Fakultät für Informatik und Mathematik

BACHELORARBEIT

3D-Rekonstruktion mit Hilfe des Poisson-Verfahrens

Verfasser:

Englberger Susanne

Lohmaierstr. 19

94405 Landau a. d. Isar

Matrikelnummer: 60318

Prüfer:

Prof. Dr. Thomas Sauer

Lehrstuhl für Mathematik mit Schwerpunkt Digitale Bildverarbeitung

Innstraße 43

94032 Passau

Eidesstattliche Erklärung

Hiermit versichere ich, Susanne Englberger, dass ich diese Bachelorarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Des Weiteren versichere ich, dass alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind, sowie dass diese Arbeit noch keiner anderen Prüfungsbehörde vorliegt.

_____ _____

Ort, Datum *Englberger Susanne*

Inhaltsverzeichnis

1. Einleitung 5
1.1 Motivation 5
1.2 Ziel der Arbeit 5
2. Definitionen und Technologien 6
2.1 Definitionen 6
2.1.1 Indikatorfunktion 6
2.1.2 Gradient 6
2.1.3 Isofläche 6
2.1.4 Octree 6
2.1.5 Trilineare Interpolation 7
2.2 Verwendete Technologie MeshLab 8
2.2.1 Was ist MeshLab 8
2.2.2 Vorgang eines vollständigen Rekonstruktionsvorganges mit MeshLab 9
3. Grundsätzliche Idee 12
4. Rekonstruktionsvorgang 13
4.1 Am einfachen Modell 13
4.1.1 Modellauswahl 13
4.1.2 Funktionsraum definieren 13
4.1.3 Vektorfeld definieren 15
4.1.4 Poisson-Lösung 15
4.1.5 Isofläche entnehmen 16
4.2 Am komplexen Modell 17
4.2.1 Abschätzen der lokalen Punktedichte 18
4.2.2 Berechnung des Vektorfeldes 18
4.2.3 Auswahl eines Isowertes 20
5. Ergebnisse 21
5.1 Erfolgreiche Rekonstruktionen 21
5.2 Fehlerhafte Rekonstruktionen 23
5.3 Vergleich eines Modells mit verschiedenen Einstellungen 25
6. Vergleich mit anderen Methoden 28
6.1 Vergleich mit VRIP 30
6.2 Vergleich mit FFT-basierter Rekonstruktion 31
7. Fazit und Ausblick 32

7.1 Fazit 32
7.2 Ausblick 32
8. Anhang 34
8.1 Anhang: 35
9. Abkürzungsverzeichnis 43
10. Abbildungsverzeichnis 44
11. Quellverzeichnis 46

1. Einleitung

1.1 Motivation

Unter der Leitung von Aless Lasaruk beschäftigte sich ein Team von FORWISS der Universität Passau mit dem Thema der 3D-Rekonstruktion. Ziel war es, aus möglichst wenigen, vom Laien fotografierten Bildern einer Person eine komplette Büste erstellen zu können. Dabei wurde zunächst die Rekonstruktion an einfachen Modellen mit einer höheren Anzahl an Bildern vorgenommen und zunehmend der Schwierigkeitsgrad erhöht, bis die zuvor festgelegten Vorstellungen erreicht wurden. Bei diesem Projekt unterstützte ich das Team als Hilfswissenschaftlerin. Im Zuge dessen war es meine Aufgabe, Teilpunktewolken entstanden durch die Bilder des jeweiligen Menschen aneinander zu fügen und anschließend einen Rekonstruktionsvorgang durchzuführen.

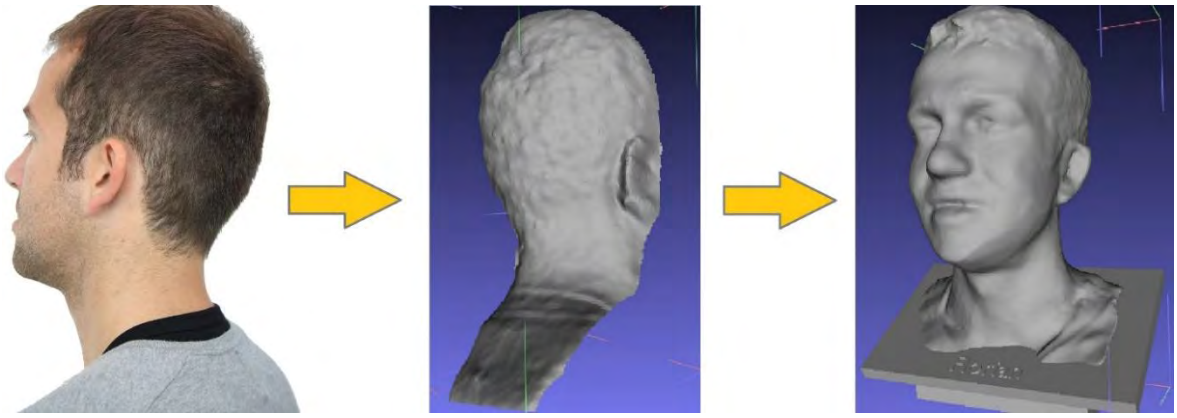


Abbildung 1: Aus Bildern eine Büste erstellen

Hierbei wendete ich bei jedem Vorgang den Filter „Poisson Surface Reconstruction“ an und musste feststellen, dass dieser die unterschiedlichsten Ergebnisse lieferte.

