



UNIVERSITÄT  
KOBLENZ · LANDAU

Fachbereich 4: Informatik



# 3D-Kurven-Skelett Approximation aus Oberflächendaten von Objekten

Diplomarbeit  
zur Erlangung des Grades  
DIPLOM-INFORMATIKER  
im Studiengang Computervisualistik

vorgelegt von

Patric Lambrecht

**Betreuer:** Dipl.-Inform. J. Hedrich, Institut für Computervisualistik,  
Fachbereich Informatik, Universität Koblenz-Landau

**Erstgutachter:** Dipl.-Inform. J. Hedrich, Institut für Computervisualistik,  
Fachbereich Informatik, Universität Koblenz-Landau

**Zweitgutachter:** Prof. Dr.-Ing. Dietrich Paulus, Institut für  
Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau

Koblenz, im Dezember 2011



## Kurzfassung

3D-Kurven-Skelette werden oft verwendet, da sie im Vergleich zu der von Harry Blum 1967 vorgestellten Medialen Axen Transformation, eine Repräsentation der Objekt-Oberfläche darstellen, die in der Weiterverarbeitung weniger komplex und rechenintensiv ist. Diese Arbeit hat das Ziel, ein Approximation-Verfahren für 3D-Kurven-Skelette zu entwickeln, welches die oben genannten Vorteile erhält und sich auf verschiedene Szenarien von Objektoberflächen-Daten anwenden lässt. Hierzu wird zunächst die Skelett-Definition sowie deren Umsetzung in existierenden Verfahren vorgestellt. Um auf dieser Basis das Vorgehen zu erläutern, welches unter der Verwendung von Oberflächenpunkt-Normalen Skelettpunkte berechnet und daraus einer Skelettstruktur ermittelt. Die Umsetzung des Verfahrens als auch dessen Vor- und Nachteile werden mit bestehende Methodiken verglichen und bewertet. Abschließend werden verschiedene Aspekte dargelegt, welche in dieser Arbeit unberücksichtigt blieben, aber zu einer Erweiterung des Verfahren herangezogen werden können.

## Abstract

3D-Curve-Skeletons are often used, because the object surface representation is less complex and also needs less computing power in further processing, compared to the representation created by the Medial Axis Transformation introduced 1967 by Harry Blum. This theses aims at developing a 3D curve skeleton approximation algorithm that keeps these advantages and is also able to handle different scenarios of the object surface input data. Therefore the skeleton-defintion and its implementation in existing algorithms are explained. These will be used in the second part to explain the approach, which uses surface point normals to develop a skeleton structure. The approach itself and its advantages and disadvantages will be evaluated and compared to the existing methods. As around up, some aspects will be discussed that were not taken into consideration within the theses, but could be used to extend the algorithm.



## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Vereinbarung der Arbeitsgruppe für Studien- und Abschlussarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. ja  nein

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ja  nein

Koblenz, den 20. Dezember 2011



# Inhaltsverzeichnis

|          |                                                                                   |           |
|----------|-----------------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Einleitung</b>                                                                 | <b>13</b> |
| 1.1      | Fragestellung der Arbeit . . . . .                                                | 13        |
| 1.2      | Überblick . . . . .                                                               | 14        |
| <b>2</b> | <b>Theoretische Grundlagen</b>                                                    | <b>15</b> |
| 2.1      | Skelettdefintion . . . . .                                                        | 15        |
| 2.1.1    | Mediale Transformation im $\mathbb{R}^2$ . . . . .                                | 15        |
| 2.1.2    | MAT im $\mathbb{R}^3$ . . . . .                                                   | 17        |
| 2.2      | Invarianz gegenüber Translation, Rotation und Skalierung . . . . .                | 17        |
| 2.3      | Bijektivität der Abbildung . . . . .                                              | 18        |
| 2.4      | Skelett Geometrie . . . . .                                                       | 19        |
| 2.5      | Approximation Methoden . . . . .                                                  | 19        |
| 2.5.1    | Boundary Evolution . . . . .                                                      | 19        |
| 2.5.2    | Voronoi Skelett . . . . .                                                         | 22        |
| 2.6      | Kurvenskelette . . . . .                                                          | 24        |
| 2.6.1    | Geodätische Funktion $\theta$ . . . . .                                           | 25        |
| 2.6.2    | On-the-fly Curve-skeleton Computation(OTFCSC) . . . . .                           | 27        |
| 2.6.3    | Curve Skeleton Extraction From Incomplete Point Cloud . . . . .                   | 28        |
| <b>3</b> | <b>Eigener Ansatz</b>                                                             | <b>31</b> |
| 3.1      | Theoretische Annahme . . . . .                                                    | 31        |
| 3.2      | Normalisierung der Punktmenge . . . . .                                           | 32        |
| 3.3      | Normalen-Schätzung . . . . .                                                      | 32        |
| 3.3.1    | Bestimmung der Punktnormalen durch PCA . . . . .                                  | 32        |
| 3.3.2    | Ideale Nachbarschaftsgröße der Normalenschätzung $\rho_e$ . . . . .               | 33        |
| 3.3.3    | Reduktion der Punktdichte (Punktzahl) . . . . .                                   | 34        |
| 3.4      | Bestimmung der Normalenschnitte . . . . .                                         | 35        |
| 3.4.1    | Konkave/Konvexe Ausprägung lokaler Objekt Oberfläche . . . . .                    | 35        |
| 3.4.2    | Idealen Nachbarschaftsgröße $\rho_c$ für die Normalenschnitt-Berechnung . . . . . | 36        |
| 3.5      | Basis Parameter: $\gamma, \rho_e, \rho_c$ . . . . .                               | 38        |
| 3.6      | Erstellung des Skelettgraphen . . . . .                                           | 38        |

|          |                                                                                      |           |
|----------|--------------------------------------------------------------------------------------|-----------|
| 3.6.1    | Reduktion der potentiellen Skelettpunkte $S_{pot}$ . . . . .                         | 38        |
| 3.6.2    | Ermittlung des Growing-Initial Zustands . . . . .                                    | 39        |
| 3.6.3    | Iteratives Wachstum des Skeletgraphen . . . . .                                      | 39        |
| 3.7      | Vergleich mit existierenden Verfahren . . . . .                                      | 43        |
| 3.7.1    | On-the-fly Curve-skeleton Computation . . . . .                                      | 43        |
| 3.7.2    | Curve Skeleton Extraction From Incomplete Point Cloud . .                            | 45        |
| <b>4</b> | <b>Implementierung</b> . . . . .                                                     | <b>47</b> |
| 4.1      | Modularer Aufbau . . . . .                                                           | 47        |
| 4.1.1    | Daten einlesen . . . . .                                                             | 47        |
| 4.1.2    | Verarbeitung und Datenhaltung . . . . .                                              | 48        |
| 4.1.3    | Datenausgabe . . . . .                                                               | 48        |
| 4.2      | Parameter . . . . .                                                                  | 49        |
| 4.3      | Schrittweiser Programmablauf . . . . .                                               | 49        |
| <b>5</b> | <b>Test und Evaluierung</b> . . . . .                                                | <b>51</b> |
| 5.1      | Testdaten . . . . .                                                                  | 51        |
| 5.1.1    | Testdatensatz Name: Homer . . . . .                                                  | 53        |
| 5.1.2    | Testdatensatz Name: Modelier-Männchen . . . . .                                      | 54        |
| 5.2      | Ideale Werte für Parameter $\rho_c$ und $\rho_e$ . . . . .                           | 55        |
| 5.2.1    | Auswirkung von Werteänderungen für $\rho_e$ bei gleich bleibendem $\rho_c$ . . . . . | 55        |
| 5.2.2    | Auswirkung von Werteänderungen für $\rho_c$ bei $\rho_e = 9$ . . . . .               | 58        |
| 5.2.3    | Berechnungsdauer für unterschiedliche $\rho_e$ und $\rho_c$ . . . . .                | 59        |
| 5.3      | Evaluierung des Verfahrens für unvollständige Datensätze . . . . .                   | 61        |
| 5.4      | Komplexe Objekte . . . . .                                                           | 63        |
| <b>6</b> | <b>Zusammenfassung</b> . . . . .                                                     | <b>65</b> |
| 6.1      | Ausblick . . . . .                                                                   | 67        |



# Tabellenverzeichnis

|     |                                                                            |    |
|-----|----------------------------------------------------------------------------|----|
| 5.1 | Anzahl Normalenschnitte (gefiltert/ungefiltert) für wechselndes $\rho_c$ : |    |
|     | Homer . . . . .                                                            | 59 |
| 5.2 | Berechnungszeit der Normalenschätzung für wechselndes $\rho_e$ : Homer     | 59 |
| 5.3 | Berechnungszeit der Normalenschnitte für wechselndes $\rho_c$ : Homer .    | 60 |
| 5.4 | Ergebnis-Vergleich beider Testdatensätze . . . . .                         | 63 |



# Abbildungsverzeichnis

|      |                                                                         |    |
|------|-------------------------------------------------------------------------|----|
| 2.1  | Veranschaulichung der Grassfire-Analogie . . . . .                      | 16 |
| 2.2  | Beschreibung eines Objekts durch maximale Kreise . . . . .              | 16 |
| 2.3  | Skelettinvarianz gegenüber Translation . . . . .                        | 18 |
| 2.4  | Punktkategorien der Skelettstruktur . . . . .                           | 20 |
| 2.5  | Distanztransformation - Grauwert kodiert . . . . .                      | 20 |
| 2.6  | Beispiel einer Gradientenkarte . . . . .                                | 21 |
| 2.7  | Voronoi Skeletierung - Skelettkantensuche . . . . .                     | 23 |
| 2.8  | Voronoi Skeletierung; Binärbild, Voronoi-Diagramm, Skelett . . . . .    | 24 |
| 2.9  | 3D Voronoi Skelett . . . . .                                            | 25 |
| 2.10 | Anwendung der geödätschen Funktion . . . . .                            | 26 |
| 2.11 | On-the-fly Curve-skeleton Computation (schrittweise) . . . . .          | 27 |
| 2.12 | Curve Skeleton Extraction From Incomplete Point Cloud . . . . .         | 28 |
| 2.13 | ROSA Rotational Symmetry Axis für unvollständige Punktwoken . . . . .   | 29 |
|      |                                                                         |    |
| 3.1  | Abhängigkeit Normale zur Nachbarschaft . . . . .                        | 34 |
| 3.2  | Normalen - Schnittpunkte (konvex/konkav) . . . . .                      | 36 |
| 3.3  | Schnittpunkt Kompensation . . . . .                                     | 37 |
| 3.4  | Diskretisierung der Punktwolke mit Voxel Gitter . . . . .               | 39 |
| 3.5  | Bestimmung des initialen Growing Zustandes . . . . .                    | 40 |
| 3.6  | Diskretisierung der Punktwolke mittels Voxel Gitter . . . . .           | 41 |
| 3.7  | Suchbereichs für bester nächster Nachbar Suche . . . . .                | 42 |
| 3.8  | Tiefenscanner Daten . . . . .                                           | 44 |
|      |                                                                         |    |
| 5.1  | Syntetisch erzeugtes Objekt - Homer . . . . .                           | 53 |
| 5.2  | Tiefenscanner Daten . . . . .                                           | 54 |
| 5.3  | Normalenschnitte Homer für $\rho_e$ Werte 9 und 18 . . . . .            | 56 |
| 5.5  | Normalenschnitte Homer für $\rho_e$ Werte 144, 288, 512 . . . . .       | 56 |
| 5.4  | Normalenschnitte Homer für $\rho_e$ Werte 36 und 72 . . . . .           | 57 |
| 5.6  | Normalenschnitte im Verhältnis zu $\rho_e$ für Homer . . . . .          | 57 |
| 5.7  | Gefilterte Normalenschnitte für $\rho_c = 72$ und 144 . . . . .         | 58 |
| 5.8  | Ausführungszeit der Normalenschätzung für wachsendes $\rho_e$ . . . . . | 60 |

|      |                                                                                  |    |
|------|----------------------------------------------------------------------------------|----|
| 5.9  | Gefilterte Normalenschnitte für $\rho_c = 72$ und $144$ . . . . .                | 61 |
| 5.10 | M.figur Normalenschnitte ( $\gamma = 0,01, \rho_e = 18, \rho_c = 72$ ) . . . . . | 62 |
| 5.11 | M.figur Seitenansicht ( $\gamma = 0,01, \rho_e = 18, \rho_c = 72$ ) . . . . .    | 63 |
| 5.12 | Drachenmodell mit gefilterten Normalenschnitten . . . . .                        | 64 |

# Kapitel 1

## Einleitung

Die von Harry Blum im Jahr 1967 mathematisch beschriebene Mediale Achsen Transformation bildet die Grundlage für die Objekt-Repräsentation durch Skelette, welche in unterschiedlichen Bereichen der Informatik Anwendung finden. Hierzu zählen unter anderem die Medizinische Bildverarbeitung und die Computergrafik. Kurven-Skelette sind eindimensionale Objekte, welche die Oberfläche eines Objektes in vereinfachter Form repräsentieren. Diese vereinfachte Repräsentation wird verwendet, da die Verarbeitung der aus der Medialen Achsen Transformation resultierenden Skelette im  $\mathbb{R}^3$  durch die zweidimensionalen Flächen innerhalb des Skelettes komplex und rechenintensiv ist.

### 1.1 Fragestellung der Arbeit

Daher stellt sich die Frage nach einem geeigneten Kurven-Skelett-Approximations-Verfahren das auf Basis der Oberflächen Daten eines Objektes arbeitet, eine moderate Komplexität und damit auch einen geringen Rechenaufwand verursacht. Anhand weniger Parameter soll der Grad der Approximationsgenauigkeit und der daraus resultierende Berechnungsaufwand steuerbar sein.

In diesem Zusammenhang wird analysiert werden, welche Bedingungen die als Eingabe erforderlichen Oberflächen-Daten erfüllen müssen, und wie sich deren Unvollständigkeit auf das in dieser Arbeit vorgestellte Verfahren auswirken. Im Folgenden bedarf es der Beobachtung von zusätzlichen Eingabe-Parameter und deren Einfluss auf das Ergebnis für diese unterschiedlichen Szenarien in Relation auf die zu verarbeitenden Daten. Ebenso wird ein Vergleich mit bereits existierenden Verfahren angestellt werden, der zeigt ob diese Vor- oder Nachteile gegenüber dem hier entwickelten Verfahren haben.

## 1.2 Überblick

Als Diskussions Grundlage werden in Kapitel zwei theoretischen Grundlagen der Skelett-Approximation nach Blum erklärt, indem verschiedene Ansätze wie Boundary-Evolution, Thinning und die Skelett-Approximation auf Basis von Voronoi-Diagrammen eingeführt werden. Zum Abschluss diese Kapitels werden exemplarisch zwei existierende Verfahren der Kurven-Skelett-Approximation vorgestellt, deren Vor- und Nachteile im Vergleich zum hier vorgestellten Verfahren in Kapitel drei aufgezeigt werden.

In Kapitel drei erläutert zunächst die diesem Verfahren zu Grunde liegende These:

„Skelettpunkte resultieren aus den Schnittpunkten der  
Oberflächen-Punkt-Normalen“

Die These wird unter Verwendung der theoretischen Grundlagen aus Kapitel zwei erläutert und die Berechnung dieser Schnittpunkte vorgestellt. Im Folgenden wird ein Growing-Algorithmus entwickelt, der es ermöglicht, diese Schnittpunkte in einen Skelett-Graph zu überführen. Desweiteren werden Rahmenbedingung sowie Vor- und Nachteile dieses Ansatzes aufgezeigt und in Relation zu den beiden bereits eingeführten Verfahren betrachtet.

Weiterhin wird erläutert wie das Verfahren implementiert ist und welche externen Bibliotheken verwendet werden. Hierzu werden die einzelnen Arbeitsschritte die im Programmcode modular abgebildet sind erklärt und deren Bezug zur theoretischen Beschreibung des Verfahrens hergestellt.

Da Kurven-Skelette in der Literatur nur unscharf definiert sind, erfolgen die in Kapitel fünf durchgeführten Evaluierung hauptsächlich auf visuell. Die Veränderung der für den Algorithmus notwendigen Eingabeparameter sowie deren Auswirkung auf das Ergebnis werden erklärt und die idealen Eingabewerte im Verhältnis zur Objekt-Oberflächen-Punktewolke bestimmt. Die Evaluierung bezieht hierbei sowohl auf vollständige als auch die als Ziel der Arbeit angestrebten Repräsentation unvollständiger Objekt-Oberflächen-Punktewolke ein.

Zum Abschluss der Arbeit wird dargelegt, in welchen Bereichen der Algorithmus noch Schwächen aufweist und wo Potenzial zur Verbesserung bzw. Erweiterung besteht. Es werden Verwendungsszenarien aufgezeigt, durch die die Anwendbarkeit auf unvollständige Objekt-Oberflächen-Punktewolken und den geringen Rechenaufwand möglich werden.

# Kapitel 2

## Theoretische Grundlagen

Zum besseren Verständnis der Thematik Skelett-Approximation werden hier zunächst die Skelett-Definition nach Harry Blum, Skelett-Eigenschaften als auch Algorithmen vorgestellt, die eine Skelett-Approximation gemäß der blumschen Definition durchführen. Im Anschluss wird ein weiterer Skeletttypus, die Kurven-Skelette eingeführt, welche zwar auf dieser Skelett-Definition aufbauen, diese aber nicht in ihrem vollen Umfang berücksichtigen und somit getrennt zu behandeln sind. Hierzu werden zwei Algorithmen zur Erstellung von Kurven-Skelette vorgestellt.

Im Anschluss an dieses Kapitel werden im Zusammenhang der Darstellung des eigenen Ansatzes alle vorgestellten Verfahren bezüglich ihrer Vor- und Nachteile verglichen und bewertet.

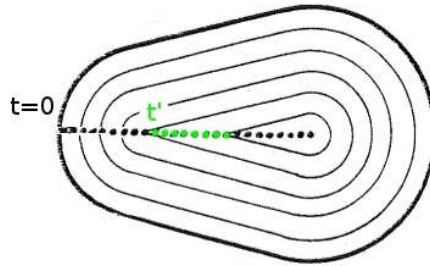
### 2.1 Skelettdefinition

Ein Skelett ist eine vom Objekt abstrahierte Repräsentationsform, und basiert auf der von Harry Blum entwickelten „Grassfire“-Analogie [Blu67].

Sei das Objekt eine Grassfläche  $G$ , an der zum Zeit  $t = 0$  gleichzeitig die Menge aller Punkten der Umrandung  $U$  entzündet werden (Abb. 2.1). Die Feuerfront verbrennt mit voranschreitender Zeit  $t_i$  immer mehr Fläche, bis sie zu einem Zeitpunkt  $t'$  mit sich selber kollidiert. An dieser Stelle befinden sich die Medialen Punkte (Medial Locus), die durch Verknüpfung die Medialen Achsen (Medial Axis) ergeben.

#### 2.1.1 Mediale Transformation im $\mathbb{R}^2$

Diese Transformation von Objektumrandung/-oberfläche ist die Mediale Achsen Transformation (MAT). Damit die Bijektivität der Medial Achsen Transformation (MAT) gewährleistet ist muss in jedem Medialen Punkte der Abstand  $r$  zur



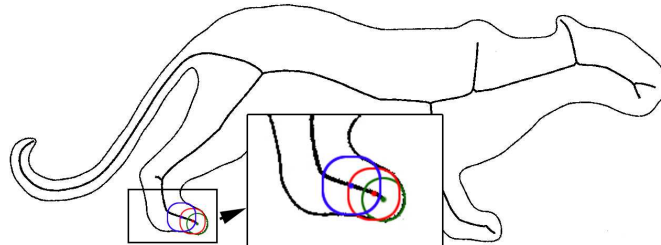
**Abbildung 2.1:** Veranschaulichung der Grassfire-Analogie. Die Objektgrenzen (stark gezeichnete Linie) zum Zeitpunkt  $t = 0$ . Die konzentrischen Linien zeigt die Feuerfront zu verschiedene Zeitpunkten  $t_i$  zwischen  $t = 0$  und  $t'$ . Hier grün markiert ist der Zeitpunkt  $t'$  an dem die Feuerfront kollabieren [Blu67]

Umrandung des Objektes bekannt sein. Dieser Abstand  $r$  ist der Radius des maximalen Kreises  $rD$  mit  $s$  als Mittelpunkt, welcher auf der Medialen Achse liegt und daher ein Skelettpunkt ist. Der maximale Kreis  $rD$  liegt komplett innerhalb des Objekt ( $O$ ) und es gibt keinen anderen Kreis  $rD'$  der ihn umschließt und ebenfalls komplett innerhalb von  $O$  liegt.

