* startet die Berechnung des Konturbaums. Nach Beendigung dieses Vorgangs kann der berechnete Baum gespeichert werden.

Zur Speicherung des Baums wird über *Save Contour Tree* die Speicherdatei ausgewählt. Das Dateiformat wurde bereits in einem vorherigen Abschnitt genauer beschrieben.

Der Button *Close Window* schließt den Dialog.

Über die Check Box kann die Berechnung der Volumendaten für jeden Knoten des Konturbaums aktiviert/deaktiviert werden.

Unter dem Feld *Actual Loaded Data File* wird der Pfad und der Dateiname der gerade geladenen Datei angezeigt.

Im Textfeld unter *Current Action* werden die erfolgten bzw. gerade laufenden Vorgänge mitprotokolliert.

Abbildung 60 zeigt den Konturbaum Dialog vor Beginn der Berechnung des Konturbaums.

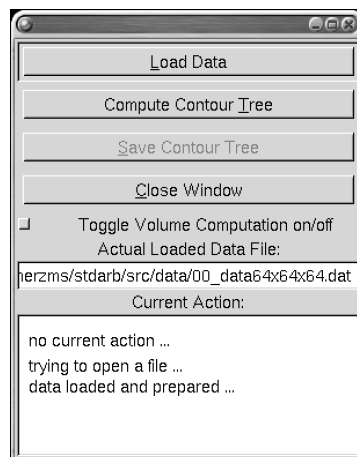


Abbildung 60: Konturbaum Dialog

### A.1.3 Interaktionsmöglichkeiten - Konturbaum

Bei der Darstellung des Konturbaums stehen folgende Interaktionsmöglichkeiten zur Verfügung:

Maus-Interaktion im Adjust Camera Modus

*linke Maustaste*: Bei Bewegung mit gedrückter linker Maustaste verschiebt sich die Ansicht des Baums je nach Bewegungsrichtung in x- oder y-Richtung.

*mittlere Maustaste*: Bei Bewegung mit gedrückter mittlerer Maustaste erfolgt eine Vergrößerung/Verkleinerung der Ansicht des Baums. Bewegung nach oben: Vergrößerung, nach unten: Verkleinerung

*rechte Maustaste*: keine Belegung

Maus-Interaktion im Pick Objects Modus

*linke Maustaste*: Selektion, Beschreibung im vorherigen Abschnitt.

*mittlere Maustaste*: keine Belegung

*rechte Maustaste*: keine Belegung

*Taste L*: An-/Abschalten des Zeichnens der Kanten zum absoluten Minimum

*Tasten +/-*: Skalierung (+: Verkleinerung, -: Vergrößerung) der y-Komponente im Volumendaten Modus.

## A.2 Darstellung der Isoflächen

Die selektierten Isoflächen werden in einem eigenen Fenster gezeichnet. Abbildung 61 zeigt ein solches Fenster.

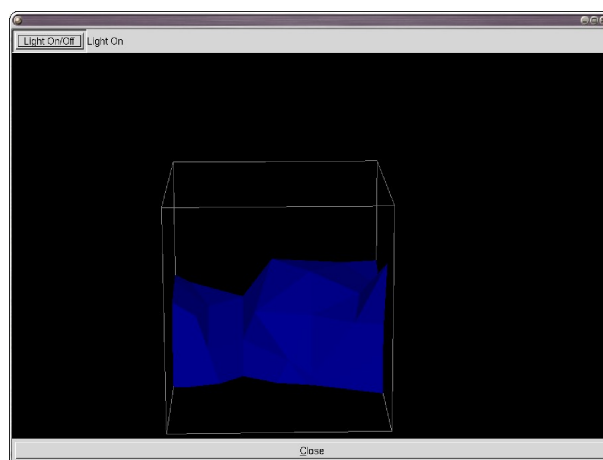


Abbildung 61: Fenster zur Isoflächendarstellung

Der Button *Light On/Off* schaltet die Beleuchtung ein/aus. Der aktuelle Status wird neben dem Button angezeigt. Zu Beginn ist die Beleuchtung ausgeschaltet.