1.2 Ziel der Arbeit

Das Ziel dieser Bachelorarbeit war es daher herauszufinden, was hinter dem Filter „Poisson Surface Reconstruction“ steckt, um mit Hilfe dieses Wissens anschließend gezielter Rekonstruktionen durchführen zu können. Dazu musste zunächst nach bereits bestehender Dokumentation zu diesem Thema recherchiert und diese im Anschluss ihrer Wertigkeit zufolge ausgewertet werden.

2. Definitionen und Technologien

In diesem Kapitel werden zum besseren Verständnis zunächst einige Begriffe definiert, welche im Gesamten Bericht vermehrt auftreten. Anschließend wird ein kurzer Überblick über die verwendete Technologie gegeben.

2.1 Definitionen

2.1.1 Indikatorfunktion

Die Indikatorfunktion $I_{\{R\}}(x): \mathbb{R} \to \{0,1\}$ bezüglich einer Menge $R \subseteq \mathbb{R}$ ist definiert als:

$$I_{\{R\}}(x) = \begin{cases} 1, & x \in R \\ 0, & sonst \end{cases}$$

2.1.2 Gradient

Im Duden wird der Gradient definiert als „Gefälle oder Anstieg einer Größe auf einer bestimmten Strecke“ [1]. Mathematisch hingegen heißt der Vektor

$$\nabla \mathrm{f}(x) = \begin{pmatrix} f_{x_1}(x) \\ \dots \\ f_{x_n}(x) \end{pmatrix}$$

der Gradient von f an der Stelle x. Der Gradient einer Funktion f im Punkt x zeigt in die Richtung des steilsten Anstieges von f. Seine Länge gibt seine Steigung an [2].

2.1.3 Isofläche

Eine Isofläche ist eine Fläche, die im Raum benachbarte Punkte gleicher Merkmale oder Werte miteinander verbindet [3].

2.1.4 Octree

Ein Octree ist eine Datenstruktur der Informatik. Ein Octree ist ein gewurzelter Baum, dessen Knoten jeweils entweder acht direkte Nachfolger oder keine Nachfolger haben [4].

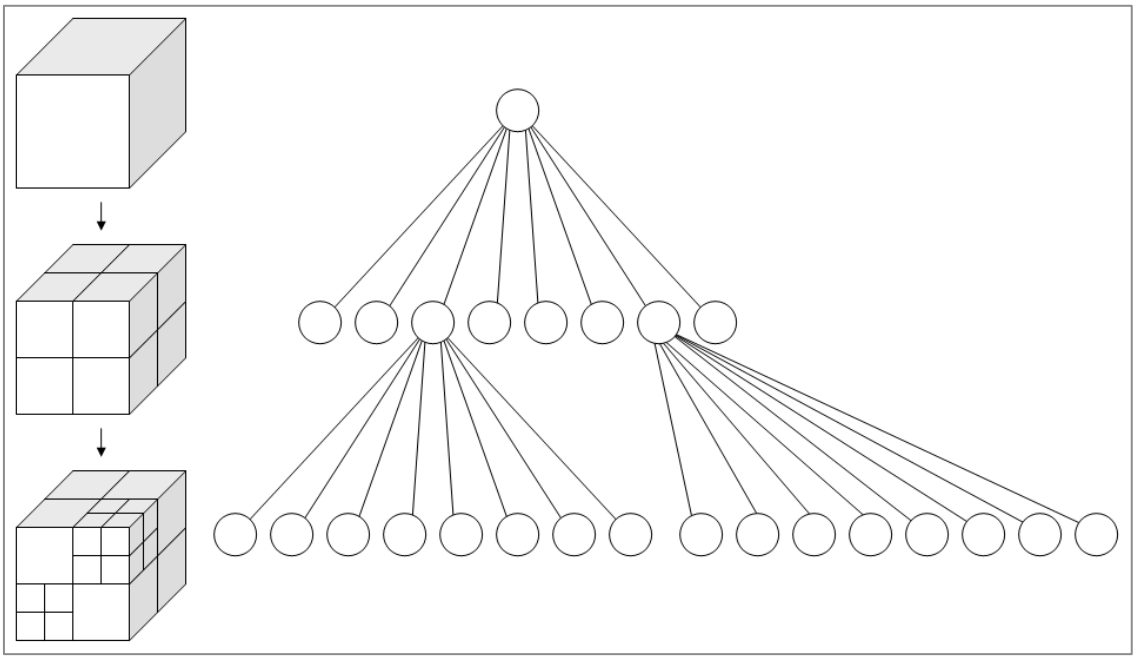


Abbildung 2: Octree-Struktur

2.1.5 Trilineare Interpolation

Trilineare Interpolation ist der Vorgang einer linearen Interpolation auf Eckpunkte innerhalb eines 3D-Gitters. Es wird ein herkömmlicher Einheitswürfel mit der linken unteren Ecke als Basis betrachtet.