**Definition:** Maximaler Kreis ( $rD$ ) in Objekt ( $X$ )[SP08]

$$s \in \mathbb{R}^2 \text{ und } rD_s \subset \mathbb{R}^2$$

$$rD_s \text{ ist maximaler Kreis} \Leftrightarrow rD_s \subset O \bigwedge \nexists rD'_s : rD_s \in rD'_s \in O \quad (2.1)$$



**Abbildung 2.2:** Objekt mit seinen Skelettlinien. Der rote, blaue und grüne Kreis zeigt für jeweils einen Punkt auf der Skelettlinie den maximalen Kreis, der lokal die Objektoberfläche beschreibt [SP08]

**Definition:** Skelettpunkte-Menge ( $S_O$ ) für  $O$  anhand von Zentren max. Kreise ( $rD_s$ ) [Jai89]

$$S(O) = \bigcup_{r>0} s : \text{ist Mittelpunkt von } rD_s \quad (2.2)$$

Für eine zu Grunde gelegten diskretisierten Objektraums ergeben sich für jeden maximalen Kreis  $rD$  mindestens zwei Berührungspunkte mit der Umrandung des



## 2.2. INVARIANZ GEGENÜBER TRANSLATION, ROTATION UND SKALIERUNG 17

Objektes  $U$ . Daraus folgt, dass es von jedem Punkt  $s$  der Medialen Achse  $MA_O$  mehr als einen Punkt  $p$  mit dem kürzesten Abstand  $\theta$  zur Umrandung gibt. Der Abstand  $\theta$  ist der euklidische Abstand  $d(s, p_i)$  zwischen Skelettpunkt  $s$  und seine jeweils korrespondierenden Umrandungspunkt  $p_i$

**Definition:** Alternative Definition über den kürzesten Abstand [WIKI]

$$MA_O = \{s \in \mathbb{R}^2 : \exists p_1 \neq p_2 \in \mathbb{R}^2 \cup \theta = d(s, p_1) = d(s, p_2)\} \quad (2.3)$$

### 2.1.2 MAT im $\mathbb{R}^3$

Alle Skelettpunkte  $s$  sind im  $\mathbb{R}^3$  analog zur  $\mathbb{R}^2$  Definition Mittelpunkte maximaler Kugel  $rK$  mit einem Radius  $r$ . Ebenso gilt für jede maximale Kugel  $rK$  das es keine Kugel  $rK'$  gibt, die komplett innerhalb des Objektes  $O$  und  $rK$  beinhaltet. Die Oberflächenpunkte  $P$  und Skelettpunkte  $S$  sind im  $\mathbb{R}^3$  definiert. Der Unterschied im  $\mathbb{R}^3$  besteht darin das die maximale Kugel die Objektoberfläche in mindestens drei Punkten berührt.

## 2.2 Invarianz gegenüber Translation, Rotation und Skalierung

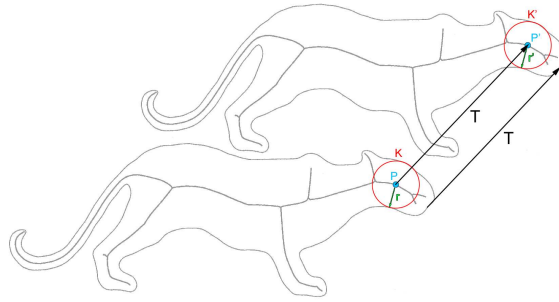
Die Mediale Achsen Transformation ist eine Abbildung, die invariant gegenüber Translation, Rotation und Skalierung ist [Ogn92].

Jeder Skelettpunkt  $s$  besteht aus dem Tupel  $(pos, r)_s$ , wobei  $pos$  die Position innerhalb von  $\mathbb{R}^2$  bzw.  $\mathbb{R}^3$  und  $r$  den Radius (Abstand zu Objektoberfläche) darstellt. Bei einer Translation wird auf jeden Skelettpunkt  $s$  die Translationsmatrix  $T$  angewendet, mit der auch die Oberflächenpunkte  $P$  des Objekt transliert wurden. Bei dieser Operation bleibt der jedem Skelettpunkt zugeordnete Radius  $r$  erhalten, während sich durch die Translation für die Position  $pos$  eine neue Position  $pos'$  ergibt.

$$(pos', r)_s = (pos, r)_s * T \quad (2.4)$$

Die Rotation wird durch die Anwendung der Rotationsmatrix  $R$  auf Objektoberflächenpunkte  $P$  und die Skelettpunktupel  $(pos, r)_s$  durchgeführt. Der Radius  $r$  verändert sich durch die Rotation nicht.

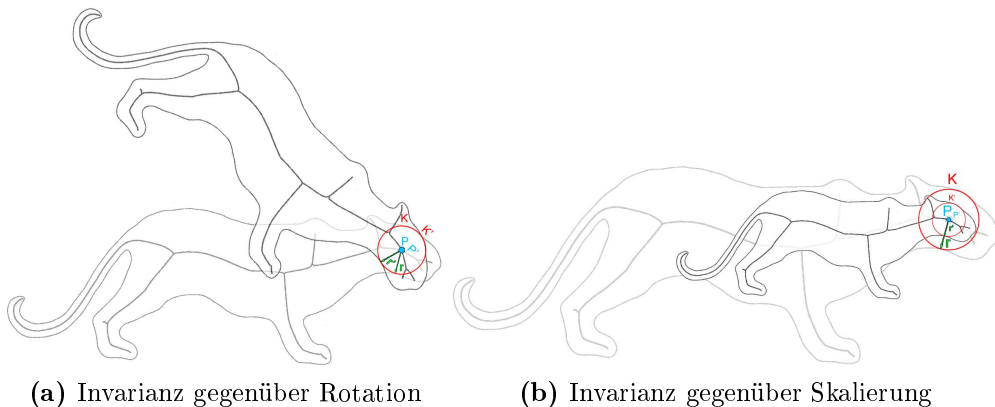
$$(pos', r)_s = (pos, r)_s * R \quad (2.5)$$



**Abbildung 2.3:** Die Translation  $T$  wirkt sich auf Objekt und Skelett gleich aus, indem die Position verändert wird. Alle Radien für die Skelettpunkte bleiben erhalten (hier  $r = r'$ ).

Durch eine Skalierung wird der Radius  $r$  abhängig vom Skalierungsfaktor  $\nu$  verändert. Die Position der Skelettpunkte  $s$  bleibt unverändert.

$$(pos, r')_s = (pos, r * \nu)_s \quad (2.6)$$



(a) Invarianz gegenüber Rotation

(b) Invarianz gegenüber Skalierung

## 2.3 Bijektivität der Abbildung

Die Mediale Achsen Transformation ist eine bijektive Abbildung. Daher gilt das sich aus den Skelettpunkten  $s$  unter Verwendung der Radien  $r$  und der dadurch definierten maximalen Kreise  $rD$  durch Anwendung der Umkehrfunktion der Abbildung  $\odot$  alle Punkte der Objektoberfläche  $P$  wiederherstellen lassen.

$$P = \bigcup_{r>0} (s \odot rD) \quad (2.7)$$

## 2.4 Skelett Geometrie

Die Menge der Skelettpunkte  $S$  besitzt drei Teilmengen  $A_1$ ,  $A_2$  und  $A_3$  [SP08]. Diese Aufteilung ergibt sich aus der Anzahl der Punkte aus  $S$ , die in der Nachbarschaft  $N(s)$  eines Skelettpunktes  $s$  liegen. Ein Punkt liegt in der Nachbarschaft  $N(s)$ , wenn er direkt über die Mediale Achse mit dem Punkt  $s$  verbunden ist (Abb. 2.4).

### Endpunkt $A_1$ (Leaf-Point)

Jeder Punkt in  $A_1$  besitzt genau einen benachbarten Punkt, ist aber über die Mediale Achse mit allen anderen Punkten aus  $A_1$  verbunden. Jedes Skelett, ausgenommen das eines Kreises bzw. einer Kugel, besitzt mindestens zwei Leaf-Points, in denen die Mediale Achse beginnt bzw. endet.

### Mittelpunkt $A_2$ (Branch-Point)

Jeder Punkt in  $A_2$  besitzt genau zwei Nachbarn, wodurch die Verbindungen zwischen Leaf- und Junction Points beschrieben werden.

### Verbindungspunkt $A_3$ (Junction-Point)

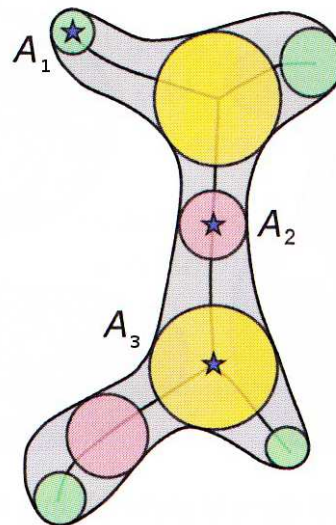
Jeder Punkt in  $A_3$  besitzt mehr als zwei benachbarte Punkte. Daraus folgt das ein Junction-Point Unterstrukturen des Skelettes  $S_{sec}$  verbindet. Im Umkehrschluss lässt sich das Skelett in diesen Punkten in seine Unterstrukturen unterteilen, hierbei wird der Junction-Point  $s_{join}$  für jede Unterstruktur  $S_{sec}$  zu einem Endpunkt.

## 2.5 Approximation Methoden

In diesem Abschnitt werden zwei Skelett-Aproximations-Verfahren (Boundary Evolution, Voronoi-Skeletierung) vorgestellt. Beide Verfahren lassen sich auf Daten im  $\mathbb{R}^2$  und  $\mathbb{R}^3$  anwenden, da die darin verwendeten Funktionen in beiden Räumen definiert sind.

### 2.5.1 Boundary Evolution

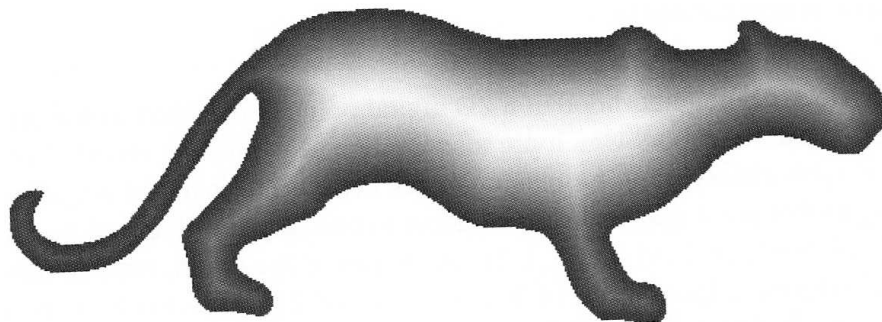
Grundlage für dieses Verfahren bildet eine diskretisierte Objekt-Präsentation (z.B. Pixel im  $\mathbb{R}^2$ ). Die Menge aller Pixel besteht in dieser Darstellung aus zwei Teilmengen  $PP_{in}$  und  $PP_{out}$ . Die Menge  $PP_{in}$  beschreibt hierbei alle Pixel die innerhalb der Objekt-Begrenzung (Boundary  $U$  - Umrandung im  $\mathbb{R}^2$ , Oberfläche im  $\mathbb{R}^3$ ) liegen und  $P_{out}$  alle Pixel die sich außerhalb dieser Grenzen befinden.



**Abbildung 2.4:** Diese Illustration zeigt die verschiedenen Skelettpunkt Kategorien jeweils mit ihren maximalen Kreisen entsprechend der Kategorie eingefärbt.  $A_1$  Endpunkte (Leaf-Point);  $A_2$  Mittelpunkte (Branch-Point);  $A_3$  Verbindungspunkte (Junction-Point) [SP08]

### Distance Transform

Für Pixel  $PP \in PP_{in}$  wird mit der kleinste euklidischen Abstand  $r$  Objekt-Begrenzung  $U$  ermittelt. In Abbildung 2.5 ist dieser Abstand als Grauwert kodiert als Wert für den Pixels eingetragen.



**Abbildung 2.5:** Distanztransformation; Die Distanzen zum jeweils nächsten Punkt der Objektumrandung ist hier als Grauwerten kodiert [SP08]

Die Gradientenkarte (Abb. 2.6) ist eine Erweiterung der hier angewandte Distanz Transformation. Jeder Gradient ist durch eine Vektor abgebildet, der in einer Pixelnachbarschaft der Pixelmenge  $PP_{in}$  den höchsten Anstieg des Abstandswertes

$r$  anzeigt. Für alle Skelettpunkte aus  $PP_{in}$  gilt das es keinen benachbarten Punkt  $PP'_{in}$  gibt, der kein Skelettpunkt  $S$  ist und einen größeren Abstand  $r$  besitzt.

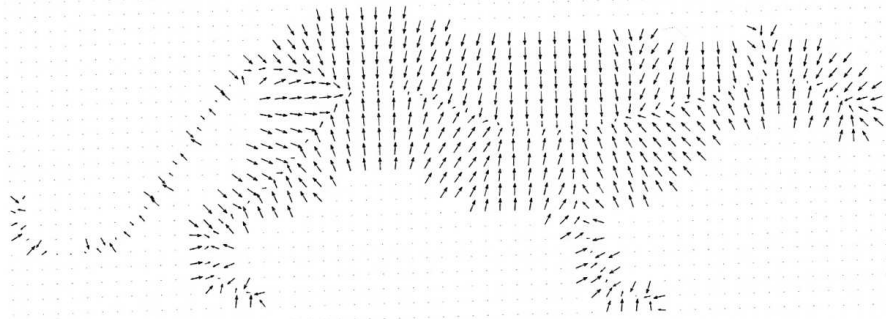


Abbildung 2.6: Beispiel einer Gradientenkarte [SP08]

### Thinning

Das Thinning ist ebenfalls ein Boundary Evolution Verfahren, welches analog zur Grassfire-Analogie von Blum iterativ die Objektbegrenzung  $U$  zur Medialen Achse hin schrumpft. Zu Beginn des ersten Iterationsschritts  $k = 0$  liegen die Daten, wie im vorangehenden Abschnitt erklärt, in den Pixel-/Voxelmenge  $PP$ , die sich  $PP_{in}$  und  $PP_{out}$  aufteilt. Diese Binärrepräsentation ist Voraussetzung für die Anwendung von Morphologischen Operationen. Es werden acht Strukturmasken  $M$  verwendet, welche sich aus der Rotation zweier Initialmasken ergeben wie sie in der folgenden Aufstellung gezeigt werden.

$$\begin{array}{rcl}
 M_0 = \begin{array}{ccc} 0 & 0 & 0 \\ x & 1 & x \\ 1 & 1 & 1 \end{array} & M_1 = \begin{array}{ccc} 0 & x & 1 \\ 0 & 1 & 1 \\ 0 & x & 1 \end{array} & \\
 M_2 = \begin{array}{ccc} 1 & 1 & 1 \\ x & 1 & x \\ 0 & 0 & 0 \end{array} & M_3 = \begin{array}{ccc} 1 & x & 0 \\ 1 & 1 & 0 \\ 1 & x & 0 \end{array} & \\
 M_4 = \begin{array}{ccc} 0 & 0 & x \\ 0 & 1 & 1 \\ x & 1 & 1 \end{array} & M_5 = \begin{array}{ccc} x & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & x \end{array} & (2.8) \\
 M_6 = \begin{array}{ccc} 1 & 1 & x \\ 1 & 1 & 0 \\ x & 0 & 0 \end{array} & M_7 = \begin{array}{ccc} x & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & x \end{array} & 
 \end{array}$$

Die Masken  $M_1 - M_8$  sind eine Erweiterung der Erosions- und Dilationsmasken, welche nur einen binären Wert für ein Pixel zulassen. Der Wert  $x$  in einer Maske steht für einen variablen Wert ( $0 \in PP_{out}$  oder  $1 \in PP_{in}$ ). Die Vereinigung der Ergebnisse aus dem Template-Matching mit allen Masken für  $PP$  ergeben das Resultat  $PP^k$  eines Iterationsschritts.  $PP^k$  wird im nächsten Iterationsschritt als Pixel-/Voxelmenge verwendet. Ist  $PP^{n-1} = PP^n$  endet die Iteration.

$$\begin{aligned}
 k = 0, PP^{k=1} &= \bigcup_{i=0}^7 PP \otimes M_i \\
 k = 1, PP^{k=2} &= \bigcup_{i=0}^7 PP^{k=1} \otimes M_i \\
 &\dots \\
 k = n, PP^{k=n} &= \bigcup_{i=0}^7 PP^{k=n-1} \otimes M_i, \quad PP^{n-1} = PP^n
 \end{aligned} \tag{2.9}$$

Ist das durch die Maske  $M_i$  definierte Template in Punkt  $PP_i$  deckungsgleich mit den Werten der Nachbar Punkten, wird der Wert von  $PP_i \in PP^{k+1}$  auf eins gesetzt.

Wie schon beschrieben wird der Objektraum im  $\mathbb{R}^3$  in Voxel aufgeteilt. Die Anzahl der Voxel wird durch die Auflösung bestimmt. Da die Rotation sowie Translation eines Objektes innerhalb des  $\mathbb{R}^3$  variabel ist, sollte die Auflösung in Dimensionen gleich sein, um die Unterschiede bei der Diskretisierung verschiedener Objekte so gering zu halten.

Die Distanz-Transformation basiert auf dem euklidischen Abstand, welcher für  $\mathbb{R}^2$  und  $\mathbb{R}^3$  definiert ist. Für das  $\mathbb{R}^3$  Thinning Verfahren müssen die Masken entsprechend dem  $\mathbb{R}^3$  erweitert werden.

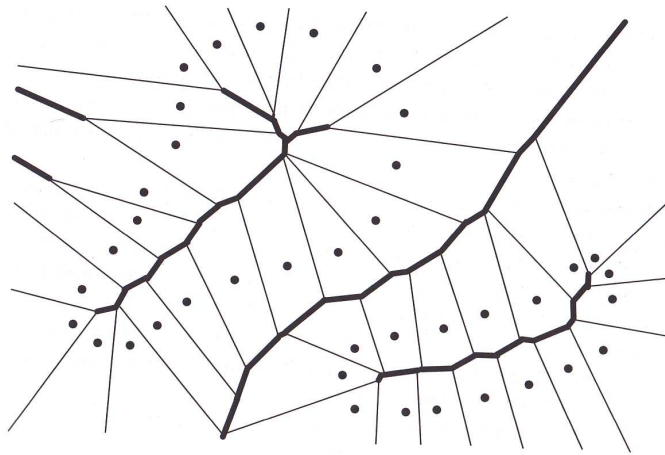
### 2.5.2 Voronoi Skelett

Das Voronoi Skelett-Approximation-Verfahren basiert auf der Erstellung eines Voronoi-Diagramms für die Oberflächenpunkte des Objektes. Das Voronoi-Diagramm  $V$  einer Punktmenge  $P$  ist eine Unterteilung die Ebene im  $\mathbb{R}^2$  in Voronoi-Zellen  $VZ$  die durch Voronoi-Kanten  $VE$  begrenzt werden.  $VP$  Voronoi-Punkte sind hierbei die Verbindungspunkte der Voronoi-Kanten. Für einen Punkt  $p_i \in P$  gibt es eine

zugeordnete Voronoi-Zelle  $VZ$ , für die gilt das für jeden Punkt aus  $q \in \mathbb{R}^2 | q \in VZ$  der euklidische Abstand  $dist(q, p_i) < dist(q, p_j)$  für alle  $p_j \in P | j \neq i$  [Fis04].