Der Close-Button schließt das Fenster.

## Interaktionsmöglichkeiten - Isoflächen

Maus-Interaktion:

*linke Maustaste:* Bei Bewegung mit gedrückter linker Maustaste dreht sich die Ansicht des Volumens je nach Bewegungsrichtung um die x- oder y- Achse.

*mittlere Maustaste:* Bei Bewegung mit gedrückter mittlerer Maustaste erfolgt eine Vergrößerung/Verkleinerung der Ansicht des Volumens. Bewegung nach oben: Vergrößerung, nach unten: Verkleinerung

*rechte Maustaste:* keine Belegung

### A.3 Icons

Alle Aktionen der Menüpunkte des Hauptfensters können auch über ein Icon aktiviert werden. Die folgende Aufstellung zeigt die Icons und ihre Funktion.

Laden/Programmende (*File*)



Laden der Volumendaten (*Load Data*).



Laden des Konturbaums (*Load Contour Tree*).



Programm Beenden (*Exit*).

Dialoge öffnen (*Tree and Surfaces*)



Konturbaum berechnen (*Contour Tree Dialog*).



Selektionsdialog öffnen (*Show Picking Dialog*).



Isoflächenfenster öffnen (*Show Surfaces*).

Maus-Interaktion: Konturbaum (*Mouse Mode*)



Umschalten in Kamera Modus (*Adjust Camera*).



Umschalten in Selektionsmodus (*Pick Objects*).

Visualisierung des Konturbaums (*Tree Display*)



Knoten und Kanten (*Points and Lines*).



mit Volumenwerten (*Volumes*).

## Abbildungsverzeichnis

1	Contour Tracking . . . . .	6
2	Opaque Cube/Cuberille . . . . .	6
3	15 Fälle des Marching Cubes . . . . .	7
4	Raycasting bei beliebig orientierten Volumen . . . . .	8
5	V-Buffer: Unterteilung in spans . . . . .	9
6	Splatting . . . . .	9
7	Shear Warp für parallele und perspektivische Projektion . . . . .	10
8	Ellipsoid/Torus und zugehörige Reeb Graphen . . . . .	11
9	Metatorus und zugehöriger Hyper Reeb Graph . . . . .	11
10	Beispiel Konturbaum, 2D . . . . .	13
11	Beispiel Konturbaum, 3D, Flächen . . . . .	14
12	Beispiel Konturbaum, 3D, Baum . . . . .	14
13	Beispiel Konturbaum, 3D, Baum mit regulären Knoten . . . . .	15
14	Additionswerte an den Vertizes . . . . .	16
15	Zerlegung eines Würfels in fünf Tetraeder . . . . .	17
16	Baumstruktur . . . . .	18
17	Durchlaufen des Volumens . . . . .	19
18	Nachbarschaftsverhältnisse . . . . .	20
19	Nachbarschaftsgraph . . . . .	20
20	Sattelpunkt in 3D . . . . .	21
21	Ein Sattelpunkt bestehend aus mehreren Knoten . . . . .	22
22	Baumvereinfachung, Beispiel in 2D . . . . .	23
23	Fälle 1-5 des Marching Tetrahedra . . . . .	25
24	Knoten und Kanten: Behandlung des absoluten Minimums . . . . .	30
25	Knoten und Kanten: Behandlung eines regulären Knotens . . . . .	31
26	Knoten und Kanten: Behandlung einer Vereinigung/Gabelung . . . . .	31
27	Volumen: Behandlung des absoluten Minimums . . . . .	32
28	Volumen: Behandlung eines regulären Knotens . . . . .	32
29	Volumen: Behandlung einer Gabelung/Vereinigung . . . . .	33
30	Volumen: Behandlung eines Sattelpunktes . . . . .	33
31	Schnittpunktinterpolation . . . . .	36
32	Isoflächen beim Marching Tetrahedra . . . . .	37
33	Testdatensatz: Konturbaum Punkte und Linien . . . . .	39
34	Testdatensatz: Konturbaum mit Volumendaten . . . . .	40
35	Testdatensatz: Isofläche . . . . .	40
36	Achsendatensatz: Gesamtansicht . . . . .	41
37	Achsendatensatz: Konturbaumausschnitt Achsenschnitt . . . . .	42
38	Achsendatensatz: Konturbaumausschnitt Kugel . . . . .	42