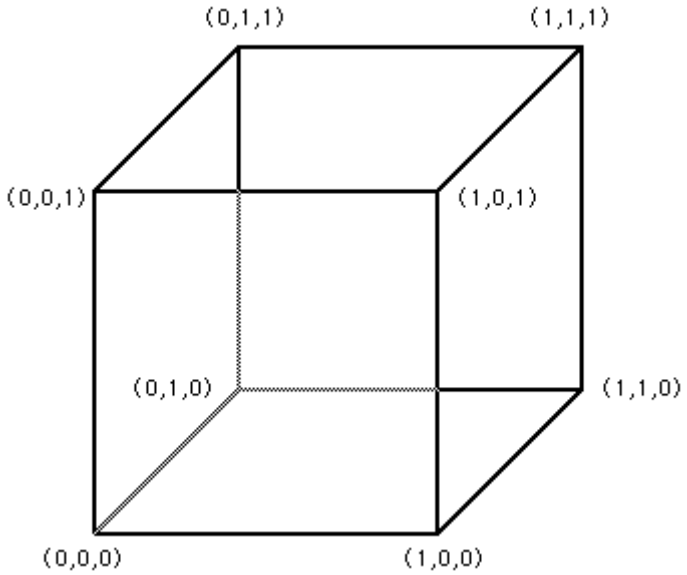


Abbildung 3: Einheitswürfel

Die Eigenschaft jedes Eckpunktes wird bezeichnet mit V_{000}, V_{100}, V_{010} bis hin zu V_{111}. Die Eigenschaft der Position (x,y,z) innerhalb des Würfels wird bezeichnet als V_{xyz} und ist definiert als:

$$\begin{aligned} V_{xyz} = \; & V_{000}(1-x)(1-y)(1-z) + \\ & V_{100}x(1-y)(1-z) + \\ & V_{010}(1-x)y(1-z) + \\ & V_{001}(1-x)(1-y)z + \\ & V_{101}x(1-y)z + \\ & V_{011}(1-x)yz + \\ & V_{110}xy(1-z) + \\ & V_{111}xyz \end{aligned}$$

Generell wird die Box allerdings nicht eine Einheitsgröße besitzen und auch nicht auf den Ursprung ausgerichtet sein. Einfache Umrechnung und Skalierung ermöglicht es dann, zuerst in diesen einfachen Fall umzurechnen und anschließend wieder in den ursprünglichen zurück. [5]

2.2 Verwendete Technologie MeshLab

2.2.1 Was ist MeshLab

MeshLab ist eine frei erhältliche und stetig weiterentwickelte Software für das Erstellen und Bearbeiten von unstrukturierten 3D-Meshes. Das Programm ist dafür geeignet, mit Hilfe von Tools zum Bearbeiten, Bereinigen, Heilen, Kontrollieren, Rendering und Konvertieren von Meshes 3D-Modelle zu erstellen.

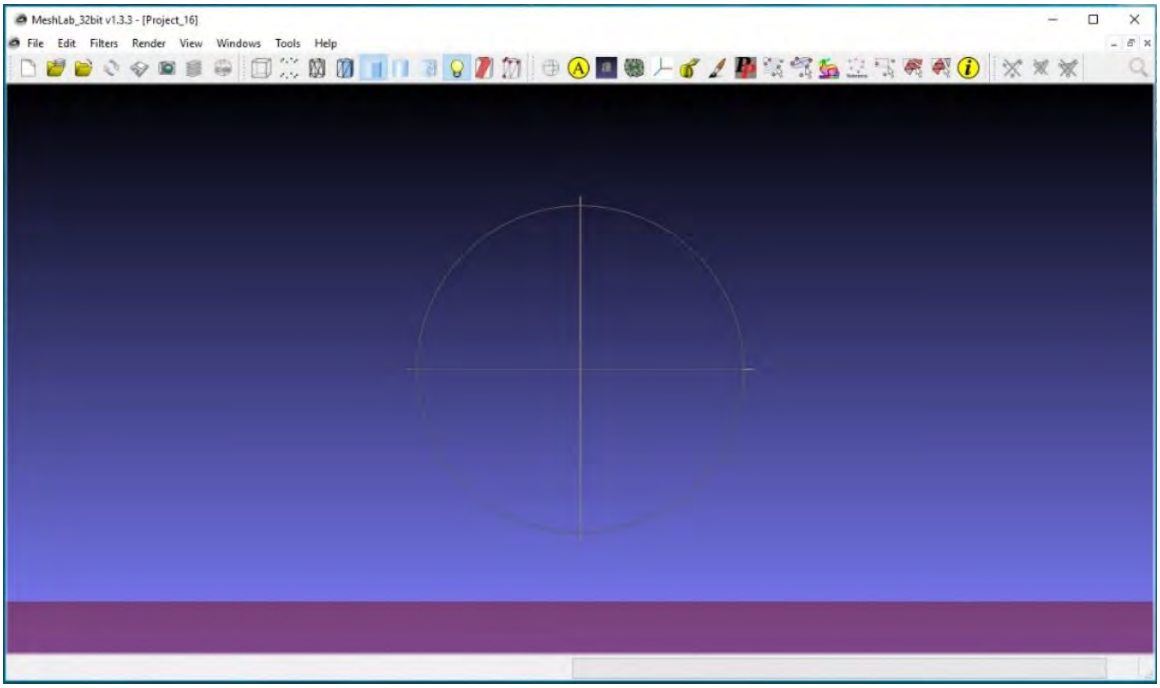


Abbildung 4: Programm Meshlab

Es basiert größtenteils auf der VGG-Bibliothek und ist für Linux, Windows und MacOSX verfügbar. Den Ursprung hat das Projekt Ende 2005, als Studenten die Idee dazu hatten und den Großteil des Codes umsetzten. Folgende Funktionen besitzt Meshlab:

- Auswahl und Entfernung von Teilen des Meshes. Sogar für große Modelle
- Verbindungen zeichen zum Auswählen, Glätten und Einfärben von Meshes
- Eingabe und Ausgabe der verschiedensten Formate wie PLY, STL, OFF, usw.
- 3D-Scanning-Tools um Meshes miteinander zu verbinden und mit dem Poisson surface Reconstruction Algotihmus zu rekonstruieren

[6].