Bei der Skelett-Approximation ist die Punktmenge  $P$  durch die Umrandungspunkte des Objektes geben. Im ersten Schritt dieses Verfahren wird eine Delaunay-Triangulierung der Punktmenge  $P$  durchgeführt, welche ein Dreiecksnetz aufspannt. Eckpunkte für die Dreiecke bilden die Punkte der Punktmenge  $P$ . Für jedes Dreieck  $K$  gilt, dass sich innerhalb des Dreiecks-Umkreises kein Punkt  $q \in P$  gibt, der kein Eckpunkt des Dreiecks  $K$  ist.

Die Mittelsenkrechte auf den Kanten der Delaunay-Dreiecke bilden die Menge der Voronoi-Kanten  $VE$ . Die Voronoi-Punkte  $VP$  ergeben sich aus den Schnittpunkten dieser Voronoi-Kanten.



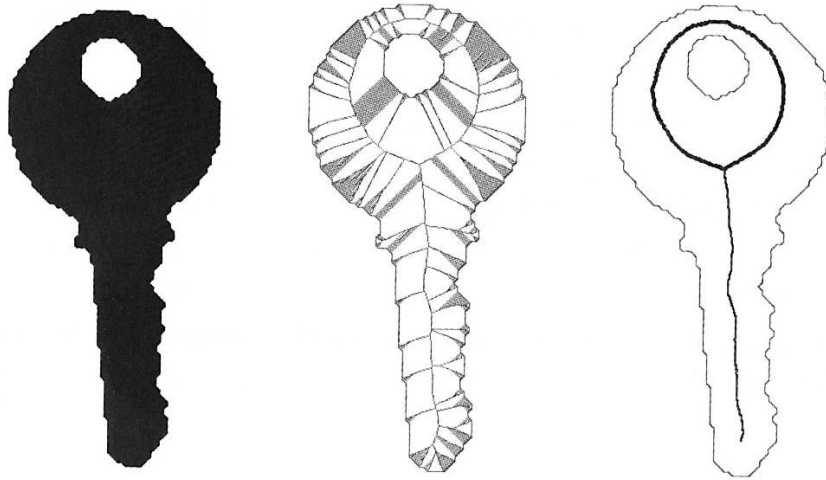
**Abbildung 2.7:** Voronoi Skelett; Die markierte Voronoi-Kanten sind Kanten, bei denen es sich potentiell um Skelettkanten handelt. In dem hiergezeigten Beispiel wurde noch keine Aufteilung in die beiden Mengen  $VP_{in}$  und  $VP_{out}$  durchgeführt. Die Punkte sind die Objektoberflächenpunkte  $P$ .

Im nächsten Schritt wird das Voronoi-Diagramm an der Umrandung des Objektes geschnitten. Dadurch ergeben sich für die Voronoi-Kanten zwei Untermengen pro Elemente des Voronoi-Diagramms (Abb. 2.7), Voronoi-Kanten die innerhalb des Objektes ( $VE_{in}$ ) und die außerhalb des Objektes liegen ( $VE_{out}$ ). Kanten, welche durch den Schnitt an der Objektumrandung geteilt werden, sind keiner der Mengen zugeordnet. Voronoi-Punkte sind  $VP_{in}$  zugeordnet, wenn sie innerhalb der Objektgrenzen liegen und  $VP_{out}$  wenn nicht.

Jeder Punkt der Menge  $VP_{in}$  erfüllt die blumsche Bedingung für Punkte einer Medialen Achse, insofern dass er zu mindestens zwei bzw. drei (in  $\mathbb{R}^3$ ) Umran-

zungspunkten äquidistant ist. Diese Bedingung gilt für alle Punkte  $s$  die auf den Voronoi-Kanten in  $VE_{in}$  liegen.

Da jede  $VE$  im  $\mathbb{R}^2$  über die zugehörigen Voronoi-Regionen  $VR$  mit zwei Umrandungspunkten  $p, q \in P$  verknüpft ist und es eine Nachbarschaftsbeziehung  $N_2(p_i, p_{i+1})$  gibt, so gilt diese auch für korrespondierenden Voronoi-Kanten  $N_2(VE_i, VE_{i+1})$ . Die Verbindung der Voronoi-Kanten zum Gesamtskelett erfolgt unter Berücksichtigung dieser Verknüpfung [SP08].



**Abbildung 2.8:** Diese Grafik zeigt die drei Schritte vom Binärbild über das an der Objekumrandung geklippede Voronoi-Diagramm zum Skelett [SP08]

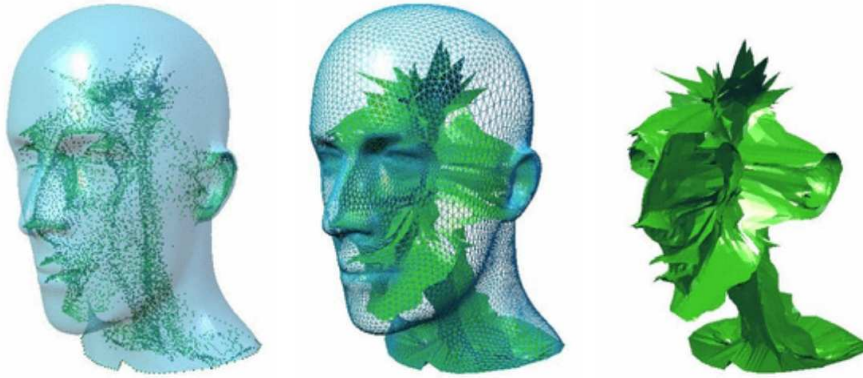
Voronoi-Diagramme sind für  $\mathbb{R}^3$  definiert [BK06], und ermöglichen daher die Anwendung des Verfahrens auf eine Objektoberflächen-Punktemenge  $P$  aus  $\mathbb{R}^3$ . Anhand der Punkte  $P$  wird der Raum in konvexe Polyeder aufgeteilt. Diese Polyeder verhalten sich analog zu den Voronoi-Regionen  $VR$  im  $\mathbb{R}^2$  sind durch Polygone begrenzt die den Voronoi-Kanten  $VE$  im  $\mathbb{R}^2$  entsprechen.

Das Ergebnis-Skelett des Verfahrens besteht im  $\mathbb{R}^3$  aus Polygone welche miteinander verknüpft sind (Abb. 2.9).

## 2.6 Kurvenskelette

Kurvenskelette sind die Reduktion einer Medialen Achsen Repräsentation für  $\mathbb{R}^3$  zu einer eindimensionalen Kurvenrepräsentation des Objektes. Das Ergebnis einer





**Abbildung 2.9:** Vollständiges Modell eines Kopfes mit den Voronoi-Punkten innerhalb des Objektes  $VP_{in}$ ; Kopfmodell mit den zugehörigen innerhalb liegenden Voronoi-Kanten  $VE_{in}$ ; Abbildung von  $VE_{in}$  ohne das Modell [YBS03]

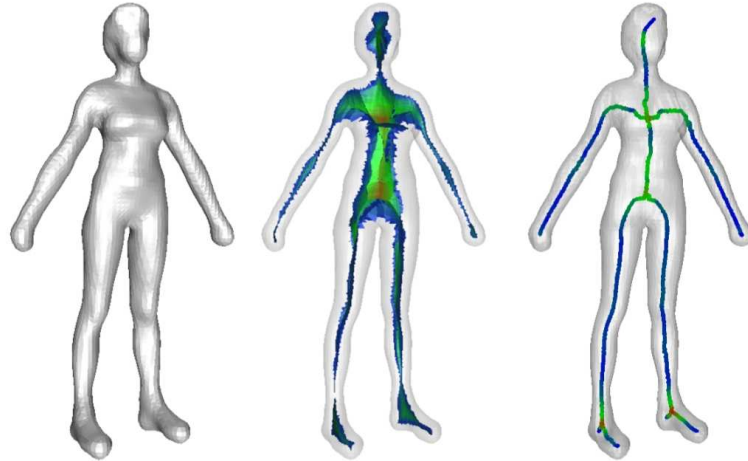
Medialen Achsen Transformation in  $\mathbb{R}^3$  ist ein dreidimensionales Skelett  $S_{MAT}$ , das Flächen beinhaltet (Abb 2.9).

Um ein Kurvenskelett  $S_{curve}$  zu erzeugen, werden im diskretisierten Raum alle Punkte  $s_F$  dieser Flächen  $F$  mit einer Medialen Geodätschen Funktion  $\theta$  bewertet. Anhand der bewerteten Punkte  $s_F$  wird wie in Abbildung mit Hilfe einer Funktion  $\kappa$ , auf die im folgenden Kapitel noch näher eingegangen wird, 2.10 eine eindimensionale Kurve  $C$  erzeugt. Das Kurvenskelett  $S_{curve}$  ist durch diese Reduktion keine bijektive Abbildung des Objektes  $O$ . Durch das Entfernen von Punkte  $s_F$ , welche durch die geodätschen Funktion einen niedrigeren Wert erhalten haben, wird auch die an diesen Punkten gespeicherte Information über die Objektoberfläche verworfen. Somit fehlen diese Information bei Anwendung der Umkehrfunktion  $\oslash$  zur Rekonstruktion des Objekts  $O$ . Es wird dadurch nicht in vollem Detailumfang aus dem Kurvenskelett wiederhergestellt.

$$\begin{aligned} rated S_F &= \theta(s_F, f) | \forall F \in S_{MAT} \\ S_{curve} &= \kappa(S_F) \end{aligned}$$

### 2.6.1 Geodätsche Funktion $\theta$

Die geodätsche Funktion  $\theta$  findet den kürzesten Weg zwischen zwei Punkten einer Fläche  $s_{1F}$  und  $s_{2F}$ . Dabei berücksichtigt die Funktion Gewichtung  $W(s_F)$  die



**Abbildung 2.10:** Vollständiges 3D Modell einer Frau; Modell mit Flächenskelett, welches farblich kodiert das Ergebnis der geodätischen Funktion zeigt. Approximiertes Kurvenskelett auf Basis der Information aus der zweiten hier gezeigten Illustration [KSJ06]

Anhand von verschiedenen Attributen der Punkte innerhalb dieser Fläche [KSJ06] getroffen wird.

Das für die Extraktion des Kurven-Skeletts am stärksten gewichtete Attribut  $A_{i,s_F}$  eines Punkte  $s_F$  ist sein Abstand zum Rand der Fläche, der er zugeordnet ist. Hiermit wird eine gemittelte Gerade durch die Fläche  $F$  ermittelt. Weitere Punktattribute, die im Zusammenhang mit der Skelett-Approximation verwendet werden, sind der Abstand zum Schwerpunkt der Skelettpunktemenge  $S$  in der der Punkt liegt. Desweiteren gibt es neben diesen statischen auch dynamische Attribute, welche sich erst zur Laufzeit eine Skelett-Approximations-Verfahrens ergeben. Hier ist zum Beispiel die Wertigkeit eines Punktes im Bezug auf die Flussrichtung eines Skelettgraphen zum Zeitpunkt  $t_i$  bei dessen Erzeugen zu nennen.

Die Abbildung 2.10 zeigt eine solche Gewichtung farblich kodiert. Der Gewichtung  $W(s_f)$  des Punktes  $s_f$  errechnet sich aus verschiedenen Attributen  $A_{i,s_F}$ , welche dem Punkte zugeordnet sind. Die Attribute werden entsprechend ihrer Wertigkeit im Bezug auf das Verfahren mit einem Multiplaktor  $m$  gewichtet, aufsummiert und zur Normalisierung durch die Attributanzahl  $\#_A$  geteilt.

$$W(s_f) = \frac{1}{\#_A} \sum_{i=0}^{\#_A} m * A_{i,s_F} \quad (2.10)$$

### 2.6.2 On-the-fly Curve-skeleton Computation(OTFCSC)

Bei diesem Verfahren wird das Objekt durch eine aus seinem Zentrum wachsenden Punktwolke (Deformable Modell)  $DM$  approximiert [SLSK07].

Im ersten Iterationschritt  $i = 0$  wird diese Punktwolke als Sphäre in das Zentrum des Objektes gelegt. Zu jedem Schritt  $i$  gibt es für diese Punktwolke  $DM$  Wachstumsfronten  $GF$ . Im Schritt  $i = 0$  gibt es eine Wachstumsfront  $GF_0$ , die aus allen Oberflächepunkten der Punktwolke  $DM$  besteht, für die gilt das keine Punkt  $\in DM$  außerhalb der Objektgrenzen liegt. Mit jedem Iterationsschritt wird die Punktwolke  $DM$  expandiert bis diese im Schritt  $i'$  die Objekt-Oberfläche  $U$  in mindestens drei Punkten berührt. Ist diese Bedingung erfüllt, wird für die aktive Wachstumsfront  $GF_i$  der Mittelpunkt  $gf_{mean}$  berechnet. Dieser Punkt  $gf_{mean}$  wird der Menge der potentiellen Skelettpunkte  $S_{pot}$  zugeordnet. An Verzweigungen des Objektes teilt sich die Front  $GF_i$  in  $k$  Teilfronten  $GF_{i,k}$  auf (Abb. 2.11 - Aufteilung in die einzelnen Finger), die sich wenn es durch die Objektstruktur bedingt ist auch wieder zu einer Front verbinden können. Teilen oder verbinden sich Fronten, so wird der aus den Fronten resultierende Punkt  $gf_{mean}$  als Junction-Point innerhalb der Skeletts markiert.

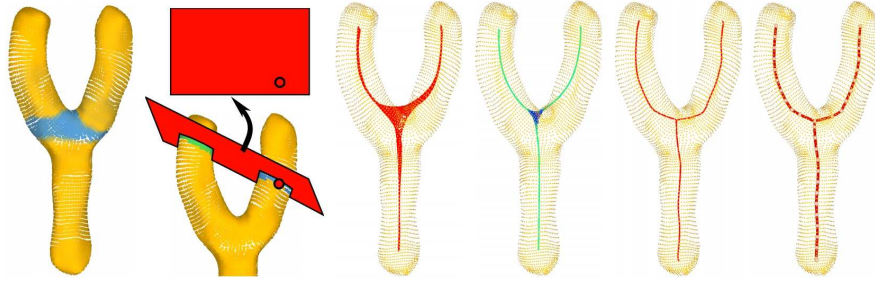


**Abbildung 2.11:**  $i = 0$  - Deformable Modell  $DM$  als Sphäre im Objektzentrum; Teilung der Fronten in  $k$  Teilfronten  $GF_{i,k}$ ; Jede Teilfront expandiert weiter; Der erzeugte Skelettgraph des Objektes

Die Iteration endet, sobald durch Expansion des Deformable Modells keine Front mehr verändert wird. Mittels der Funktion  $\kappa$  wird unter Verwendung der ermittelten Skelettpunkte der Skelettgraph erzeugt werden.

### 2.6.3 Curve Skeleton Extraction From Incomplete Point Cloud

Dieses Verfahren nutzt zur Ermittlung der Skelettpunkte die rotational symmetry axis (ROSA), was eine zylindrische Form des Objektes voraussetzt [TZCO09]. Jeder Punkt  $p$  der Objekt-Oberflächepunkte  $P$  ist definiert durch ein Tupel  $(pos, n_p)_P$ , wobei  $pos$  die Information über die Punktposition im  $\mathbb{R}^3$  und  $n$  die Normale welche sich aus der Nachbarschaft von  $p$  ableitet. Die Berechnung dieser Normalen wird im nachfolgenden Kapitel erläutert.

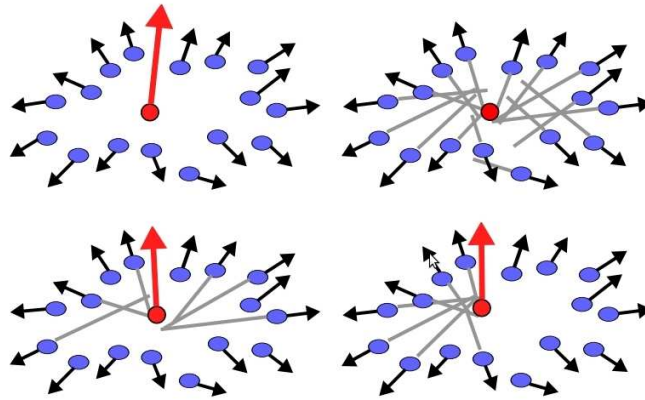


**Abbildung 2.12:** Objekt-Modell mit blau markiertem Joining-Bereich; Schnitt von Objekt und Schnittebene  $\eta$ ; Durch das Verfahren ermittelte potentielle Skelettpunkt (vor dem Thinning); Ergebnis des Thinnings - Hell grün gefärbt sind hier die Skelettpunkt welche durch Anwendung von  $\kappa$  eindeutig zu einem Skeletgraph führen. Der blaue Bereich markiert die Punktewolke welche im nächsten Schritt durch ihren Mittelpunkt ersetzt wird; Skelettpunkt vor Anwendung von  $\kappa$ ; Skeletgraph nach Anwendung von  $\kappa$  [TZCO09]

In jedem Punkt  $p_i$  wird eine Schnittebene  $\eta_i^*$  ermittelt, welche die lokale Rotations-Axen-Symmetrie, gegeben durch die benachbarten Punkte  $p_n$  in der Nachbarschaft  $N_p$  von  $p$ , mit der geringsten Abweichung abbildet. Dieses nicht lineare Optimierungs-Problem wird mit einem iterativen Ansatz gelöst. Man beginnt mit einer initialen Orientierung  $v_i^{t=0}$  der Ebene minimiert die Varianz der Winkel zwischen der Ebenen-  $n_{\eta_i}$  und den Punktnormalen  $n_p$ .

$$v_i^{t+1} = \operatorname{argmin}_{n_{\eta_i} \in \mathbb{R}^3, \|v\|=1} \operatorname{var}\{ \langle n_{\eta_i}, n_{p_j} \rangle : p_j \in N_p^{(t)} \}, t \geq 0 \quad (2.11)$$

Für die ideale Schnittebene  $\eta_{best}$  wird wie in Abbildung 2.13 gezeigt der Punkt  $q_{rosa}$  auf der ROSA anhand eines Geradenschnitte innerhalb einer  $\epsilon$ -Umgebung, um diese aus Punktnormale  $n_p$  und Punkt  $p$  konstruierten Gerade  $g$  ermittelt. Die so entstanden Punkte werden der Menge  $S_{pot}$  zugeordnet, welche die Menge der potentiellen Skelettpunkte repräsentiert. Diese Punktmenge wird wie in Schritt drei und vier der Abbildung 2.12 durch ein Thinning-Verfahren ausgedünnt. Bleiben



**Abbildung 2.13:** Diese Grafik zeigt die Oberflächenpunkte  $P$  auf der Schnittebene  $\eta$  als blaue Punkte mit ihren durch schwarze Pfeile illustrierten Normalen  $n_p$ . Der in zweiten Bild durch Geradenschnitte ermittelte Skelettpunkt ist hier mit seiner normalen rot dargestellt; Ermittlung des Skelettpunktes durch Schnitt der Geraden (grau dargestellt); Bild drei und vier zeigen die Ermittlung des Schnittpunktes bei unvollständigen Daten [TZCO09]

nach dem Thinning Regionen in Form einer Punktwolke erhalten werden diese (siehe Schritt 5 - Abb. 2.12)) durch ihren Mittelpunkt ersetzt. Dadurch erhält man die Skelettpunkte-Menge  $S$  mit der durch Anwendung von  $\kappa$  der Skelettgraph erzeugt wird.

Desweiteren zeigt die Abbildung 2.13, warum dieses Verfahren ebenfalls für unvollständige Punktwolke (Objektoberflächen-Daten) funktioniert. Ein Schnittpunkt zur Bestimmung des potentiellen Skelettpunkte kann bereits ermittelt werden, wenn mindestens zwei Objektoberflächenpunkte  $p$  in der Schnittebene  $\eta$  liegen. Steigt die Anzahl der Punkte  $p$  die in der Schnittebene liegen, so wird die Genauigkeit des potentiellen Skelettpunktes  $s_{pot}$  erhöht.