39	Achsendatensatz: Isoflächen . . . . .	43
40	Kopfdatensatz: Gesamtansicht . . . . .	43
41	Kopfdatensatz 1: Konturbaum . . . . .	44
42	Kopfdatensatz 1: Isofläche . . . . .	44
43	Kopfdatensatz 2: Konturbaum Punkte und Linien . . . . .	45
44	Kopfdatensatz 2: Konturbaum Volumenmodus . . . . .	46
45	Kopfdatensatz 2: Isofläche . . . . .	46
46	Klassisches Tree Layout . . . . .	48
47	Darstellungsmöglichkeiten für Graphen: oben: HTrees, Radial Trees, unten: Cone Trees, Tree-Maps . . . . .	48
48	Geänderte Datenaufbereitung . . . . .	49
49	Erweiterter Graph mit Quelle und Abfluss . . . . .	50
50	Baum mit Durchnummerierung der Knoten . . . . .	51
51	Kritische Punkte verbunden durch kritische Kurven . . . . .	52
52	Framework für Level Sets . . . . .	53
53	File Menü . . . . .	57
54	Tree and Surfaces Menü . . . . .	57
55	Mouse Mode Menü . . . . .	58
56	Tree Display Menü . . . . .	58
57	Beispiel für Visualisierung mit Knoten und Kanten . . . . .	58
58	Beispiel für Visualisierung mit Volumendaten . . . . .	59
59	Dialoge zur Selektion . . . . .	59
60	Konturbaum Dialog . . . . .	60
61	Fenster zur Isoflächendarstellung . . . . .	61

## Literatur

- [1] van Kreveld, M.; van Oostrum, R.; Bajaj, C.L.; Pascucci, V.; Schikore, D.R.: *Contour Trees and Small Seed Sets for Isosurface Traversal*. Proceedings of the 13th ACM Symposium on Computational Geometry, 212-220. ACM, 1997.
- [2] de Berg, M.; van Kreveld, M.: *Trekking in the Alps without freezing or getting tired*. Algorithmica, 18:306-323, 1997.
- [3] Carr, H.; Snoeyink, J.; Axen, U.: *Computing Contour Trees in all Dimensions*. Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, 918-926, 2000.
- [4] Tarasov, S.P.; Vyalys, M.N.: *Construction of Contour Trees in 3D in  $O(n \log n)$  steps*. Proceedings of the 14th ACM Symposium on Computational Geometry, 68-75, ACM, 1998.
- [5] Milnor J.W.: *Morse Theory*. Princeton University Press, Princeton, NJ, 1963.
- [6] Fujishiro, I.; Takeshima, Y.; Azuma, T.; Takahashi, S.: *Volume Data Mining Using 3D Field Topology Analysis*. IEEE Computer Graphics and Applications, September/Oktober 2000.
- [7] Kraus, M.; Ertl, T.: *Topology-Guided Downsampling*. Proceedings of International Workshop on Volume Graphics '01, 139-147, 2001.
- [8] Hirsch, M.W.: *Differential Topology*. Springer Verlag, Heidelberg, 1994.
- [9] Eppstein, D.: *Some Open Problems in Graph Theory and Computational Geometry*. ICS 200, University of California, 2001.
- [10] Cignoni, P.; Montani, C.; Scopigno, R.: *Tetrahedra Based Volume Visualization*. Workshop Visualization and Mathematics 1997, Berlin 1997.
- [11] Fuchs, H.; Kedem, Z.M.; Uselton, S.P.: *Optimal Surface Reconstruction from Planar Contours*. Communications of the ACM, vol. 20, 693-702, Oktober 1977.
- [12] Herman, G.T.; Chen, L.S.; Reynolds, R.A.; Udapa, J.K.: *Surface Shading in the Cuberille environment*. IEEE Computer Graphics and Applications, 5 (12), 33-43, 1985.
- [13] <http://www.siggraph.org/education/materials/HyperVis/vistech/volume/surface.htm>
- [14] Lorensen, W.E.; Cline, H.E.: *Marching Cubes: A high resolution 3D surface constructing algorithm*. Computer Graphics, 163-168, Juli 1987.
- [15] Nielson G.M.; Hamann, B.: *The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes*. IEEE Proceedings on Visualization 91, 83-91, IEEE, 1991.