2.2.2 Vorgang eines vollständigen Rekonstruktionsvorganges mit MeshLab

Als erstes müssen von den einzelnen Meshes, welche anhand der Bilder entstanden sind, unerwünschte Wände und Artefakte entfernt werden. Dies geschieht folgendermaßen:

- STL-Datei importieren
- Z-painting anklicken (Pinsel inTool-Leiste)

- Pinsel mit roter Spitze auswählen (dritter Button von links unten)
- Passenden Einstellungen vornehmen (Pinselgröße, Pinselform, usw.)
- Mit Pinsel ungewünschte Stelle markieren
- Entf-Taste drücken
- Abspeichern

Anschließend werden die einzeln zugeschnitten Meshes nacheinander eingefügt. Da die einzelnen Meshes nun wahllos orientiert in der Oberfläche liegen, müssen sie punktuell zusammengeführt werden. Dies geschieht folgendermaßen:

- Import Mesh: STL-Dateien einzeln importieren
- Align anklicken (gelbes A in Tool-Leiste)
- Erste STL-Datei auswählen, die fixiert werden soll.
  Anklicken – Glue Here Mesh
  Mesh wird mit einem Stern * gekennzeichnet
- Zweite STL-Datei auswählen, die an die Fixierte angebunden werden soll.
  Anklicken – Point Based Glueing
- Beide Meshes passend drehen und jeweils links und rechts mindestens vier gleiche Punkte wählen, an denen die beiden verbunden werden – OK
- Nächste STL-Datei auswählen und gleich verfahren
- Zum Schluss Process klicken

Die einzelnen Meshes sind jedoch noch immer eigenständige Dateien. Aus diesem Grund empfiehlt sich hier, diese zu einer einzelnen STL-Datei zusammen zu fügen. Sollte nämlich der eigentliche Rekonstruktionsvorgang fehlschlagen, muss wieder von vorne begonnen werden. Hat man die einzelnen Dateien allerdings vereint, so muss lediglich der Rekonstruktionsvorgang wiederholt werden. Das Zusammenfügen zu einer Datei funktioniert so:

- ► Import Mesh: STL-Dateien einzeln importieren
- ► Filters – Mesh Layer – Flatten Visible Layers – Apply – Close
- ► File – Export Mesh As… – File of Type: STL File Format (*.stl) – Save

Abschließend kann der eigentliche Rekonstruktionsvorgang durchgeführt werden:

- STL-Datei, die gefüllt werden soll, importieren
- ► Filters – Point Set – Surface Reconstruction: Poisson

- Apply – Close

Bei der Poisson Surface Reconstruction müssen einige Werte ausgewählt werden. Diese erklären sich nach der folgenden Beschreibung des Rekonstruktionsvorganges.

3. Grundsätzliche Idee

Die grundsätzliche Idee des Poisson-Verfahrens klingt zunächst recht einfach. Es muss lediglich eine 3D-Indikatorfunktion berechnet werden, welche definiert ist mit 1 für Punkte innerhalb des Modells und 0 außerhalb. Anschließend entnimmt man die passende Isofläche und erhält die rekonstruierte Oberfläche.

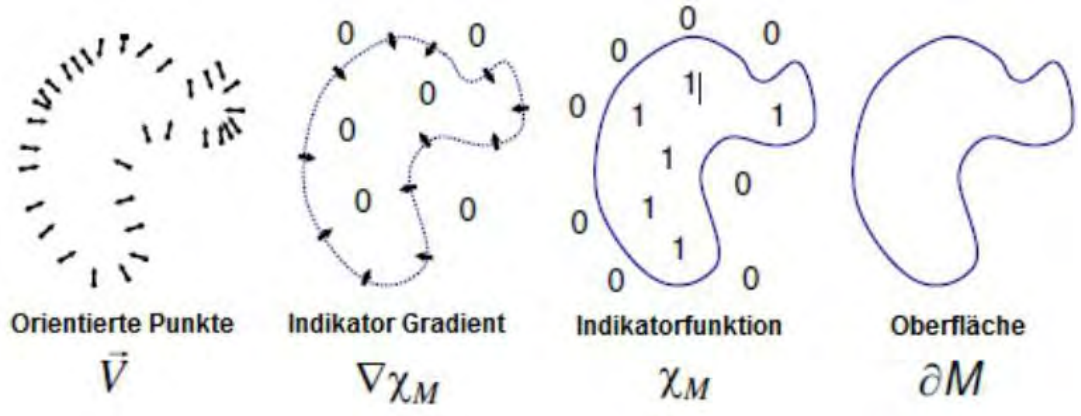


Abbildung 5: Grundsätzliche Idee des Poisson-Verfahrens

Ausgangspunkt ist eine orientierte Punktewolke $\vec{V}$ mit nach innen zeigenden Normalen an den Punkten, welche die Fläche des Objektes begrenzt (*Bild 1*). Mit Hilfe des Normalenfeldes dieser Punktewolke wird nun der Gradient der Indikatorfunktion $\nabla\chi_M$ berechnet (*Bild 2*). Durch den berechneten Gradienten kann anschließend die Indikatorfunktion χ_M abgeleitet werden. Diese besitzt überall innerhalb der Fläche den Wert 1 und überall sonst den Wert 0 (*Bild 3*). Zum Schluss wird die Oberfläche realisiert mittels eines Marching-Cube ähnlichem Algorithmus, der mit einer Octree-Struktur arbeitet (*Bild 3*). Das heißt die Punktewolke wird in Voxelgitter zerlegt und entschieden, welche Punkte die Isofläche des Objektes definieren. Die Schnittpunkte der Isofläche mit den Kanten des Voxel generieren eine Triangulierung, wodurch das 3D-Gittermodell entsteht [7].