# Kapitel 3

## Eigener Ansatz

### 3.1 Theoretische Annahme

Zur Ermittlung eines Skelettes gilt folgende theoretische Annahme: Jeder Punkt  $p_i$  besitzt basierend auf den Positionen seiner benachbarten Punkten eine Normale  $n_i$ . Jeder mit  $p_i$  verknüpfte Skelettpunkt  $s_i$  liegt auf einer Geraden  $g_i$ , welche sich aus  $p_i$  und  $n_i$  herleitet. Da jeder Skelettpunkt mindestens drei ( $\mathbb{R}^3$ ) korrespondierende Punkte  $p_i$  der Objektoberfläche besitzt, folgt daraus, dass sich der Skelettpunkt  $s_i$  am Schnittpunkt dreier Geraden  $g$  befinden muss.

In jedem Punkt  $p_i$  lässt sich die Objektoberfläche durch eine Menge an Kurven  $C$  beschreiben. Jede Kurve  $c_i$  beschreibt die lokale Oberfläche in jeweils einer anderen Richtung. Voraussetzung für die hier getroffene Annahme ist, dass sich eben diese lokale Objektoberfläche durch mindestens eine konvex Kurve  $c_i$  beschreiben lässt.

Eine konvexe Oberfläche ist eine Eigenschaft, die für agierende Objekte erfüllt ist, was in den Abbildungen im Kapitel Evaluation belegt wird. Ist ein Kurve  $c_j$  konkav, und wird damit die Bedingung für die Berechnung des Skelettpunktes nicht erfüllt, so wird dies durch mindestens eine andere Kurve im Punkt  $p_i$  kompensiert, in dem die Geraden der Punkte dieser Kurve die erforderlichen Schnittpunkte liefern.

## 3.2 Normalisierung der Punktemenge

Um die Unterschiede zwischen verschiedenen Oberflächenpunktewolken im Bezug auf deren Auflösung, Wertebereich und Lage im  $\mathbb{R}^3$  zu minimieren und dadurch eine Vergleichbarkeit zu schaffen, wird die Punktewolke normalisiert. Hierzu wird eine Bounding Box ermittelt, die die Punktewolke komplett umschließt und deren Kanten alle die gleiche Länge und durch die Eckpunkten  $BV_{max}(x, y, z)$  und  $BV_{min}(x, y, z)$  beschrieben wird. Die Koordinaten des Eckpunkte für  $BV_{max}$  wird durch Bestimmung der maximalen Komponente  $max_P(x, y, z)$  aller Punkte  $p$  unter Berücksichtigung der drei Dimensionen  $x, y, z$  ermittelt. Analog werden auch die Koordinaten für  $BV_{min}$  anhand der minimalen Komponente  $min_P(x, y, z)$  bestimmt.

$$\begin{aligned} BV_{max}(max_P(x, y, z), max_P(x, y, z), max_P(x, y, z)) \\ BV_{min}(min_P(x, y, z), min_P(x, y, z), min_P(x, y, z)) \end{aligned} \quad (3.1)$$

Anhand des Bounding Box Zentrums wird die zur Translation der Punktewolke in den Ursprung des Koordinaten-System notwendige Translationsmatrix erstellt und alle Punkte mit dieser transliert. Im nächsten Schritt wird der Wertebereich der Punktewolke normiert und auf  $[-1, 1]$  abgebildet. Damit sind die Voraussetzung einer besseren Vergleichbarkeit geschaffen.

## 3.3 Normalen-Schätzung

Das Problem der Bestimmung der Normalen  $n$  für einen Punkt  $p$  auf der Objektoberfläche wird durch die Schätzung einer zur Oberflächen tangetiellen Ebene approximiert. Diese Schätzung wird durch die Lösung des Problems zur Bestimmung des minimalen quadratischen Abstands (Least-square plane fitting problem) der Ebene zur Objektoberfläche ermittelt.

### 3.3.1 Bestimmung der Punktnormalen durch PCA

In dem hier vorgestellten Ansatz wird zur Lösung dieser Probleme eine PCA (Principal Component Analysis) angewendet, welches die Normalenschätzung auf eine Analyse der Eigenvektoren und Eigenwerte der Covarianz-Matrix  $C$  der nächsten



Nachbarn des betrachteten Punktes reduziert [Rus09]. Für eine Nachbarschaft der Größe  $k$  ist die Covarianz-Matrix wie folgt definiert:  $\bar{p}$  ist der Centroid (Schwerpunkt) der Punktnachbarschaft.

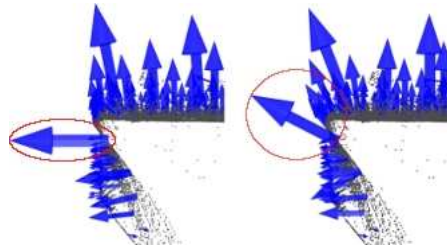
$$C = \frac{1}{k} \sum_{i=1}^k (p_i - \bar{p}) \cdot (p_i - \bar{p})^T, C \cdot v_j^{\rightarrow} = \lambda_j \cdot v_j^{\rightarrow}, j \in \{0, 1, 2\} \quad (3.2)$$

Die Approximation der Normalen  $n$  ergibt sich aus dem zum kleinsten Eigenwert  $\lambda_{min}$  korrespondierenden Eigenvektor  $v_{min}^{\rightarrow}$ . Um  $n$  in Abhängigkeit zum Blickpunkt  $v_p$  die Orientierung der Normalen zu bestimmen sind weitere Berechnungen notwendig. Da die Normale in dem hier vorgestellten Verfahren zusammen mit dem korrespondierend Oberflächenpunkt  $p$  in eine Geradengleichung umgeformt wird, ist die korrekte Orientierung nicht von Relevanz und muss rechnerisch nicht ermittelt werden.

### 3.3.2 Ideale Nachbarschaftsgröße der Normalenschätzung $\rho_e$

Die Nachbarschaftsgröße  $\rho_e$  wird durch eine festgelegte Anzahl von Nachbarn oder durch einen gewählten Radius bestimmt. Da alle Punkte  $p_i$  der Nachbarschaft in der Ermittlung des Nachbarschafts-Centroid  $\bar{p}$  und der Covarianz-Matrix  $C$  mit gleichem Gewicht eingehen, wird die Normalen-Schätzung durch die Wahl der Nachbarschaftsgröße beeinflusst. Wird die Nachbarschaft zu groß gewählt, so werden Punkte zur Berechnung herangezogen, die nicht die lokale Struktur der Oberfläche repräsentieren (Abb. 3.1) und somit die Normalenschätzung verfälschen. Kleine Oberflächendetails werden bei zu großen Nachbarschaften nicht richtig oder gar nicht in der Schätzung berücksichtigt. Ist die Nachbarschaft zu klein gewählt, führt ein vorhandenes Rauschen innerhalb der betrachteten Daten zu einer fehlerhaften Normalenschätzung.

Desweiteren muss die Dichte  $\tau$  der Punktwolke in Relation zur Objektgröße bei der Wahl des Parameter  $\rho_e$ , welcher den Radius oder die Anzahl der berücksichtigt benachbarten Punkte bestimmt, betrachtet werden. Während sich der Radius antiproportional zur Dichte der Punktwolke  $\tau$  verhält, entwickelt sich die Größe der Nachbarschaft proportional zu  $\tau$ . Verringert sich die Dichte  $\tau$ , liegen bei einem festgelegten Radius weniger Punkte innerhalb des Radius, welche zur Schätzung der Normalen verwendet werden. Dadurch wird diese auf weniger Komponenten (Oberflächen-Punkten) basiert ungenauer.



**Abbildung 3.1:** Die Abhängigkeit zur Nachbarschaftsgröße ist in dieser Abbildung sehr gut zu erkennen. Die markierte Normale weicht im zweiten Teil der Abbildung sehr stark von der ersten ab, welche in diesem Fall besser geschätzt ist

Hingegen wächst der Bereich für eine festgelegte Anzahl benachbarter Punkte. Da bei reduzierter und nicht reduzierter Dichte  $\gamma$  die Normalen-Schätzung auf einer identischen Anzahl an Oberflächen-Punkten basiert, bleibt in diesem Fall die Genauigkeit erhalten.

### 3.3.3 Reduktion der Punktdichte (Punktanzahl)

Die Punktdichte (Punktanzahl) wird über einen Reduktionsfaktor  $\gamma$  gesteuert. Die Punktanzahl der reduzierten Punktwolke  $numPoint_{redu}$  ergibt sich aus dem Produkt der Anzahl der Punkte  $numPoint$  vor der Reduktion und  $\gamma$ . Der Quotient von  $numPoint$  und  $numPoint_{redu}$  ergibt die Bereichsgröße, aus der jeweils ein Punkt anhand einer Zufallszahl für die reduzierte Punktwolke extahiert wird. Sind die Objekt-Oberflächen-Punkte nicht gleichmäßig auf der Oberfläche verteilt, was meist für synthetisch erzeugte Datensätzen zutrifft, funktioniert diese Art der Dichte Reduktion nicht und es muss ein Voxel-Gitter mit einer festgelegten Auflösung und einer vorherigen Normalisierung des Objektraums verwendet werden.

Da die Anzahl der Objektoberflächenpunkte die Anzahl der Normalen-Schätzungen sowie Normalenschnitte, welche im folgenden Abschnitt erläutert werden, bestimmt, steht sie im direkten Zusammenhang zur benötigten Berechnungsdauer.  $\gamma$  muss also so gewählt werden, dass der im Skelett abgebildete Detailgrad der Objekt-Oberfläche den jeweiligen durch die Weiterverarbeitung bestimmten Anforderungen genügt andererseits aber der Rechenaufwand moderat bleibt.

## 3.4 Bestimmung der Normalenschnitte

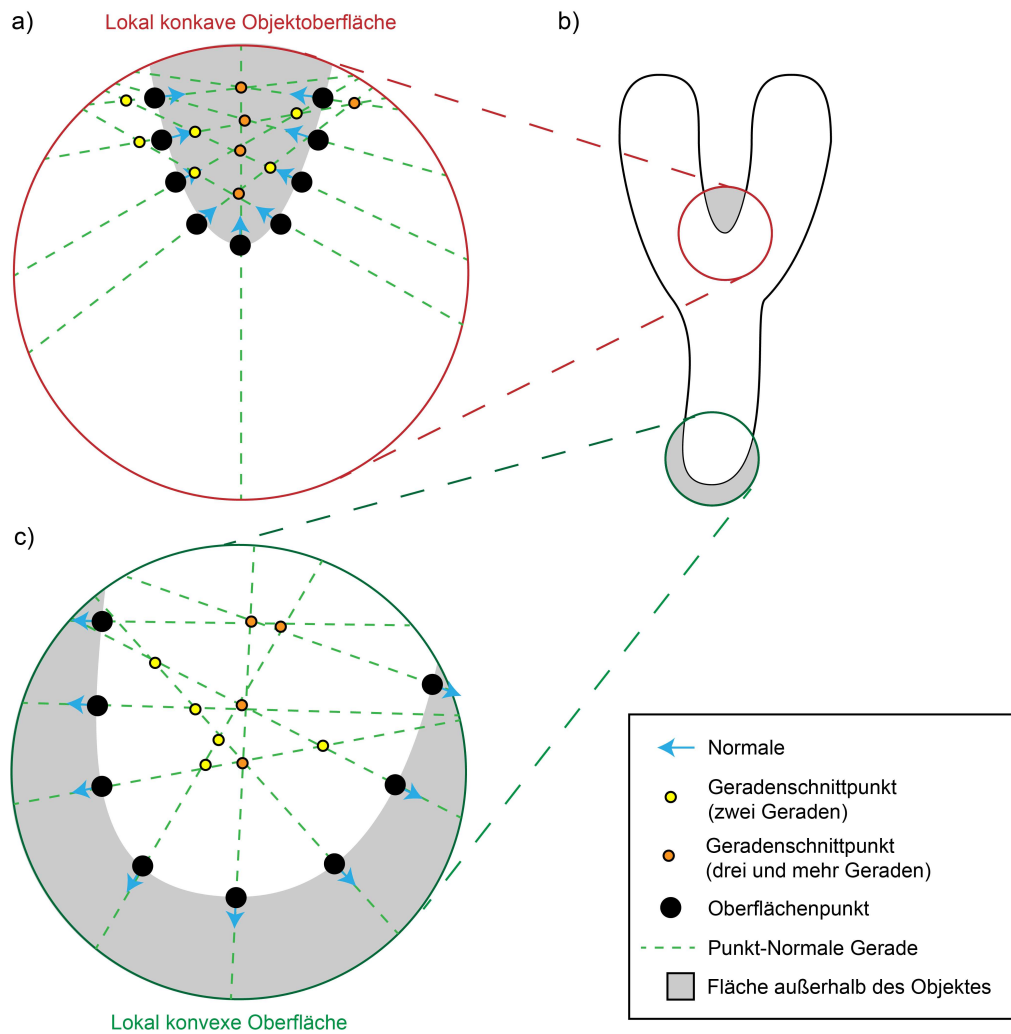
Für jeden Punkt  $p$  und seine zugehörige Normale  $n$  lässt sich eine Gerade  $g$  erzeugen. Diese Gerade wird mit allen Geraden  $g_j$  der zu  $p$  benachbarten Punkte  $p_j$  geschnitten (Abb. 3.2). Der Eingabeparameter  $\rho_c$  gibt die Größe dieser Nachbarschaft an. Bei Schnittpunkt-Berechnung der Geraden wird eine  $\epsilon$ -Umgebung um beide Geraden berücksichtigt, in der berechnete Schnittpunkt liegen können, um gegenüber fehlerhaften Normalenschätzung tolerant zu sein. Für jeden Schnittpunkt  $nc_i$  der Element der Menge der potentiellen Skelettpunkte  $S_{pot}$  ist, gilt  $nc_i \in BB_o$ , wobei  $BB_o$  die Bounding Box der Punktwolke im  $\mathbb{R}^3$  ist. Desweiteren gilt, dass der euklidische Abstand zwischen  $p$  und dem Schnittpunkt  $nc_i$  mit geringer Toleranz dem von  $p_j$  und  $nc_i$  entsprechen muss, um der Definition 2.3 aus Kapitel zwei zu genügen. Damit ferner ebenfalls die theoretische Annahme dieses Ansatzes, dass jeder Skelettpunkt im  $\mathbb{R}^3$  auf dem Schnittpunkt von mindestens drei Geraden liegt, wird für jede Nachbarschaft der Schnittpunktberechnung ein Schnittpunkt-Bereich bestimmt. Dieser basiert auf den gefundenen Schnittpunkten. Liegt ein Schnittpunkt außerhalb dieses Bereiches, wird er verworfen.

### 3.4.1 Konkave/Konvexe Ausprägung lokaler Objektoberfläche

Dieser Ansatz der Skelettpunkt Herleitung hat nur für eine konvexe Ausprägung einer lokalen Objektfläche potentielle Skelettpunkte als Ergebnis, da wie in Abbildung 3.2 illustriert alle Schnittpunkte für den konkaven Fall außerhalb des Objektes liegen und damit keine potentiellen Skelettpunkte sind.

Ist ein Objekt abgeschlossen, existiert zur lokal konkaven Oberfläche eine korrespondierende lokal konvexe Oberfläche, die die fehlenden Schnittpunkte liefert (Abb 3.3). Ein Objekt ist abgeschlossen, wenn durch die Objektbegrenzung  $U$  bestimmt werden kann welche Punkte innerhalb und welche außerhalb des Objektes liegen. Für unvollständige Objekt-Daten ist diese ebenfalls erfüllt, jedoch werden lokal konkave Oberflächen, deren korrespondierende konvexe Oberflächen Abschnitte Teil der fehlenden Daten sind, nicht kompensiert.

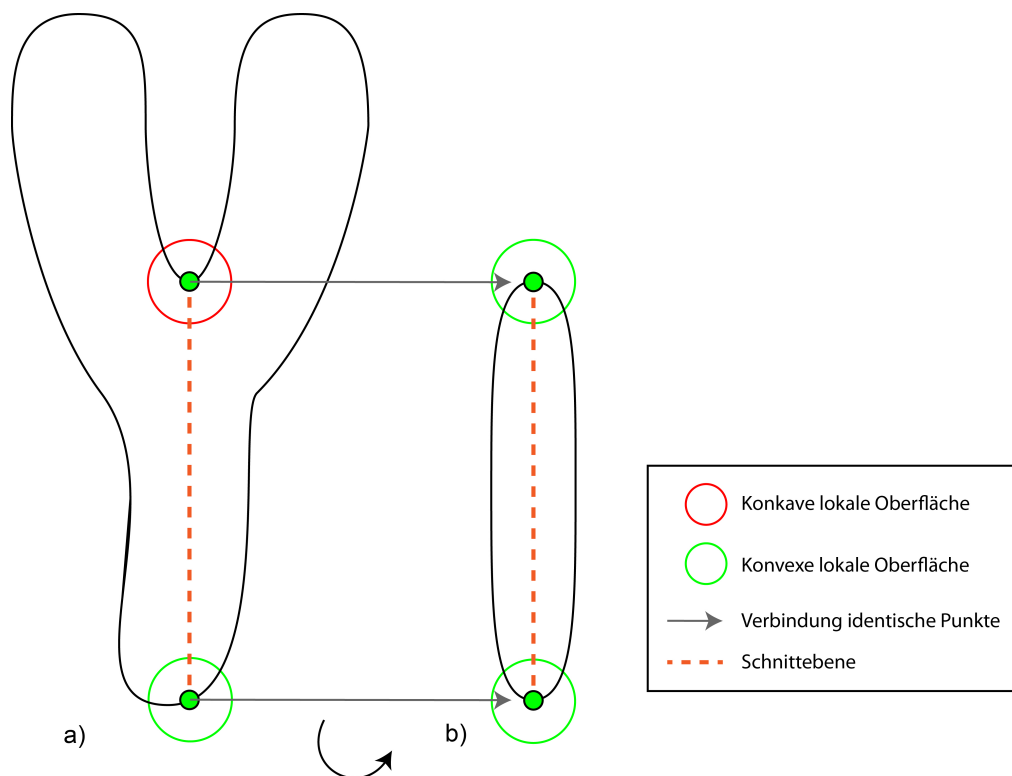
Desweiteren wird die Skelettapproximation auf artikuliernde und skelettähnliche Objektes angewandt, welche lokale konvexe Oberflächen Strukturen aufweisen.



**Abbildung 3.2:** a) Schnittpunktermittlung für eine lokal konkave Objektoberfläche ohne Berücksichtigung des identischen euklidischen Abstands zwischen Punkten und Schnittpunkt; b) Gesamtansicht des Objektes mit zwei Detailausschnitten; c) Schnittpunkte (Potentielle Skelettpunkte) und deren Ermittlung für eine lokale konvexe Objektoberfläche

### 3.4.2 Idealen Nachbarschaftsgröße $\rho_c$ für die Normalenschnitt-Berechnung

$\rho_c$  bestimmt die Nachbarschaftsgröße für die die Normalenschnitte berechnet werden. Da diese Größe die Anzahl der Normalen-Erstellung sowie deren Schnittbe-



**Abbildung 3.3:** a) Ansicht des Objektes in  $\mathbb{R}^3$  mit lokal konvexer und konkaver Objektoberfläche, b) Gedrehte Ansicht Korrespondierende lokal konvexe Objektoberflächen

rechnung direkt beeinflusst, muss auch hier auf die Wahl des richtigen Parameterwertes  $\rho_c$  geachtet werden.