- [16] Montani, C.; Scateni, R.; Scopigno R.: *Discretized marching cubes*. Proceedings of the Conference on Visualization, 281-287, Los Alamitos, CA, USA, Oktober 1994. IEEE Computer Society Press.
- [17] Shen, H.-W.; Johnson, C.R.: *Sweeping Simplicies: A fast iso-surface extraction algorithm for unstructured grids*. IEEE Visualization '95, 143-150, Oktober 1995.
- [18] Danskin, J.; Hanrahan, P.: *Fast Algorithms for Volume Ray Tracing*. Proceedings 1992 Workshop on Volume Rendering, 91-98, 1992.
- [19] Yagel, R.; Kaufman, A.: *Template-Based Volume Viewing*. Proceedings of Eurographics 92, 153-167, 1992.
- [20] Yagel, R.; Shi, Z.: *Accelerating Volume Animation by Space-Leaping*. Proceedings of Visualization, 1993.
- [21] Upson, C.; Keeler, M.: *V-Buffer: Visible Volume Rendering*. Computer Graphics, 59-64, Juli 1988. ACM Siggraph 88 Conference Proceedings.
- [22] Westover, L.: *Footprint Evaluation for Volume Rendering*. Computer Graphics, 24, August 1990, 367-376.
- [23] <http://www.cs.rug.nl/michel/waveletsplattting/wavelets.html>
- [24] Lacroute, P.; Levoy, M.: *Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation*. Proceedings of SIGGRAPH '94, 451-458, Juli 1994.
- [25] Cabral B.; Cam N.; Foran J.: *Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware*. Proceedings of 1994 Symposium on Volume Visualization, Oktober 1994, 91-98.
- [26] Whitaker, R.; Breen, D.; Museth, K.; Soni, N.: *A Framework for Level Set Segmentation of Volume Datasets*. Volume Graphics Workshop, Juni 2001.
- [27] Audette, M.A.; Peters, T.M.: *Level-set surface segmentation and registration for computing intrasurgical deformations*. Proceedings of SPIE, Nr. 3661, 1999.
- [28] Bajaj, C.; Pascucci, V.; Schikore, D.: *The contour spectrum*. Proceedings IEEE Visualization Conference, Oktober 1997.
- [29] Bajaj, C.; Pascucci, V.; Schikore, D.: *Visualization of Scalar Topology for Structural Enhancement*. Technical Report CSD-TR-96-006, Department of Computer Sciences, Purdue University, West Lafayette, IN, 1996.
- [30] Herman, I.; Melançon, G.; Scott M.: *Graph Visualization and Navigation in Information Visualization: a Survey*. IEEE Transactions on Visualization and Computer Graphics, Nr. 6, 2000.

- [31] Foley, J.D.; van Dam, A.; Feiner, S.K.; Hughes, J.F.: *Computer Graphics Principles and Practice*. 2nd Edition, Addison-Wesley: Reading, 1990.
- [32] Woo, M.; Neider, J.; Davis, T.; Shreiner, D.: *Open GL Programming Guide*. 3rd Edition, Addison-Wesley, 1999.
- [33] Dalheimer, M.: *Programming with Qt*. O'Reilly, 1999.



## **Erklärung**

Hiermit versichere ich, diese Arbeit  
selbständig verfasst und nur die  
angegebenen Quellen benutzt zu haben.

---

( Markus Merz)