4. Rekonstruktionsvorgang

Zunächst wird der Rekonstruktionsvorgang am vereinfachten Modell betrachtet. Das heißt, es wird angenommen, dass sich die Punktproben konstant und gleichmäßig über die Modelloberfläche verteilen. Daraus wird ein Funktionsraum mit hoher Auflösung nahe und grober Auflösung entfernt davon definiert und dann das Vektorfeld $\vec{V}$ als lineare Summe dieser Funktionen formuliert. Im Anschluss wird die Poisson-Gleichung aufgestellt und gelöst und darauf folgend die Isofläche aus der resultierenden Indikatorfunktion entnommen. Abschließend wird der Algorithmus für nicht gleichmäßig verteilte Punktproben erweitert [7].

4.1 Am einfachen Modell

4.1.1 Modellauswahl

Eine genaue Repräsentation der impliziten Funktion ist nur nahe der rekonstruierten Oberfläche möglich. Dies regte die Verwendung eines anpassungsfähigen Octrees an, da damit erstens die implizite Funktion repräsentiert und zweiten das Poisson – System gelöst werden kann. Die Positionen der Punktproben werden benutzt, um Octree θ zu definieren und anschließend wird eine Funktion F_0 mit jedem Knoten $o \in \theta$ des Baums in Verbindung gebracht. Nun werden der Baum und die Funktion so gewählt, dass folgende drei Punkte gelten:

1. Vektorfeld $\vec{V}$ kann als lineare Summe von F_0 repräsentiert werden
2. Poisson – Gleichung ausgedrückt durch F_0 kann gelöst werden
3. Repräsentation der Indikatorfunkton und der Summe von F_0 kann nahe der Oberfläche bewertet werden.

[7]

4.1.2 Funktionsraum definieren

Gegeben sind eine Sammlung von Punktproben S, eine maximale Baumtiefe D und Knoten $o \in \theta$.

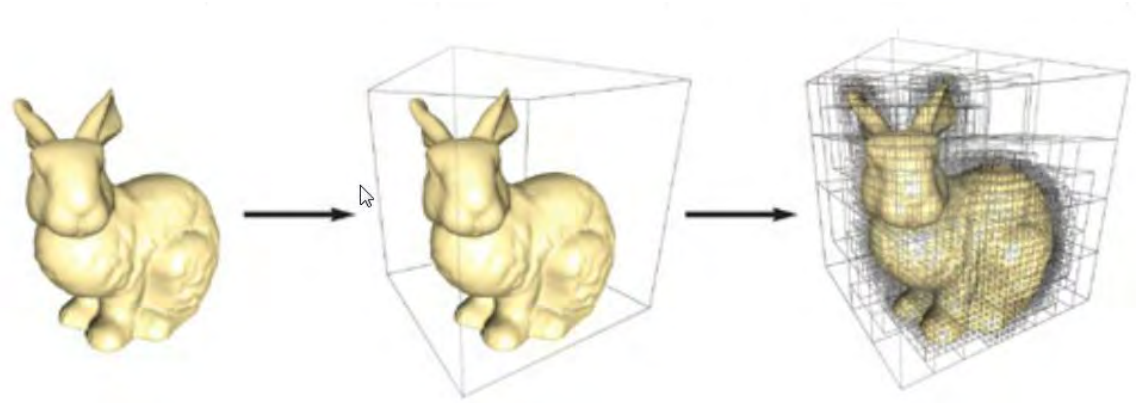


Abbildung 6: Octree θ über Punktprobe S

Dadurch wird als erstes ein Octree θ mit der Eigenschaft definiert, dass jede Punktprobe in einen Knoten der Tiefe D fällt und im Anschluss ein Funktionsraum $\mathcal{F}_{\theta,F}$ für den gilt:

Für jeden Knoten $o \in \theta$ wird F_0 als Einheitsintegral gesetzt, zentriert zum Knoten o und ausgedehnt durch die Größe von o:

$$F_o(q) = F\left(\frac{q - o.c}{o.w}\right)\frac{1}{o.w^3}$$

mit:

$$o.c : Knotenzentrum$$

$$o.w : Knotengröße$$

$$F_P(q) = F(q - p) : Glättungsfaktor\ umgerechnet\ auf\ Punkt\ P$$

$$\underline{\Rightarrow\ Funktionsraum\ \mathcal{F}_{\theta,F} = Span\{F_o\}}$$

Dieser Funktionsraum hat eine detaillierte Struktur ähnlicher zu der einer Wavelet Darstellung. Feinere Knoten werden verknüpft mit höheren Frequenzfunktionen und die Funktionsdarstellung wird bei Annäherung an die Oberfläche immer präziser.