Durch  $\rho_c$  wird der im Skelett abgebildete Detailgrad der Objektoberfläche gesteuert. Wählt man  $\rho_c$  in Relation zur Anzahl der Oberflächenpunkte  $P$  zu groß, ergibt sich kein lokaler Bereich und es kommt zum Anstieg der fehlerhaften potentiellen Skelettpunkte in  $S_{pot}$ , während analog der Rechenaufwand steigt, da mehr Geradenschnitte durchgeführt werden müssen. Ist  $\rho_c$  zu klein gewählt ergeben sich nur wenige oder gar keine Schnittpunkte, da sich aus der geringeren Punktzahl auch eine geringe Anzahl an Geraden  $g$  ergibt, und dadurch die Wahrscheinlichkeit eines Geradenschnittes sinkt.

In Kombination mit großem  $\gamma$  entsteht durch ein klein gewähltes  $\rho_c$  folgende Problematik: Die Geraden treffen sich durch den sehr geringen Winkel zwischen den Normalen meist außerhalb des Objektes. Dies tritt auf, wenn die Punktdichte  $\gamma$  so hoch ist das für jeden Punkt in der Nachbarschaft von  $p$  die gleiche Normale  $n$  geschätzt wird, und somit parallel sind und sich nicht schneiden.

### 3.5 Basis Parameter: $\gamma, \rho_e, \rho_c$

Wie in der Einleitung schon erwähnt, ist das Ziel des in dieser Arbeit vorgestellten Verfahrens, ein Kurvenskelett zu ermitteln. Das Verfahren soll hierzu die Möglichkeit geben, anhand von Parametern den Detailgrad der Approximierung zu regeln.

Die Parameter können nicht isoliert betrachtet werden, da sie, wie schon in den vorangegangenen Abschnitten erläutert, im Kausalzusammenhang stehen. Im Kapitel Test und Evaluation wird dieser Kausalzusammenhang anhand der durchgeführten Test visuell verdeutlicht. Es wird gezeigt wie durch gezielte Wahl der Parameter für verschiedene Parameter-Tupel  $(\gamma, \rho_e, \rho_c)_{normals}$  ähnlicher Ergebnisse erzielt werden können.

Mit  $(\gamma, \rho_e, \rho_c)_{normals}$  wird das Zwischenergebnismenge  $S_{pot}$  der potentiellen Skelettpunkte ermittelt, welche der nachfolgende Abschnitt des Algorithmus zur Erstellung des Skelettgraphen nutzt. Daher haben diese Parameter ebenfalls hohen Einfluss auf die Güte des Gesamtergebnisses des Verfahrens.

## 3.6 Erstellung des Skelettgraphen

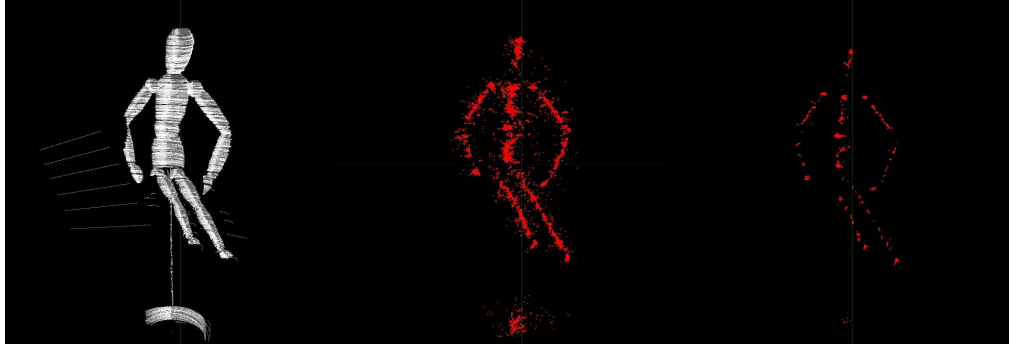
Der zweite Abschnitt des Verfahrens behandelt das Problem der Skelettgraph-Erstellung auf Basis einer gegebenen Punktemenge potentieller Skelettpunkte  $S_{pot}$ . Das angestrebte Ergebnis ist ein durchgängiger Graph der durch alle Skelettpunkte  $S$  einer Teilmenge von  $S_{pot}$  verläuft.

### 3.6.1 Reduktion der potentiellen Skelettpunkte $S_{pot}$

Um die Bewertung der Punkte in  $S_{pot}$  zu ermöglichen, ist es nötig anhand eines Voxel Gitters die Bereiche hoher Punktdichte zu filtern. Die für den Bereich ermittelte Dichte wird als Attribut für den Punkt mit dem geringsten Abstand zum Zentrum des Voxels gesetzt und der diskreten Skelettpunktemenge  $S_{dis}$  hinzugefügt (Abb. 3.6).

Als Auflösung des Gitters wird  $500 \times 500 \times 500$  gewählt um die Mittelwertfilterung der Diskretisierung möglichst gering zu halten. Aus jedem belegten Voxel  $V_{set}$  wird ein diskreter Skelettpunkt  $s_{dis}$  mit der Punktdichte  $density(V_{set})$ . Die

Punktkoordinaten bleiben erhalten. Daraus ergibt sich für jeden Punkt folgendes Tupel  $s_{dis}(x, y, z, density(V_{set}))$ .



**Abbildung 3.4:** Unveränderte Oberflächen-Punktewolke; Ergebnis des ersten Schritt des Verfahrens (Normalenschnitte  $\tau = 0.01, \rho_e = 9, \rho_c = 30$ ); Reduktion der potentiellen Sklettpunkte-Menge mit einem Voxel Gitter

### 3.6.2 Ermittlung des Growing-Initial Zustands

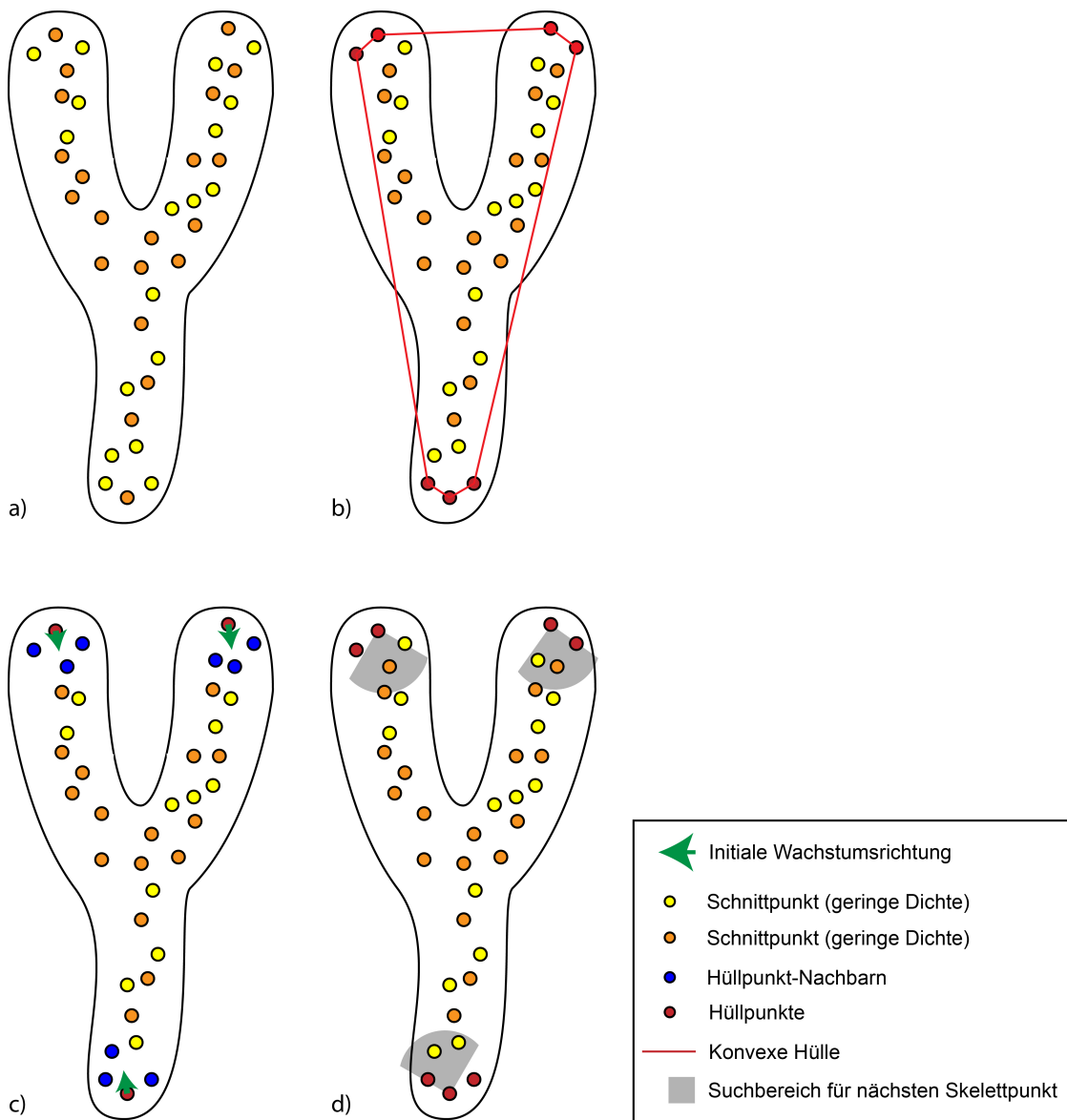
Für den Growing Algorithmus müssen zunächst die Startpunkte  $IP$  und deren zugehörige angenommene Wachstumsrichtung  $DIR_{IP}$  ermittelt werden. Als Menge der Startpunkte  $IP$  wird in diesem Verfahren die Punkte der Konvexen Hülle von  $S_{dis}$  verwendet. Die Konvexe Hülle von  $S_{dis}$  erzeugt der in einer externen Bibliothek implementierte Quickhull-Algorithmus [BDH96].

$DIR_{IP}$  eines Startpunktes  $IP$  wird durch den Vektor zwischen  $IP$  und dem Mittelpunkt der ( $k = 10$ ) nächsten Nachbarn beschrieben. Diese Nachbarschaftsgröße wird gewählt da,  $IP$  die Punktmenge  $S_{dis}$  in einer Richtung abschließt und damit in dieser keine benachbarten Punkte besitzt.

Ist jedem  $IP$  eine initiale Richtung  $DIR_{IP}$  zugeordnet, ist der Initial-Zustand eine Liste von Tupeln  $(IP, DIR_{IP})$  für die Erzeugung des Skelettgraphen erreicht.

### 3.6.3 Iteratives Wachstum des Skeletgraphen

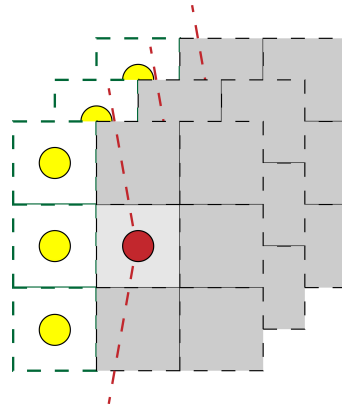
Der Skeletgraph  $SG$  setzt sich aus  $SG_i$  Teil-Skeletgraphen zusammen, die in jedem Iterationsschritt wachsen.  $SG$  und  $SG_i$  sind hierbei definiert als eine Liste aus Punkten der Menge  $S_{dis}$  und ihren jeweiligen Verbindungen (Punkt zu Punkt-Verbindungen) innerhalb des Graphen.



**Abbildung 3.5:** a) zeigt die diskretisierte Punktmenge  $S_{dis}$  mit unterschiedlichen Schnittpunktdichten (gelb - gering / orange - hoch)  
 b) Ermittlung der konvexen Hülle für  $S_{dis}$  und damit der Startpunkte  $IP$   
 c) Bestimmung der initialen Wachstumsrichtung für die Punkte  $IP$  anhand der nächsten Nachbarn (blau markiert)  
 d) Illustration des Suchbereichs für den besten nächsten Nachbarn für die Punkte  $IP$

Die Anzahl der Verknüpfungen ergibt sich aus den in Kapitel zwei vorgestellten Kategorien der Skelettpunktes (2.4 auf Seite 19). So besitzen End-Punkte, die





**Abbildung 3.6:** Diskretisierung der der Punktwolke durch ein Voxel Gitter (Scharzgestrichelt). Die gelben Punkte illustrieren die maximale Anzahl der Nachbarn für einen Punkt (rot markiert) der konvexen Hülle (rot-gestrichelte Linie) und veranschaulichen warum diese Anzahl im  $\mathbb{R}^3$   $k = 10$  gewählt werden muss.

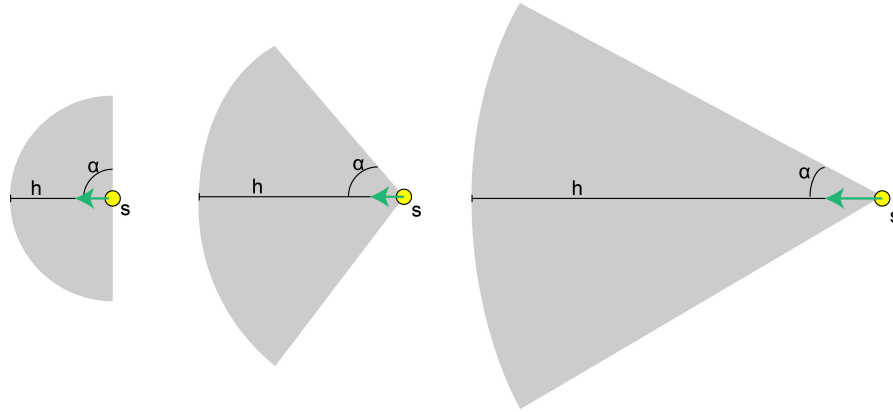
hier der Menge  $IP$  entsprechen, eine Verbindung. Verbindungs-Punkte an den sich eine beliebige Menge Teil-Skeletgraphen  $SG_i$  zusammenschließen, besitzen mindestens drei Verbindungen zu anderen Skelettpunkten. Alle weiteren Skelettpunkte gehören zur Kategorie Mittelpunkt (Branch-Point) und besitzen genau zwei Verbindungen. In jedem Iterationsschritt ist eine Liste von Tupeln gegeben, welche aus einem bereits dem Skeletgraphen hinzugefügten Punkt  $s$  aus  $S_{dis}$  und einer ihm zugeordneten Wachstumsrichtung  $dir(s)$  bestehen. Die Liste wird im folgenden als aktive Wachstumsfront  $F$  bezeichnet. Jeder hinzugefügter Punkt  $s$  ist Teil der Menge der Skelettpunkte  $S$ .

$$F = \langle (s_0, dir(s_0)), (s_1, dir(s_1)), \dots, (s_n, dir(s_n)) \rangle \quad (3.3)$$

Für jeden Tupel dieser Liste wird jeweils ein bester Nachbar gesucht. Hierzu wird ein kegelförmiger Bereich wie in Abbildung 3.7 bestimmt durch die Wachstumsrichtung  $dir(s)$ , dem Winkel  $\alpha$  und Kegelhöhe  $h$ , vom Punkt  $s$  ausgehend aufgespannt. Alle Punkte aus  $S_{dis}$  die in diesem Bereich  $SB$  liegen, werden in die Suche miteinbezogen. Der beste Nachbarschaftspunkt aus diesem Bereich  $p_{bestNeigh}$  ist der, für den gilt, das Produkt aus Punktdichte  $density(p_{bestNeigh})$  und euklidischem Abstand zum Zentrum der Punktwolke  $center_{S_{dis}}$  ist maximal.

$$p_{bestNeigh} = p_i \in SB : \max(density(p_i) * d(p_i, center_{S_{dis}})) \forall p \in SB \quad (3.4)$$

Der kegelförmige Suchbereich wird durch  $dir(s)$ ,  $\alpha$  und  $h$  bestimmt. Während  $dir(s)$  gegeben ist, werden die Parameter  $\alpha$  und  $h$  dynamisch gesetzt. So wird initial für jeden Punkt  $s$  der nächste Nachbar in einem möglichst geringen Abstand gesucht und damit ein sehr kleines  $h$  gewählt. Hingegen kann für diesen geringen Abstand der Öffnungswinkel des Kegels, welcher durch  $dir(s)$  und den Winkel  $\alpha$ , der die Abweichung von dieser Richtung beschreibt, sehr groß gewählt werden.



**Abbildung 3.7:** Diese Grafik zeigt, wie sich der Suchraum für die Suche nach dem besten nächsten Nachbarn verändert, wenn kein Nachbar gefunden wird. Initial (links):  $h = \frac{1}{250}$ ,  $\alpha = 90^\circ$ ; Wächst  $h$  so wird der Winkel  $\alpha$  geschrumpft um nicht zu sehr von der Wachstumsrichtung (grüner Pfeil) abzuweichen

Der durch die Diskretisierung (3.6.1 auf Seite 38) gegebenen kleinsten Abstand  $\frac{1}{500}$  bestimmt den für  $h$  angenommenen Wert. Es wird  $h = \frac{1}{250}$  angenommen damit alle direkten Nachbarn innerhalb des Gitters bei der Suche berücksichtigt werden. Mit  $\alpha = 90^\circ$  werden alle Punkte im vorderen durch die Wachstumsrichtung bestimmten Halbraum des Punktes betrachtet. Befindet sich in dem so festgelegten Suchraum kein Punkt der Menge  $S_{dis}$  und kann daher kein bester nächster Nachbar gefunden werden, wird der Suchraum vergrößert, indem der Wert von  $h$  verdoppelt und der Winkel von  $\alpha$  verkleinert wird. Damit kann ein bester nächster Nachbar  $p_{bestNeigh}$  auch dann ermittelt werden, wenn die Punktmenge  $S_{dir}$  unvollständig ist. Eine Regel für die Wahl von einem idealen  $\alpha$  im Bezug auf  $h$  konnte im Rahmen dieser Arbeit nicht ermittelt werden.

Ist ein bester Nachbar für den Punkt  $s$  gefunden, so wird das Tupel zugehörige Tupel aus der Liste  $F$  entfernt und durch ein neues Tupel für  $p_{bestNeigh}$  hinzugefügt. Die Richtung  $dir(p_{bestNeigh})$  ergibt sich aus der für die Suche verwendeten normierten Richtung  $dir(s)_{norm}$  und dem normierten Vector  $SP_{bestNeigh}_{norm}$ .

$$\mathit{dir}(p_{\mathit{bestNeigh}}) = \mathit{dir}(s)_{\mathit{norm}} + \mathit{SP}_{\mathit{bestNeigh}_{\mathit{norm}}} \quad (3.5)$$

Damit werden durch das Einbringen der alten Richtung Sprünge aus der Wachstumsrichtung vermieden. Ansonst kann es dazu kommen das ein Skelett-Teilgraph  $SG_i$  in einem lokalen Maximum der Punktmenge  $S_{\mathit{dis}}$  weiterwächst und sich nicht mit den anderen Skelettteilgraphen verbindet.

$p_{\mathit{bestNeigh}}$  ist ein Verbindungspunkt zweier Skelettteilgraphen  $SG_i$ , wenn gilt, dass  $p_{\mathit{bestNeigh}} \in S$ . In diesem Fall wird die Verbindung zu  $s$  dem entsprechenden Tupel des Skelettgraphen  $SG$  hinzugefügt.  $s$  wird aus der aktiven Front  $F$  gelöscht, es wird aber kein neues Tupel aus Punkt und Richtung hinzugefügt. Daraus ergibt sich, das der Growing-Prozess abgeschlossen ist sobald die aktive Front  $F$  keine weiteren zu untersuchenden Tupel mehr enthält.

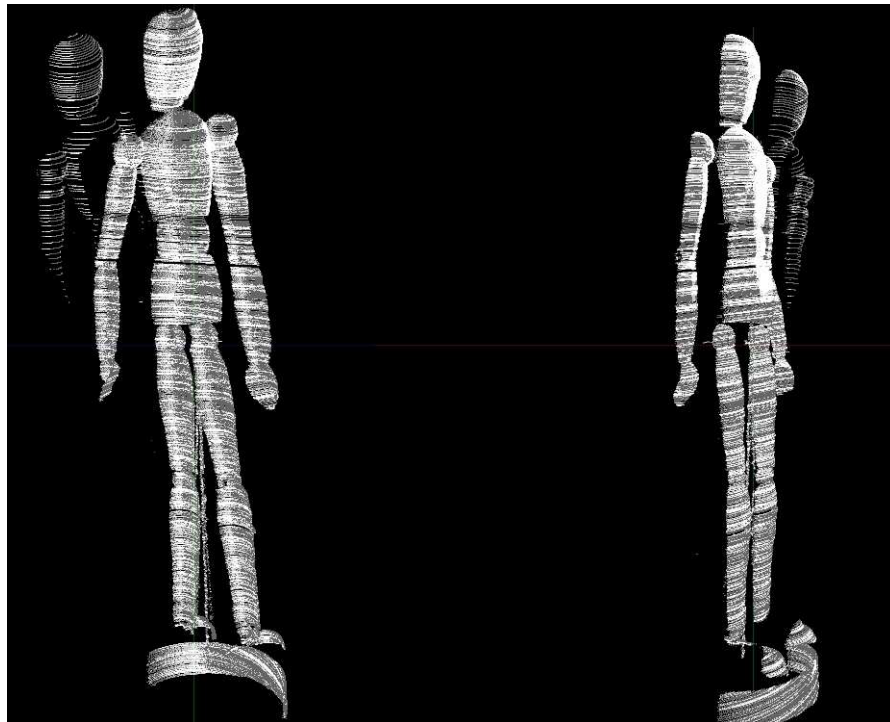
Der Skelettgraph  $SG$  enthält alle Punkte mit ihren jeweiligen Verbindung aus den Teilskelettgraphen  $SG_i$  und den zusätzlichen Verbindungen, welche durch die Vereinigung der Teilgraphen entstehen. Alle Punkte in  $SG$  besitzen, da sie der Normalenschnittpunkt-Menge  $S_{\mathit{pot}}$  angehören, zwei korrespondierenden Punkte der Objektoberfläche  $P$  aus deren Normalenschnitt sie resultieren.

## 3.7 Vergleich mit existierenden Verfahren

In diesem Abschnitt sollen die Ansätzen „On-the-fly Curve-skeleton Computation“ [SLSK07] und „Curve Skeleton Extraction From Incomplete Point Cloud“ [TZCO09] mit dem hier vorgestellten Verfahren in Bezug auf Robustheit gegenüber unvollständiger Objekt-Oberflächen-Daten, Komplexität der Ausführung und Erweiterbarkeit. Vorallem die Anwendbarkeit auf unvollständige sowie fehlerbehaftete Datensätze soll hier im Vordergrund stehen, da beim Einsatz in realen Anwendungen die Daten von Sensoren (z.B. Tiefensensoren) erzeugt werden und daher nicht einem synthetisch erzeugten dreidimensionale Modell des Objektes entsprechen.

### 3.7.1 On-the-fly Curve-skeleton Computation

Das in Kapitel zwei erläuterte Verfahren setzt vollständige abgeschlossene Objekt-Oberflächen-Daten voraus. Wie bereits erläutert, basiert es auf einem aus dem



**Abbildung 3.8:** Sensordaten eines Teifensensors: Der Datensatz weist eine Spiegelung des Kopfes auf, welche das gescannte Objekt nicht besitzt. Das Objekt um  $90^\circ$  an der Y-Achse gedreht (rechts) zeigt, dass durch den Frontal-Scan alle Informationen über die Rückansicht fehlen.

Zentrum der Oberflächenpunktewolke heraus wachsender Punktewolke *DM* (Deformable Model), für die anhand der Mittelpunkte eines ungeteilten Oberflächen-Segments (Front) die Skelettpunkte bestimmt werden. Verwendet man für das Verfahren die in der Abbildung 3.8 gezeigten Tiefendaten als Basis, wächst das Deformable Model aus der Rückseite des Objektes heraus, da es hier nicht auf Objektoberflächenpunkte trifft. Dadurch werden die Fronten verfälscht, die sich für diese Daten durch die fehlende Rückansicht und die nicht existierende Begrenzung nicht in Subfronten zur Skelett-Approximation verschiedener Teilbereiche der Objektes aufteilen. Während bei dem in dieser Arbeit vorgestellten Normalenschnitt-Verfahren die Berechnungen der Normalen und der Normalenschnitte unabhängig voneinander sind, ist in diesem Verfahren nur das Wachstum der Subfronten voneinander unabhängig. Die Berechnung kann also nicht komplett parallelisiert werden. Der Vorteil dieses Verfahrens gegenüber dem hier vorgestellten liegt darin, dass die Skelettpunkte und der Skelettgraph in einem Arbeitsschritt ermittelt werden und somit die im Normalenschnitt-Verfahren benötigte Graphen-Erstellung entfällt. Diese Trennung Arbeitsschritte bieten jedoch im Gegenzug eine bessere Mög-

lichkeit der Erweiterung, da der hier verwendete Graph-Erstellungsalgorithmus leicht durch einen anderen ersetzt werden kann.

### 3.7.2 Curve Skeleton Extraction From Incomplete Point Cloud

Der Ansatz [TZCO09] ist der Ursprung der hier umgesetzten Idee. Der Unterschied zu dem hier vorgestellten Normalenschnitt-Verfahren besteht darin, dass die Ermittlung der idealen Schnittebene entfällt und daher der Grad der Komplexität sinkt. Ungenauigkeiten bei der Normalenschätzung haben direkte Auswirkungen auf die Berechnung der Schnittebene und damit höheren Einfluss auf den ermittelten Skelettpunkt. Beide Verfahren trennen die Extraktion der Skelettpunkte von der Erstellung des Skelettgraphen und dünnen die ermittelten Skelettpunkte zur Erstellung des Graphen aus.



# Kapitel 4

## Implementierung

Das Verfahren ist als Konsolen Anwendung realisiert, die vier Eingabeparameter besitzt, welche beim Aufruf in einer festgelegten Reihenfolge übergeben werden müssen. Das Ergebnis wird in einem Fenster als interaktive 3D Ansicht angezeigt und die Ergebnisdaten in eine beim Aufruf festgelegte Datei geschrieben.

### 4.1 Modularer Aufbau

Das Programm ist intern modular aufgebaut und setzt sich aus vier Klassen zusammen. Einer Hauptklasse SkelApprox, die den Gesamt Ablauf des Verfahrens steuert und die Eingabe Parameter an die Funktionen, welche diese benötigen, übergibt.

#### 4.1.1 Daten einlesen

Zum Einlesen der Daten wird die Klasse DataReader genutzt, die es ermöglicht Punktdaten aus den Dateiformaten „ply“, „stl“ und „off“ in das Programm zu laden. Diese Klasse verwendet die externe Bibliothek VCG. VCG ist eine C++ Bibliothek welche vom Visual Computing Lab am ISTI (Institut der Information Technologie, Italien) zur Verarbeitung von 3D Gitternetzen (Meshes) entwickelt wurde.

### 4.1.2 Verarbeitung und Datenhaltung

Das Programm besitzt zur Verarbeitung zwei Datenhaltungsstrukturen. Die Datenhaltungsstruktur der hier eingesetzten externen Bibliothek PCL (Point Cloud Library).

PCL ist eine “open project „für die Verarbeitung von 3D Punktwolken. PCL ist eine Open Source Software, welche unter den Bedingungen der BSD Lizenz genutzt werden darf. PCL bietet als Framework viele Implementierungen von State of the Art Algorithmen zu 3D Punktwolken an [RCG11].

Die zweite Datenhaltungsstruktur ein Punktlist basierend auf der C++ Standard Klasse Vector (PointList) und wurde eingeführt, um den einzelnen Punkten mehr Informationen hinzufügen zu können. Das ist in der PCL Datenstruktur nicht möglich, da nicht alle in PCL implementierten Algorithmen eine von der PCL-Standardstruktur abweichende Datenhaltungsstruktur verarbeiten können.

Alle verarbeitenden Funktionen der Algorithmen sind in der Klasse SkelAlgorithm implementiert. Die Klasse hält selber keine Daten vor, sondern erhält diese beim Funktionsaufruf und liefert sie nach der Verarbeitung wieder an die Hauptklasse SkelApprox zurück. Die Punktdaten werden immer als Zeiger übergeben um unnötige Kopiervorgänge zu verhindern und damit verbundene Rechenaufwände zu verhindern.

Die Berechnung der Normalen und der Normalen Schnittpunkte sind unter Verwendung der C++ Multi-Processing Bibliothek OpenMP parallelisiert, so dass je nach Anzahl der Prozessor-Kerne eine entsprechende Anzahl Normalenschätzungen vorgenommen oder für eine entsprechende Anzahl an Nachbarschaften die Normalenschnitte ermittelt werden können.

### 4.1.3 Datenausgabe

Die Daten werden visuell in dreidimensionaler Form mit Interaktionsmöglichkeit unter Verwendung des OpenGL Framework GLUT in einem Fenster ausgegeben. So ist es möglich verschiedene Berechnungsergebnisse ein oder auszublenden. Zusätzlich werden die berechneten Daten in eine zum Aufruf des Verfahrens angegebene Datei als Komma separierte Werte geschrieben.



## 4.2 Parameter

Zum Aufruf des Programms müssen folgende Parameter angegeben werden:

- Eingabedatei Punktwolke (ply, stl oder off)
- Reduktionsfaktor  $\gamma$  der Punktemenge  $]0, 1]$  (für 1 erfolgt keine Reduktion)
- Größe der Nachbarschaft zur Normalenschätzung  $\rho_e$  (Ganze Zahlen, größer 0, kleiner der Anzahl der Punkte in der zu bearbeitenden Punktemenge)
- Größe der Nachbarschaft zur Berechnung der Normalenschnitte  $\rho_c$  (Ganze Zahlen, größer 0, kleiner der Anzahl der Punkte der zu bearbeitenden Punktemenge)
- Name der Ausgabedatei

## 4.3 Schrittweiser Programmablauf

Hier ist der schrittweise Ablauf des Programmes mit den jeweiligen Klassen und aufgerufenen Funktionen angegeben. Der hier angegebene Ablauf ist analog in der Hauptklasse des Programms implementiert.

- `DataReader.getVertices`  
Laden der Eingabe Daten und Überführung in die PCL Datenhaltungsstruktur
- `SkelAlgorithm.normalizePointCloud`  
Normalisierung der Punktwolke auf  $[-1, 1]$
- `SkelAlgorithm.reduceCloudSize`  
Reduktion der Punktwolke unter Berücksichtigung von  $\gamma$
- `SkelAlgorithm.calculateNormals`  
Berechnung der Punktnormalen für die reduzierte Punktwolke unter Berücksichtigung von  $\rho_e$
- `SkelAlgorithm.calculateHullPoints`  
Bestimmung der Hüllpunkte für die reduzierte Punktwolke

- `SkelAlgorithm.getNormalIntersections`  
Berechnung der Normalen unter Berücksichtigung von  $\rho_c$  und den ermittelten Hüllpunkten
- `SkelAlgorithm.gridFilter`  
Filterung der Normalenschnitte durch ein VoxelGitter
- `SkelAlgorithm.enrichPoints`  
Gefilterte Schnittpunkte mit Informationen anreichern: Abstand zum Zentrum, Schnittpunktdicht im Voxel, minimaler Abstand zur Objektoberfläche, korrespondierende Oberflächenpunkten, Anzahl der Voxelgitter Nachbarn in denen Schnittpunkte liegen
- `SkelAlgorithm.growSkeleton` (auskommentiert)  
Skelett-Growing-Algorithmus für die angereicherten gefilterten Schnittpunkte
- `SkelAlgorithm.saveData`  
Speicherung der angereicherten gefilterten Schnittpunkte in der angegebene Datei
- `Dataviewer.add`  
Originalpunktewolke, reduzierte Punktewolke, Berechnete Punktnormale, ungefilterte und gefilterte Normalenschnitte werden an den Viewer zur Anzeige übergeben
- `Dataviewer.show`  
Öffnet das Fenster zur Ansicht des Berechnungsergebnisses

# Kapitel 5

## Test und Evaluierung

In diesem Kapitel werden die Ergebnisse des Verfahrens vorgestellt und evaluiert. Da der Graph-Growing Algorithmus zwar implementiert ist, aber nicht zu dem gewünschten bzw. präsentablen Ergebnissen führt, wird hier nur die Berechnung der Skelettpunkte anhand des Normalenschnitt-Verfahrens evaluiert. Durch die Evaluierung soll geklärt werden, wie die Parameter  $\gamma, \rho_e, \rho_c$  gewählt werden müssen um ein gutes Ergebnis zu erzeugen, welches in einem Graph-Growing -Verfahren zu einem Skelettgraph verbunden werden kann.

Kurvenskelette sind zwar von der Medialen Achsen Transformation abgeleitet, besitzen selber aber keine allgemeingültige Basis Definition, wie zum Beispiel die von Blum. Daher werden die Ergebnisse hauptsächlich auf visueller Ebene bewertet.

### 5.1 Testdaten

Die Testdaten bestehen aus einem syntetisch erzeugtes Objekt mit einer kompletten geschlossenen Oberflächenbeschreibung. Die Oberfläche wird durch Punkte im  $\mathbb{R}^3$  beschrieben. Die Punkte sind nicht gleichmäßig auf der Oberfläche verteilt, sondern richten sich in der Dichte nach der Oberflächenstruktur des Objektes.

Desweiteren werden Daten aus einem Tiefenscan verwendet. Hierbei handelt es sich um eine unvollständige Beschreibung der Objektoberfläche. Es liegen 20 Aufnahmen aus verschiedenen Ansichten mit jeweils unterschiedlichen Posen des

Objektes vor. Alle Testdaten werden mit der in Kapitel drei vorgestellten Methode auf den Wertebereich  $[-1, 1]$  normiert und in den Ursprung verschoben. Die Testdaten beschreiben jeweils nur das Objekt. Hintergründe oder weitere Objekte im Bereich der Aufnahme wurden vor dem manuell Test entfernt.

### 5.1.1 Testdatensatz Name: Homer

- Punkte Anzahl: 5103
- Erstellungsart: Syntetisch
- Punkteverteilung: Unregelmäßig
- Art der Punktwolken: Geschlossen
- Anzahl der Aufnahmen: 1 (komplett modelliert)

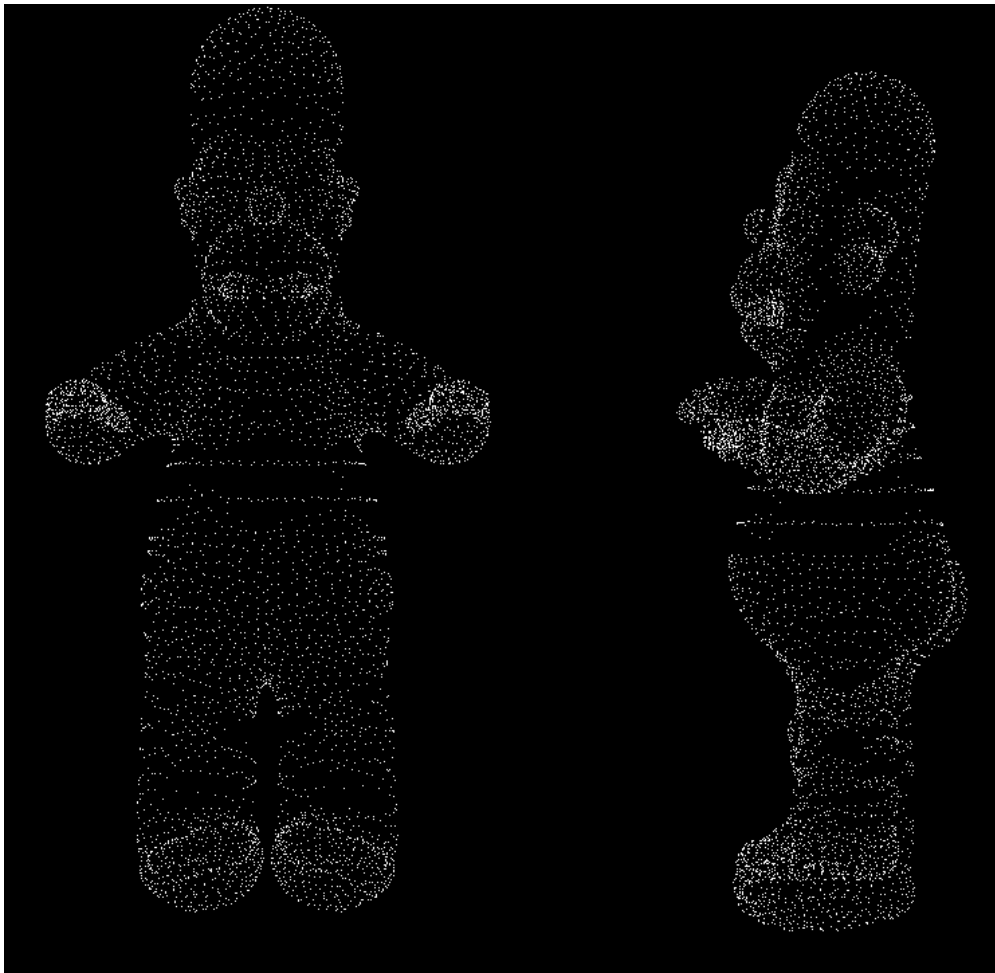
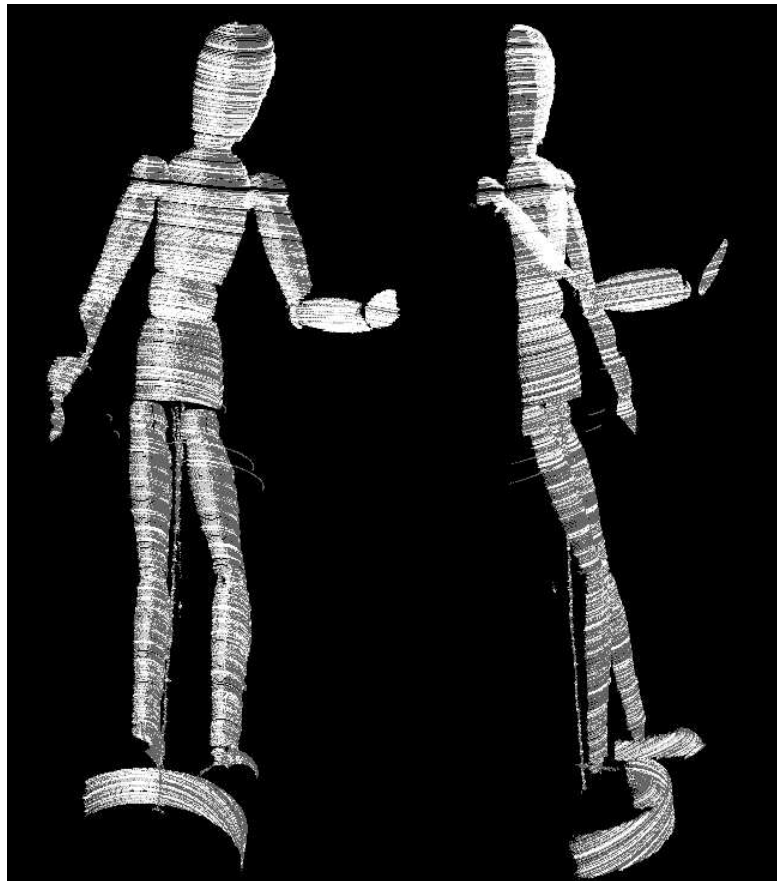


Abbildung 5.1: Der Testdatensatz Homer aus zwei Ansichten

### 5.1.2 Testdatensatz Name: Modelier-Männchen

- Punkte Anzahl: 546140
- Erstellungsart: Tiefenscanner
- Punkteverteilung: Regelmäßig
- Art der Punktwolken: Unvollständig (Rückansicht fehlt)
- Anzahl der Aufnahmen: 20 (verschiedene Ansichten / Objektposen)



**Abbildung 5.2:** Der Testdatensatz: Modeliermännchen aus zwei Ansichten in selber Pose

## 5.2 Ideale Werte für Parameter $\rho_c$ und $\rho_e$

Durch Anwendung der Parameter in verschiedenen Kombinationen soll ermittelt werden, wie diese im Bezug auf die zu Grunde liegende Punktwolke gewählt werden müssen um ein gutes Ergebnis zu erzielen.  $\rho_e$  gibt die Größe der Nachbarschaft an, die zur Normalenschätzung genutzt wird.  $\rho_c$  gibt die Größe der Nachbarschaft an, die für die Ermittlung der Normalenschnitte verwendet wird.