Jetzt kann eine Basisfunktion ausgewählt werden. Hier ist das Ziel, eine Funktion F so zu wählen, dass das Vektorfeld $\vec{V}$ präzise und effizient als die lineare Summe der Knotenfunktionen $\{F_o\}$ repräsentiert werden kann. Darum wird die Position jeder Probe mit dem Zentrum des Blattknotens ersetzt und man erhält:

$$F(q) = \widetilde{F}\left(\frac{q}{2^D}\right)$$

[7]

4.1.3 Vektorfeld definieren

Für die Genauigkeit von Unterknoten wird nicht das Zentrum des Astknotens verwendet, sondern eine trilineare Interpolation, um die Proben auf die acht am nächsten liegenden Knoten zu verteilen. Darum wird die Approximation des Gradientenfeldes der Indikatorfunktion folgendermaßen definiert:

$$\vec{V}(q) = \sum_{s \epsilon S} \sum_{o \epsilon Ngbr_D(s)} \alpha_{O,S} F_O(q) s.\vec{N}$$

mit:

$$Ngbr_D(s) = 8\ am\ nahesten\ liegenden\ D - Tiefe - Knoten\ zu\ s.p$$

$$\{\alpha_{O,S}\} = trilineare\ Interpolationsgröße$$

Da die Proben gleichmäßig verteilt sind, wird angenommen, dass der Bereich einer Stelle P_S konstant und $\vec{V}$ eine gute Approximation ist [7].

4.1.4 Poisson-Lösung

Nachdem das Vektorfeld nun definiert ist, soll die Funktion $\tilde{\chi} \epsilon \mathcal{F}_{\theta,F}$ gelöst werden, so dass der Gradient von $\tilde{\chi}$ nahe an $\vec{V}$ ist. Allerdings bedeutet die Tatsache, dass $\tilde{\chi}$ und $\vec{V}$ im Raum $\mathcal{F}_{\theta,F}$ sind nicht unbedingt, dass auch $\nabla\tilde{\chi}$ und $\nabla\vec{V}$ Raum $\mathcal{F}_{\theta,F}$ sind. Für die Lösung dieses Problems muss also die Projektion von $\nabla\tilde{\chi}$ auf Raum $\mathcal{F}_{\theta,F}$ gleich der Projektion von $\nabla\vec{V}$ sein. Da die Funktionen F_o generell keine Orthonormalbasis formen, ist das Lösen dieses Problems schwierig, jedoch ist eine Vereinfachung möglich:

$$\sum_{o\ \epsilon\ \theta} \left\| \langle \nabla\tilde{\chi} - \nabla\vec{V}\ ;\ F_O \rangle \right\|^2 = \sum_{o\ \epsilon\ \theta} \left\| \langle \nabla\tilde{\chi}\ ;\ F_O \rangle - \langle \nabla\vec{V}\ ;\ F_O \rangle \right\|^2$$

Die Gegebenheit des $\|\theta\|$ – dimensionalen Vektors v, dessen o-the Koordinate $v_o = \langle \nabla \vec{V} \, ; \, F_O \rangle$ ist, stellt nun das Ziel dar, die Funktion $\tilde{\chi}$ so zu lösen, dass der Vektor, welcher durch das Projizieren des Laplace-Operators von $\tilde{\chi}$ auf jedes F_O entstanden ist, zu v so gleich wie möglich ist. Um dies in Matrixform auszudrücken ist nun $\tilde{\chi} = \sum_o \chi_o F_o$. Anschließend wird die Matrix L so definiert, dass L_x das Punkt-Produkt des Laplace-Operators mit jeder F_o zurück gibt. Das heißt, es gilt für alle $o, o' \in \theta$:

$$L_{o,o'} = \langle \frac{\partial^2 F_o}{\partial x^2} ; F_{o'} \rangle + \langle \frac{\partial^2 F_o}{\partial y^2} ; F_{o'} \rangle + \langle \frac{\partial^2 F_o}{\partial z^2} ; F_{o'} \rangle$$

Also muss $\min_{x \in \mathbb{R}^{|\theta|}} \|L_x - v\|^2$ gefunden werden, damit es eine Lösung für $\tilde{\chi}$ gibt. Es wird eine Multigitterähnliche Annäherung verwendet, um die Beschränkung $L_d$von L im Funktionsraum aufgespannt durch die Tiefe d zu lösen und gibt diese Lösung zurück auf $\mathcal{F}_{\theta,F}$ [7].

4.1.5 Isofläche entnehmen

Damit eine rekonstruierte Oberfläche $\partial \widetilde{M}$ erhalten wird, ist es zunächst einmal nötig, einen Isowert zu wählen und im Anschluss die dazugehörige Isofläche aus der berechneten Indikatorfunktion zu entnehmen. Die Wahl des Isowertes wird so getroffen, dass die entnommene Oberfläche beinahe der Position der Eingabepunkte entspricht. Dies wird durch das Abschätzen von $\tilde{\chi}$ auf die Punktepositionen und das Benutzen des Durchschnittes der Werte der Isoflächen-Entnahme erreicht:

$$\partial \widetilde{M} = \{ q \in \mathbb{R}^3 \,|\, \tilde{\chi}(q) = \gamma \}$$

mit:

$$\gamma = \frac{1}{|s|} \sum_{s \in S} \tilde{\chi}(s.p)$$

Die Wahl dieses Isowertes hat den Vorteil, dass das Hinaufsetzen von $\tilde{\chi}$ die Isofläche nicht verändert. Das heißt die Kenntnis des Vektorfeldes $\vec{V}$ bis hin zu einer multiplikativen Konstante liefert ausreichend Informationen für die Rekonstruktion der Oberfläche. Um die Isofläche von der Indikatorfunktion zu entnehmen, wird eine Methode ähnlich dem Marching – Cube – Algorithmus verwendet. In diesem Fall jedoch gibt es eine kleine Abänderung. Anstatt den Positionen von Null-Übergängen entlang einer Kante werden Null-Übergänge berechnet durch die feinsten Ebenen-Knoten angrenzend zu den Kanten verwendet. Für den Fall, dass die Kante eines Knotens mehrere Null-Übergänge besitzt, wird der Knoten unterteilt. Anschließend wird die Triangulierung der Fläche berechnet [7].

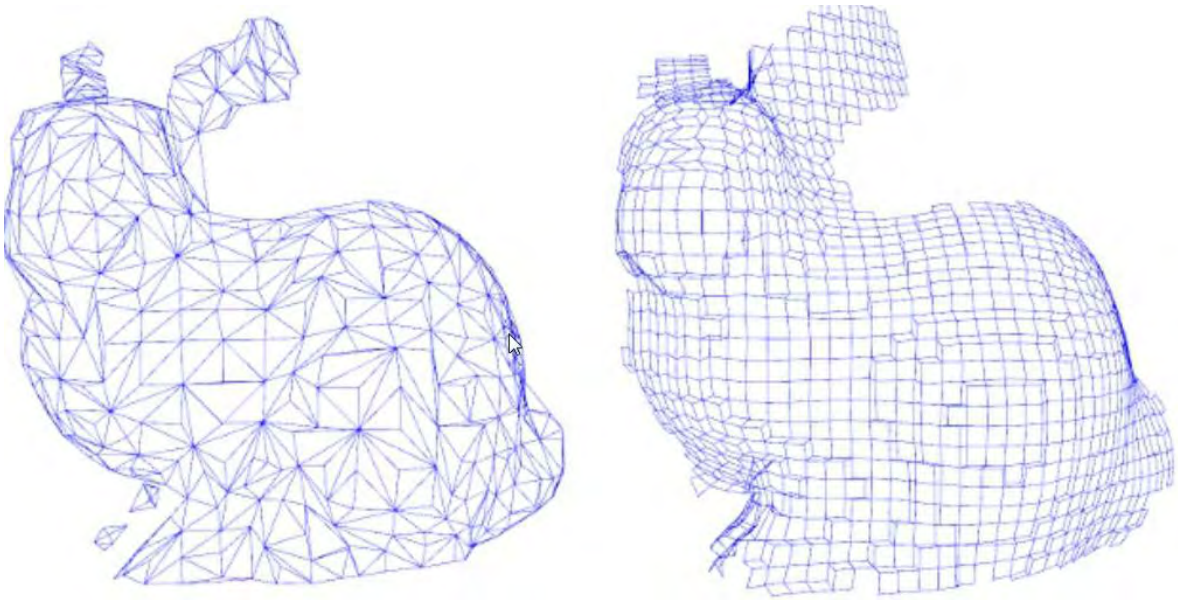


Abbildung 7: Triangulierung aus Marching-Cube-ähnlichem Algorithmus

4.2 Am komplexen Modell

Die am einfachen Modell angewandte Methode wird nun für nicht gleichmäßig verteilte Punktproben erweitert. Das Ziel ist, die Punktedichte abzuschätzen und den Einfluss jedes einzelnen Punktes dementsprechend zu skalieren. Um jedoch einfach das Ausmaß eines Kernes mit einer festen Größe verbunden mit jedem Punkt skalieren zu können, muss zusätzlich die Kerngröße angepasst werden. Das Ergebnis daraus ist eine Rekonstruktion, welche zu Eigenschaften in Gebieten von Dichteproben führt und eine geglättete Form in dünnen Regionen liefert [7].

4.2.1 Abschätzen der lokalen Punktedichte

Die Dichteberechnung wird mit Hilfe eines Kerndichtekalkulators implementiert. Das Ziel hierbei ist die Anzahl der Punkte in der umliegenden Region eines Punktes abzuschätzen. Dabei werden die Proben in ein 3D-Gitter gesplittet, indem die Split-Funktion mit einem Glättungsfaktor gefaltet und die Faltung jedes einzelnen Probenpunktes berechnet werden.

Die Faltung wird folgendermaßen berechnet. Gegeben ist eine Tiefe $\widehat{D} \leq D$, wodurch die Dichteberechnung als die Summe von Knotenfunktionen der Tiefe $\widehat{D}$ gesetzt werden kann:

$$W_{\widehat{D}}(q) \equiv \sum_{s \in S} \sum_{o \in Ngbr_{\widehat{D}}(s)} \alpha_{o,s} F_o(q)$$

Baumknoten mit geringerer Auflösung werden mit Funktionen, welche sich Gaussians mit größere Weiter annähern, assoziiert. Deshalb liefert der Parameter $\widehat{D}$ mit kleineren Werten Probendichteberechnung über größere Regionen [7].