### 5.2.1 Auswirkung von Werteänderungen für $\rho_e$ bei gleich bleibendem $\rho_c$

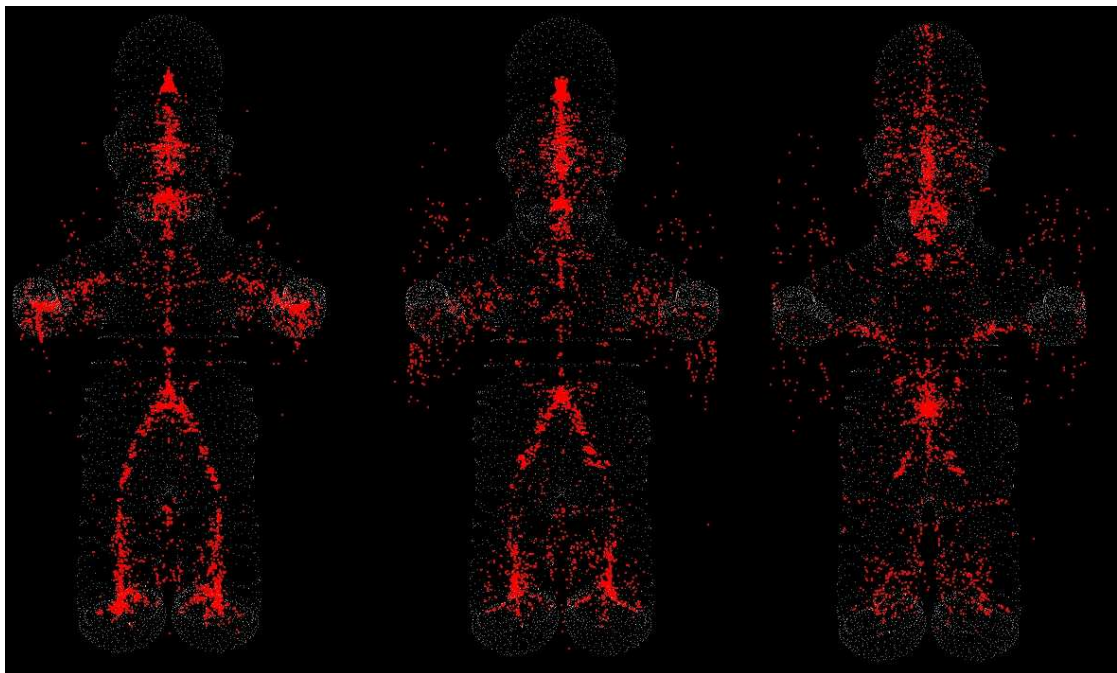
Für  $\rho_c$  wird der Wert 72 angenommen. Der Wert von  $\rho_e$  wird von 9 beginnend in jedem Testschritt verdoppelt. Der Versuch wird zunächst für den Datensatz Homer durchgeführt. Da dieser Datensatz mit 5103 Punkten klein ist und eine ungleichmäßige Punkteverteilung aufweist wird dieser nicht um den Faktor  $\gamma$  reduziert, sondern bleibt in vollem Umfang erhalten.

Die Abbildung 5.3 zeigt für die Veränderung des Wertes  $\rho_e$  von 9 auf 18 keine visuell erkennbare Auswirkung auf die Normalenschnittpunkte. Bei der Wertänderung von 36 auf 72 zeigt sich in Abb. 5.4, dass sich die Regionen hoher Normalenschnittpunktedichte im Bereich des Kopfes ins Zentrum der Figur verschieben und sich zu einer Region verbinden. Dieser Umstand lässt sich dadurch erklären, dass die Oberflächenpunkte der seitlichen Auswölbungen am Kopf der Figur bei einer Nachbarschaftsgröße von 72 weniger Einfluss auf die Normalenschätzung ausüben und daher durch die Einbeziehung der anderen Oberflächenpunkte des Kopfes „geglättet“ werden.

Sehr deutlich wird dieser Umstand für die nächsten Werte 144, 288 oder 576 (Abb 5.5). Die Normalen der Punkte beziehen sich durch das groß gewählte  $\rho_e$  nicht mehr auf lokale Oberflächenstrukturen, sondern bildet sich aus großen Teilen der Objektoberfläche die keine einheitliche Normalenrichtung aufweisen und damit zu einer fehlerhaften Normalenschätzung führen. Das Diagramm 5.6 belegt, dass ab einem bestimmten Verhältnis von Punktwolke zu Nachbarschaftsgröße durch die steigende Anzahl fehlerhafter Normalenschätzungen die Anzahl der Schnittpunkte abnimmt. Für den Datensatz Homer wird dieser maximale Wert im Bereich  $\rho_e = 18$  erreicht, wenn für  $\rho_c$  der Wert 72 angenommen wird.

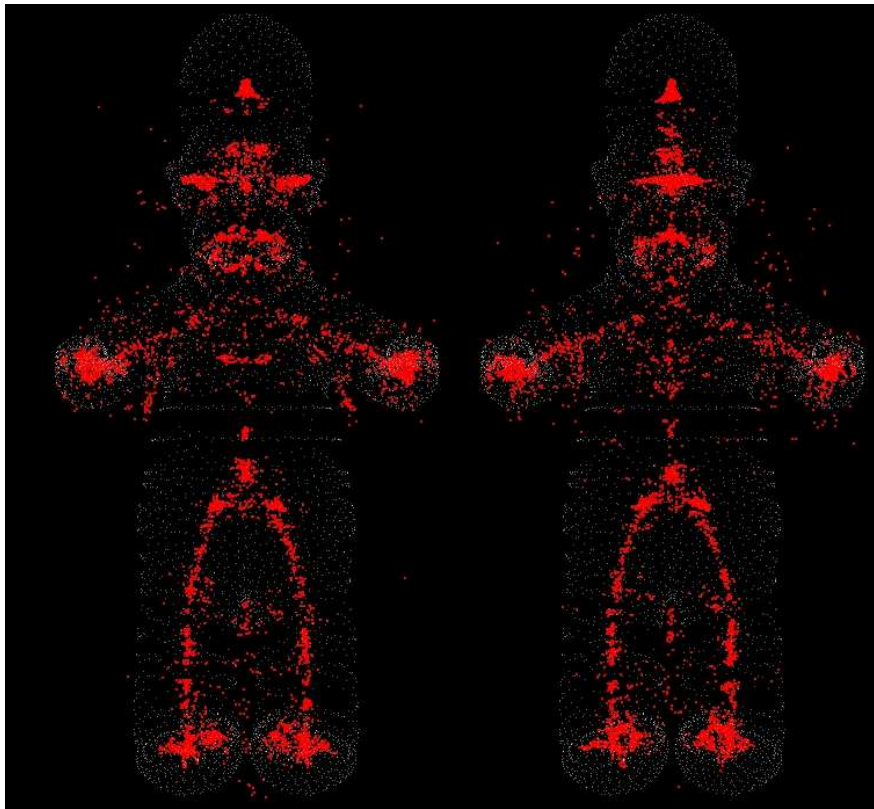


**Abbildung 5.3:** Normalenschnitte (rot) für nicht reduzierte Datensatz Homer mit  $\rho_c = 72$  und  $\rho_e = 9$  (Links) und  $\rho_e = 18$  (Rechts)

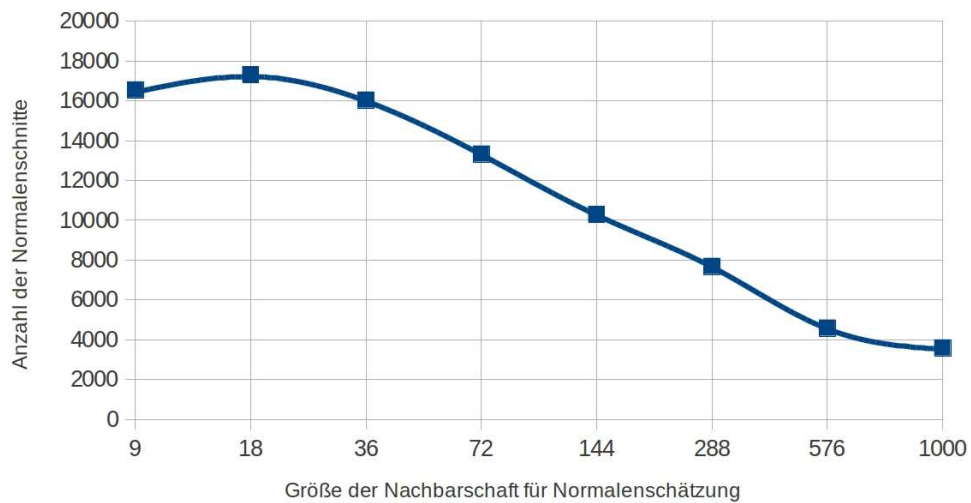


**Abbildung 5.5:** Normalenschnitte (rot) für den nicht reduzierte Datensatz Homer mit  $\rho_c = 72$  und  $\rho_e = 144$  (Links),  $\rho_e = 288$  (Mitte) und  $\rho_e = 576$  (Rechts)





**Abbildung 5.4:** Normalenschnitte (rot) für nicht reduzierte Datensatz Homer mit  $\rho_c = 72$  und  $\rho_e = 36$  (Links) und  $\rho_e = 72$  (Rechts)

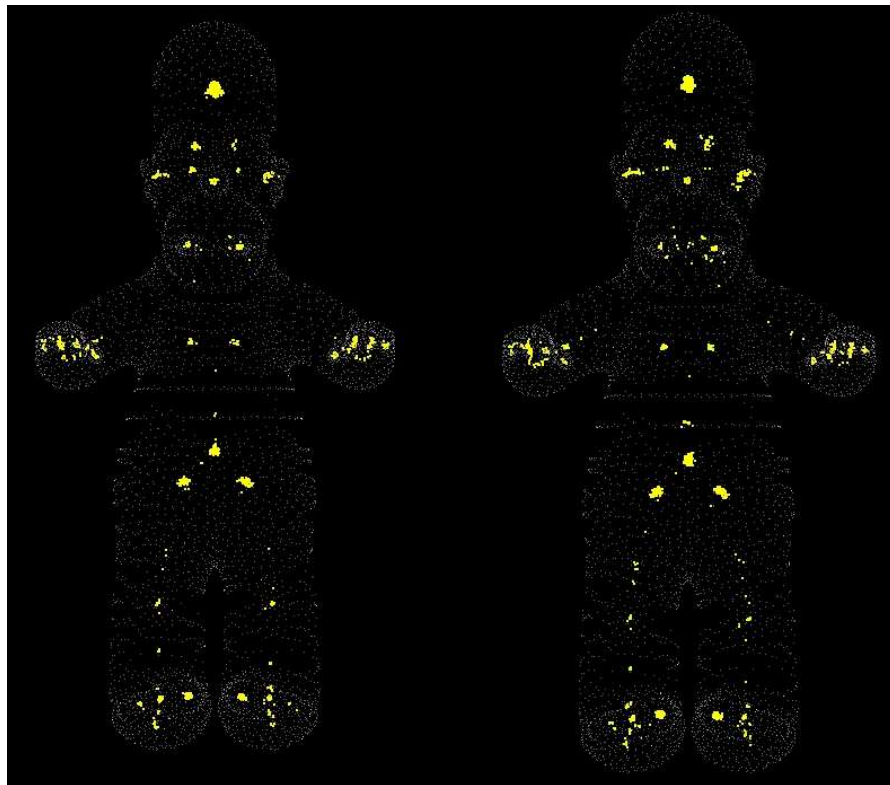


**Abbildung 5.6:** Dieses Diagramm zeigt die sinkende Anzahl der Normalenschnitte bei steigendem  $\rho_e$  für den Datensatz Homer

### 5.2.2 Auswirkung von Werteänderungen für $\rho_c$ bei $\rho_e = 9$

Im diesem Abschnitt soll ermittelt werden wie sich Veränderungen von  $\rho_c$  auf das Ergebnis auswirken. Hierzu wird für  $\rho_e$  der Wert 9 festgelegt. In diesem Test werden ebenso wie im vorangehenden die Nachbarschaftsgrößen 9, 18, 36, 72, 144, 288, 576 und zusätzlich 1000 verwendet. Die Tabelle 5.3 zeigt die Anzahl der Normalenschnitte. Betrachtet man die Anzahl der gefilterten Normalenschnittpunkte fällt auf, dass diese zum Wert 288 hin in der Größenzunahme abnehmen ein Maximum erreichen und danach die Anzahl für  $\rho_c > 288$  wieder abnimmt.

Dies deutet darauf hin, dass sich keine neuen Regionen hoher Schnittpunktdichte durch die zusätzlichen normalen Schnitte ergeben, und das Ergebnis durch ein größer gewähltes  $\rho_c$  nicht verbessert wird. Durch Abbildung 5.9 wird dieses Erkenntnis noch weiter bestärkt, da die Veränderungen der gefilterten Schnittpunktemenge hier marginal sind.



**Abbildung 5.7:** Mit Voxel Gitter gefilterte Normalenschnitte für  $\rho_e = 9$  und  $\rho_c = 72$  (Links) und  $\rho_c = 144$  (Rechts)

| Nachbarschaftsgröße<br>Schnittberechnung $\rho_c$ | Anzahl Normalenschnitte | Anzahl Normalenschnitte<br>(nach Filterung) |
|---------------------------------------------------|-------------------------|---------------------------------------------|
| 9                                                 | 3418                    | 295                                         |
| 18                                                | 6337                    | 641                                         |
| 36                                                | 10656                   | 1079                                        |
| 72                                                | 16554                   | 1421                                        |
| 144                                               | 24203                   | 1482                                        |
| 288                                               | 32941                   | 1503                                        |
| 576                                               | 36004                   | 1374                                        |
| 1000                                              | 36389                   | 1348                                        |

**Tabelle 5.1:** Anzahl der ungefilterten und gefilterten Normalenschnittpunkte im Bezug auf ansteigendes  $\rho_c$  : Homer

### 5.2.3 Berechnungsdauer für unterschiedliche $\rho_e$ und $\rho_c$

Da die Berechnung der Normalen von der Bestimmung der Normalenschnitte entkoppelt ist lassen sich für beide Schritte Zeitmessungen vornehmen. Die gemessenen Zeiten wurden auf einem System mit 6GB Arbeitsspeicher und einem Intel Core I7 (quadcore) 1.Generation ermittelt. Die zur Messung verwendete C++ Boost Bibliothek misst auch bei paralleler Verarbeitung auf Mehrkern-Prozessoren die absolute Ausführungszeit, welche der eines Singlecore System entspricht.

| $\rho_e$ | $\rho_c$ | Berechnung der Normalenschätzung<br>Ausführungszeit |
|----------|----------|-----------------------------------------------------|
| 9        | 72       | 0,04 sek                                            |
| 18       | 72       | 0,06 sek                                            |
| 36       | 72       | 0,11 sek                                            |
| 72       | 72       | 0,25 sek                                            |
| 144      | 72       | 0,72 sek                                            |
| 288      | 72       | 2,36 sek                                            |
| 576      | 72       | 8,4 sek                                             |
| 1000     | 72       | 23,29 sek                                           |

**Tabelle 5.2:** Durch Boost Lib gemessene Ausführungszeit („Single Core“ Intel Core I7, 6GB RAM) für die Normalenschätzung für ansteigendes  $\rho_e$  : Homer

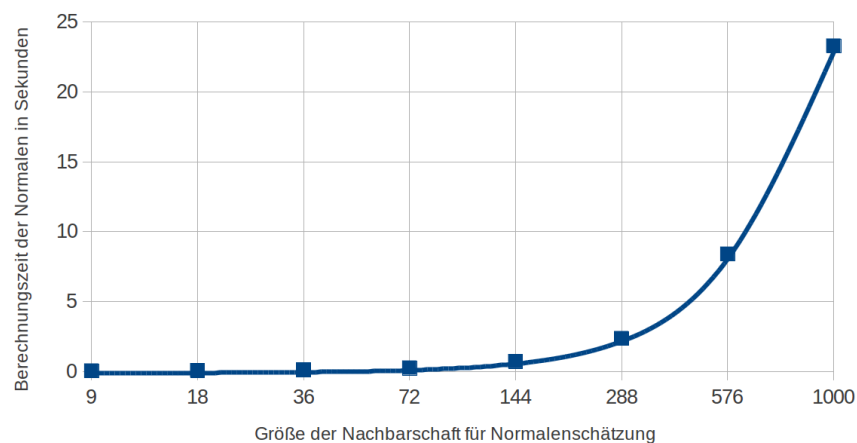
Vergleicht man die Tabellen 5.2 und 5.3, so lässt sich daraus entnehmen, dass  $\rho_c$  eine größere Auswirkung auf die Ausführungszeit hat. Wie schon gezeigt erhält man für ein klein gewähltes  $\rho_e$  bessere Ergebnisse. Wählt man nun  $\rho_e < 144$  liegt

die Ausführungszeit unter einer Sekunde und ist im Vergleich zur Schnittpunktbe-  
rechnung zu vernachlässigen.

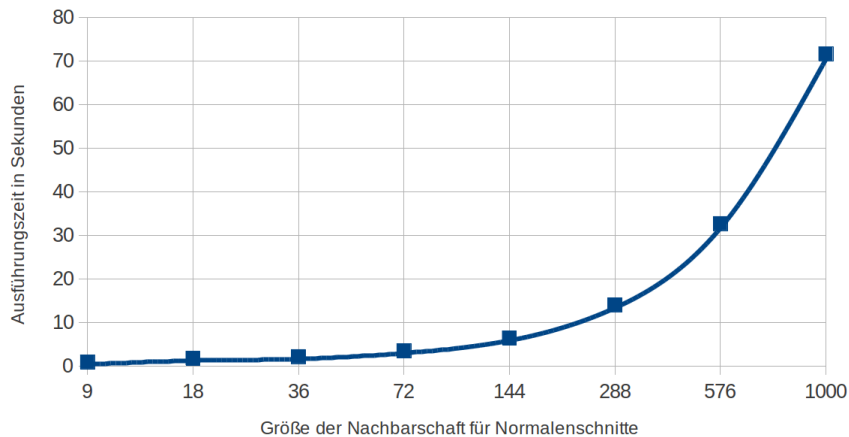
| $\rho_e$ | $\rho_c$ | Berechnung der Normalenschnitte Aus-<br>führungszeit |
|----------|----------|------------------------------------------------------|
| 9        | 9        | 0,9 sek                                              |
| 9        | 18       | 1,74 sek                                             |
| 9        | 36       | 2,13 sek                                             |
| 9        | 72       | 3,49 sek                                             |
| 9        | 144      | 6,42 sek                                             |
| 9        | 288      | 14,02 sek                                            |
| 9        | 576      | 32,67 sek                                            |
| 9        | 1000     | 71,66 sek                                            |

**Tabelle 5.3:** Durch Boost Lib gemessene Ausführungszeit („Single Core“ Intel Core I7, 6GB RAM) zur Bestimmung der Normalenschnitte für ansteigendes  $\rho_c$  : Homer

Überträgt man die Ausführungszeiten beider Tabellen in ein Diagramm so er-  
gibt sich für beide Parameter für einen Werte größer als 144 ein nahezu identischer  
exponentieller Anstieg der Ausführungszeit. Hierbei ist allerdings jeweils auf den  
Wertebereiche zu achten, da dieser für die Normalenschätzung  $[0, 25sek]$  und für  
die Normalenschnittberechnung  $[0, 80sek]$  beträgt.



**Abbildung 5.8:** Diagramm der Ausführungszeiten der Normalenschätzung aus der Ta-  
belle 5.2 mit exponentiellem Anstieg für Werte größer 144



**Abbildung 5.9:** Mit Voxel Gitter gefilterte Normalenschnitte für  $\rho_e = 9$  und  $\rho_c = 72$  (Links) und  $\rho = 144$  (Rechts)

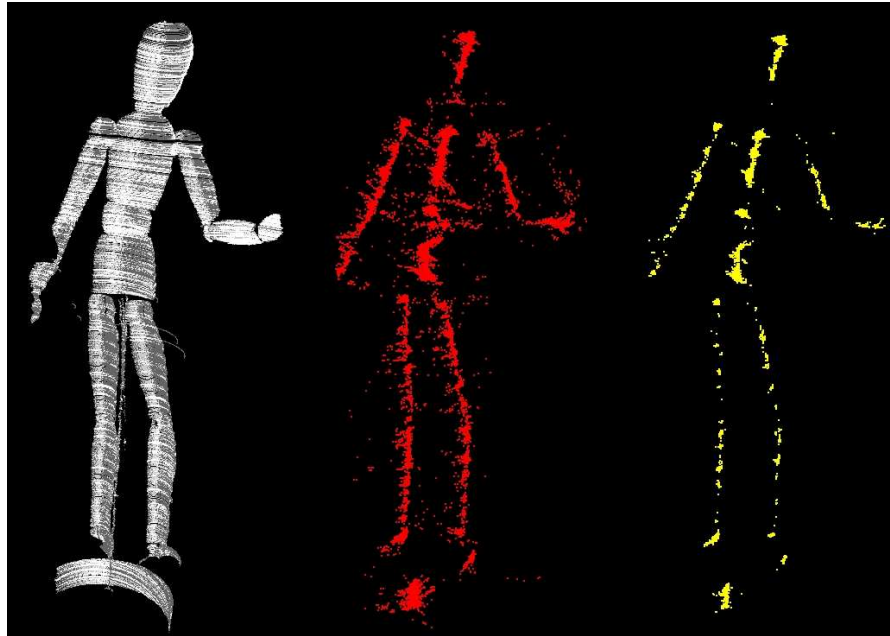
### 5.3 Evaluierung des Verfahrens für unvollständige Datensätze

In diesem Abschnitt soll nun anhand von Tests überprüft werden ob das Verfahren ebenfalls für unvollständige Datensatz, wie zum Beispiel Tiefensensor Daten, verwertbare Ergebnisse erzeugt. Hierzu wird der Datensatz Modelierfigur verwendet, der im Vergleich zum Testdatensatz Homer circa das 100fache an Oberflächenpunkten besitzt, die aber im Gegensatz zum synthetisch erzeugten Objekt gleichmäßig auf der Objektoberfläche verteilt sind. Dadurch lässt sich die durch den Parameter  $\gamma$  steuerbare in Kapitel drei vorgestellte Reduktion der Punktemenge anwenden, ohne dabei zu viele Oberflächendetailinformationen zu verlieren.

Zum direkten Vergleich wählen wir zunächst  $\gamma = 0.01$  um die Anzahl der Punkte auf die des Datensatzes Homer zu bringen.  $\rho_e$  und  $\rho_c$  werden entsprechend der Ergebnisse des vorangegangenen Tests auf die Werte  $\rho_e = 18$  und  $\rho_c = 72$  gesetzt.

Wie die Abbildung 5.10 zeigt wird hier ein deutlich besseres Ergebnis erzielt, was darin begründet ist das sich die Oberflächenpunkte gleichmäßig über die Oberfläche des Objektes verteilen. Dadurch wird sowohl die Normalenschätzung als auch die Normalenschnittberechnung begünstigt, da die durch die Nachbarschaftsgröße

bestimmten Bereiche für verschiedene Punkte auf der Oberfläche die gleichen Ausmaße annehmen und damit die Oberflächenstruktur homogener abbilden.



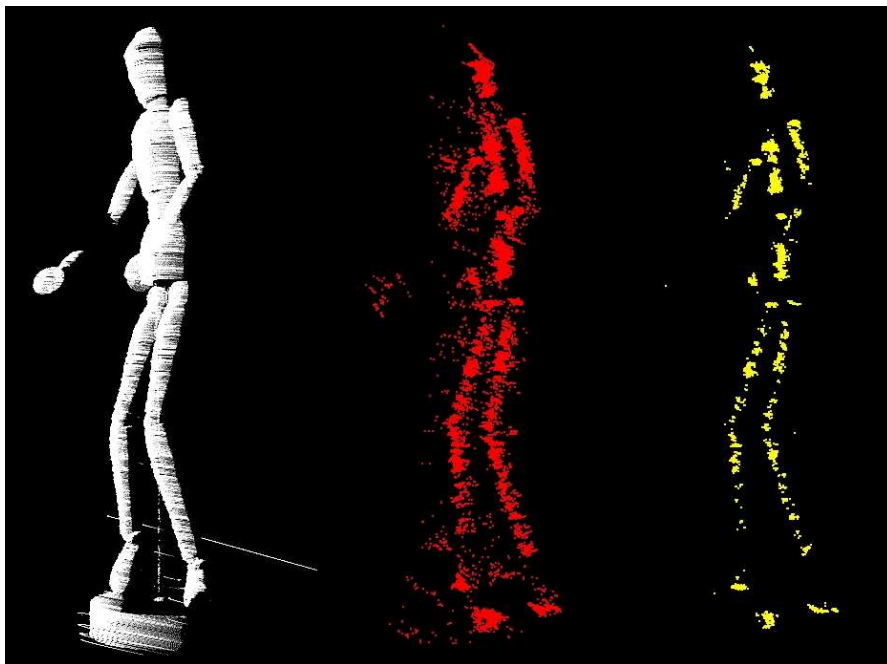
**Abbildung 5.10:** Der Datensatz Modellierfigur mit ungefilterten (Mitte) und gefilterten Normalenschnitten (Rechts) für  $\gamma = 0,01$  (Punkteanzahl nach Reduktion: 5461),  $\rho_e = 18$ ,  $\rho_c = 72$

Im direkten Vergleich der beiden Testdatensätze (Tabelle 5.4) zeigt sich, dass sich für den Testdatensatz Homer bei nahezu gleicher Ausgangspunktemenge mehr als ein Drittel weniger Normalenschnitte ergeben. Die Filterung reduziert diese Schnittpunkt-Anzahl für den Testdatensatz Homer auf 8% und der Modellierfigur auf 10%. Daraus folgt dass es für die Normalenschnitte der Modellierfigur mehr Regionen hoher Schnittpunktdichte gibt. Die höhere Ausführungszeit lässt sich dadurch begründen, dass für die Modellierfigur mehr Normalenschnittpunkte berechnet wurden, da die Bedingungen für die Berechnung häufiger gegeben waren.

Bei dem Vergleich der Ergebnisse sollte beachtet werden, dass es sich bei dem Datensatz der Modellierfigur um einen unvollständigen Datensatz handelt, für den nur die Frontal-Ansicht gegeben ist. Dennoch ist bei einer visuellen Bewertung der ungefilterten als auch der gefilterten Normalenschnitte das Skelett der Modellierfigur deutlich zu erkennen (Abb. 5.11).

| Daten   | $\gamma$ | $\rho_e$ | $\rho_c$ | Punkte-<br>menge | reduzierte<br>Punkte-<br>menge | Normalen-<br>schnitte | Gefilterte<br>Normalen-<br>schnitte | Normalen-<br>schnitte<br>Ausfüh-<br>rungszeit |
|---------|----------|----------|----------|------------------|--------------------------------|-----------------------|-------------------------------------|-----------------------------------------------|
| Homer   | 1        | 18       | 72       | 5103             | 5103                           | 17317                 | 1386                                | 3,45 sek                                      |
| M.figur | 0.01     | 18       | 72       | 546140           | 5461                           | 28767                 | 3088                                | 4,71sek                                       |

**Tabelle 5.4:** Ergebnis-Vergleich der beiden Testdatensätze

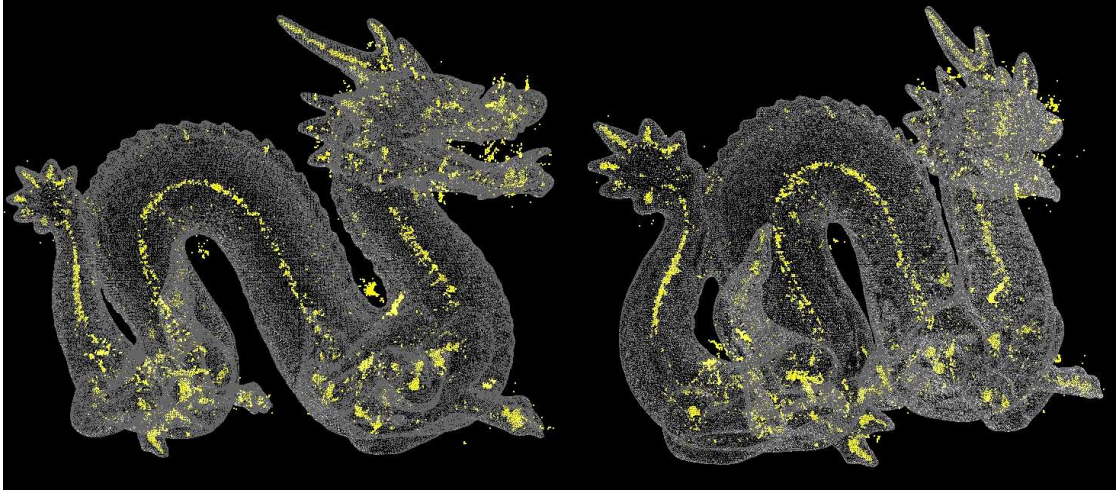


**Abbildung 5.11:** Der Datensatz Modelierfigur mit ungefilterten (Mitte) und gefilterten Normalenschnitten (Rechts) für  $\gamma = 0,01$ ,  $\rho_e = 18$ ,  $\rho_c = 72$

## 5.4 Komplexe Objekte

Auch komplexe Objekte wie der in Abbildung 5.12 gezeigte Drachen funktionieren mit diesem Verfahren. Hierbei gilt, je größer die Punktwolke desto höher die Ausführungszeit. Durch die richtigen Wahl für  $\gamma$  kann diese stark reduziert werden. Die Parameter  $\rho_e$  und  $\rho_c$  sind in Relation zur Größe der Punktwolke und dem Detailgrad der Oberfläche zu wählen. Während der schlangenförmige Körper abgebildet wird sind die Zacken auf dem Rücken des Drachen als Oberflächendetail zu

klein um mit einzufließen. Daraus folgt, dass  $\rho_e$  zu groß gewählt ist und angepasst werden muss.



**Abbildung 5.12:** Modell eines Drachen (437645 Oberflächenpunkte) mit gefilterten Normalenschnitten für  $\gamma = 0,1$  (43764 Punkte nach Reduktion),  $\rho_e = 36, \rho_c = 288$ . Ausführungszeiten: (Normalenschätzung) 0,95 sek, (Normalenschnitte) 121,62 sek

Für die Berechnung der Normalenschnitte werden 121,62 Sekunden benötigt, da der Prozess aber parallel auf mehreren Prozessor-Kernen ablaufen kann reduziert sie sich proportional zur Anzahl der Prozessor-Kerne (zwei Kerne: 60,81 sek, vier Kerne: 30,42 sek).



# Kapitel 6

## Zusammenfassung

Ziel dieser Arbeit ist die Entwicklung eines Verfahrens zur Approximation eines Kurven-Skelettes anhand von 3D Objektoberflächen.

Um in die Thematik der Skeletapproximation einzuführen, wurde zunächst auf Basis der Definition für die Mediale Axen Transformation der Begriff Skelett erklärt und durch die theoretische Beschreibung der Approximationsverfahren, welche auf dieser Definition basieren, veranschaulicht.

Skelett-Eigenschaften wie Translations-, Rotations- und Skalierungsinvarianz sowie die Skelett-Geometrie wurden erläutert um zu zeigen, warum sich die Repräsentation eines Objektes als Skelett für Anwendungszwecke wie zum Beispiel Objekterkennung eignet.

Es wurden die Verfahren Distance-Transform und Thinning in ihrem theoretischen Ablauf, sowie ihrer zu Grunde liegenden formalen Beschreibung vorgestellt. Beide Ansätze basieren auf der Boundary Evolution. Die Erläuterung der Skelett-Approximation auf Basis von Voronoi-Diagrammen sollte einen weiteren Ansatz zeigen, der sich von der Boundary Evolution unterscheidet, hingegen aber auch auf der Definition der Medialen Achsen Transformation basiert. Da die Ergebnis-Skelette dieser Verfahren im  $\mathbb{R}^3$  die Objektoberfläche nicht wie im  $\mathbb{R}^2$  als eindimensionale Kurve beschreiben, sondern Flächen beinhalten, die sich zur weiteren Verarbeitung nicht eignen, wurden im Anschluss die Kurvenskelette eingeführt.

Um eine Vergleichsgrundlage zu haben, wurden in Kapitel zwei die Verfahren „On-the-Fly Curve-skeleton Computation“ und „Curve Skeleton Extraction From Incomplete Point Cloud“ vorgestellt. Das die Definition der Medialen Ach-

sen Transformation nicht für die Approximation von Kurvenskeletten gilt, wurde an diesen Verfahren verdeutlicht. Desweiteren wurde hier die geodätische Funktion vorgestellt, die benötigt wird um mit einem Skelett-Growing-Algorithmus die Objekt-Repräsentation als eindimensionale Kurve zu erzeugen.

In Kapitel drei wurde das in dieser Arbeit entwickelte Verfahren, das durch den Ansatz „Curve Skeleton Extraction From Incomplete Point Cloud“ inspiriert ist, diesen aber anders umgesetzt, vorgestellt. Das Verfahren gliedert sich in zwei Abschnitte, den der Normalenschnittberechnung und des Skelett-Growings-Algorithmus, welche beide theoretisch beschrieben und erläutert wurden.

Die in der Einleitung genannte Hypothese, dass alle Skelettpunkte einer Objektoberfläche auf den Schnittpunkten der Punktnormalen liegen, wird durch die in Kapitel drei verwendeten Grafiken veranschaulicht.

Es wurden die drei Parameter  $\gamma$ ,  $\rho_e$ ,  $\rho_c$  eingeführt und die Berechnungen in denen sie verwendet werden erläutert. Hierbei bestimmt  $\gamma$  als Reduktionsgrad die Anzahl der Oberflächenpunkte.  $\rho_e$  gibt den Wert für die Nachbarschaft an, die der Normalenschätzung zur Ermittlung einer Normalen in einem Oberflächenpunkt zu Grunde gelegt wird.  $\rho_c$  gibt die Größe der Nachbarschaft für die Normalenschnittpunktberechnung an. Desweiteren wurde der formale Ablauf des Skelett-Growing-Algorithmus erläutert, der aus der in Schritt eins des Verfahrens ermittelten Normalenschnittpunktmenge den Skeletgraph erzeugen soll.

Um das Verfahren in seinem Programmaufbau zu verstehen wurden die einzelnen Klassen, sowie deren Funktionen innerhalb des Algorithmus erklärt. Desweiteren wurde gezeigt an welchen Stellen eine parallel Berechnung eingesetzt wird um das Verfahren für Mehrprozessor-Systeme zu beschleunigen.

Die Evaluation des Verfahrens konnte nur für den ersten Schritt erfolgen, da die Implementierung des Skeletgraphen-Growing-Verfahrens zwar anhand der formalen Beschreibung abgeschlossen wurde, aber zu keinem repräsentativen Ergebnis führte. Die Ergebnisse aus dem ersten Verfahrensschritt entsprechen den in der Einleitung geforderten Anforderungen an Güte und Ausführungszeit. Es wurde in der Evaluation gezeigt, dass das Verfahren auch auf unvollständigen Datensätzen die Skelettpunkte zur Erstellung eines Skeletgraphen ermitteln kann ohne dabei einen höheren Rechenaufwand zu erzeugen.

## 6.1 Ausblick

Die wichtigste mögliche Weiterentwicklung ist die Entwicklung oder Anbindung eines Skelett-Growing-Algorithmus, der mit den gefilterten Normalenschnitten aus dem hier vorgestellten Verfahren einen Skelettgraphen erzeugen kann.

Durch die Verwendbarkeit von unvollständigen Oberflächendaten wäre die direkte Verbindung des Verfahrens mit einem Tiefensensor möglich, um eine Verarbeitung von Tiefenbild Serien und damit eine zusätzliche Stabilisierung des Verfahrens zu erreichen oder Skelettanimationen zu berechnen. Aus dieser Animation eines Skelettes könnten Gelenkpunkte ermittelt werden, die sich bei einem statischen Tiefenbild nur durch Schätzung bestimmen lassen.

Durch die hier vorgestellten Parameter kann das Verfahren an verschiedenste Anforderungen angepasst werden und somit für die Objekterkennung mit einer hohen Genauigkeit in der Approximation verwendet werden, aber auch für die Body Pose Estimation mit einer geringeren Genauigkeit eingesetzt werden, um hier kürzere Laufzeiten zu erhalten.



# Literaturverzeichnis

- [BDH96] BARBER, C. B. ; DOBKIN, David P. ; HUHDANPAA, Hannu: The Quick-hull Algorithm for Convex Hulls. In: *ACM Transactions on Mathematical Software* 22 (1996), Nr. 4, S. 469–483
- [BK06] BAUDSON, Christoph ; KLEIN, Edgar: *Berechnung und Visualisierung von Voronoi-Diagrammen in 3D*, Rheinische Friedrich-Wilhelms-Universität Bonn, Diss., 2006
- [Blu67] In: BLUM, H.: *A transformation for extracting new descriptors of shape*. Bd. Models for the Perception of Speech and Visual For. MIT Press, 1967, S. pp. 362–380
- [Fis04] FISHER, John: Visualizing the Connection Among Convex Hull, Voronoi Diagram and Delaunay Triangulation / Michigan Technological University. 2004. – Forschungsbericht
- [Jai89] JAIN, Anil K.: *Fundamentals of Digital Image Processing*. USA : Prentice Hall, 1989
- [KSJ06] K., Tamal ; SUN, Dey ; JIAN: *Defining and Computing Curve-skeletons with Medial Geodesic Function*. 2006
- [Ogn92] OGNIEWICZ, Robert L.: *Discrete Voronoi Skeletons*, Swiss Federal Institute of Technology Zurich, Diss., 1992
- [RCG11] RUSU, R.B. ; COUSINS, S. ; GARAGE, W.: 3D is here: Point Cloud Library (PCL). In: *Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011*
- [Rus09] RUSU, Radu B.: *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*, Computer Science department, Technische Universität München, Germany, Diss., 2009. – Advisor: Univ.-Prof.

Michael Beetz (TUM) Ph.D.; Committee: Univ.-Prof. Dr. Nassir Navab (TUM), Univ.-Prof. Michael Beetz (TUM) Ph.D., Prof. Kurt Konolige (Stanford) Ph.D., Prof. Gary Bradski (Stanford) Ph.D.; summa cum laude.

- [SLSK07] SHARF, Andrei ; LEWINER, Thomas ; SHAMIR, Ariel ; KOBELT, Leif: On-the-fly Curve-skeleton Computation for 3D Shapes. In: *Computer Graphics Forum, (Proceedings Eurographics 2007)* 26 (2007), Nr. 3, S. 323–328
- [SP08] SIDDIQI, Kalem ; PIZER, Stephen M.: *Medial Representations: Mathematics, Algorithms and Applications*. Springer Publishing Company, Incorporated, 2008
- [TZCO09] TAGLIASACCH, Andrea ; ZHANG, Hao ; COHEN-OR, Daniel: *Curve Skeleton Extraction from Incomplete Point Cloud*. 2009
- [YBS03] YOSHIZAWA, Shin ; BELYAEV, Alexander G. ; SEIDEL, Hans-Peter: *Free-form Skeleton-driven Mesh Deformations*. 2003