4.2.2 Berechnung des Vektorfeldes

Mit Hilfe der Dichteberechnung wird nun die Aufsummierung aus 4.1.3 abgeändert, sodass jeder Probenbeitrag proportional zu seiner angeschlossenen Umgebung der Oberfläche ist. Wird die Tatsache benutzt, dass die Umgebung umgekehrt proportional zu der Probendichte ist, so wird folgendes gesetzt:

$$\vec{V}(q) \equiv \sum_{s \epsilon S} \frac{1}{W_{\widehat{D}}(s.p)} \sum_{o \epsilon Ngbr_D(s)} \alpha_{O,S} F_O(q)$$

Wird allerdings nur die Größe des Probenbeitrages angepasst, ergibt sich eine schwache Rauschfilterung in schwach besetzten Punktregionen (siehe Abbildung 8).

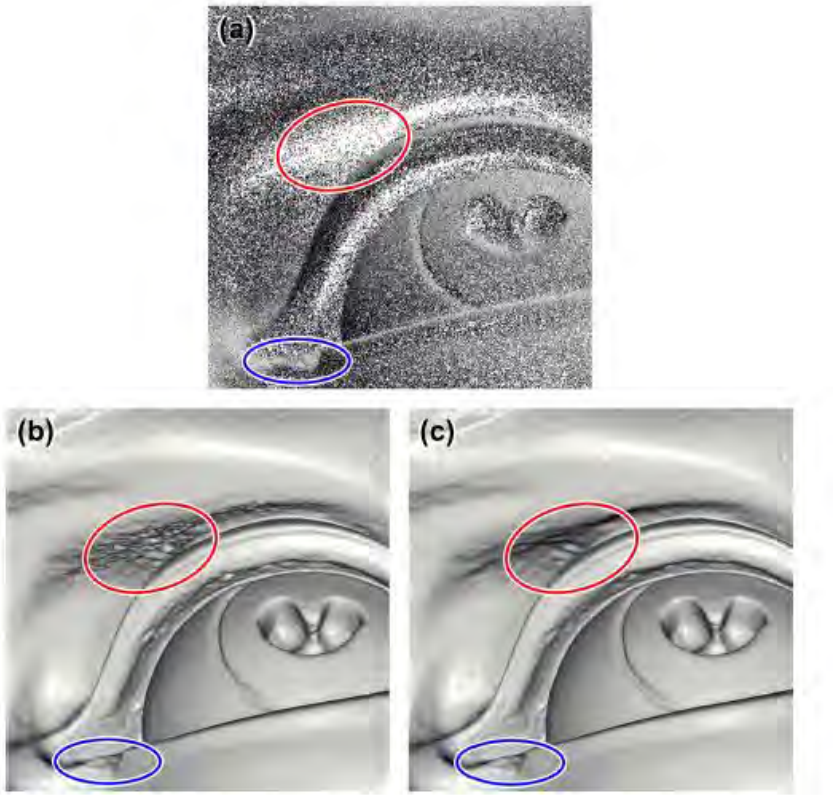


Abbildung 8: Punktprobe (a) ; FFT-Annäherung (b) ; Poisson-Rekonstruktion (c)

Deshalb wird zusätzlich die Weite des Glättungsfaktors $\hat{F}$ zur Probendichte angepasst. Diese Anpassung der Filtergröße bewahrt die feinen Details in dichten Regionen, während Rauschen in dünn besiedelten Gebieten ausgeglättet wird. Wird die Tatsache verwendet, dass Knotenfunktionen mit geringerer Tiefe mit größerem Glättungsfaktor übereinstimmen, so ergibt sich:

$$\vec{V}(q) \equiv \sum_{s \epsilon S} \frac{1}{W_{\hat{D}}(s.p)} \sum_{o \epsilon Ngbr_{Depth(s.p)}(s)} \alpha_{O,S} F_O(q)$$

Hier repräsentiert *Depth(s.p)* die gewünschte Tiefe eines Probenpunktes $s \in S$ und ist definiert durch den Durchschnitt der Punktedichte W über all die Proben:

$$Depth(s.p) \equiv \min(D, D + log_4\left(\frac{W_{\hat{D}}(s.p)}{W}\right))$$

Somit ist die Weite des Glättungsfilters, mit dem s zu $\vec{V}$ beiträgt, proportional zum Radius des angebundenen Oberflächenfleckes P_s [7].

4.2.3 Auswahl eines Isowertes

Abschließend wird nun noch der Punkt der Oberflächenentnahme angepasst, indem ein Isowert gewählt wird, welcher der gewichtete Durchschnitt der Werte von $\tilde{\chi}$ der Punktepositionen ist:

$$\partial\tilde{M} \equiv \{q \in \mathbb{R}^3 \mid \tilde{\chi}(q) = \gamma\}$$

mit:

$$\gamma = \frac{\sum \frac{1}{W_{\hat{D}}(s.p)} \tilde{\chi}(s.p)}{\sum \frac{1}{W_{\hat{D}}(s.p)}}$$

[7]

5. Ergebnisse

5.1 Erfolgreiche Rekonstruktionen

Während des Projektes wurden eine Vielzahl von Rekonstruktionsvorgängen vorgenommen. Dabei haben sich durchaus einige Modelle sehr erfolgreich rekonstruieren lassen. Um das Programm und die Arbeitsweise des Poisson Surface Reconstruktion Algorithmus kennen zu lernen, wurde zu Beginn ein Standard-Kopf verwendet. Dieser besitzt keine Haare, Augenbrauen und sonstige Unregelmäßigkeiten und wurde zusätzlich punktuell eingemessen, wodurch exakte Meshes entstanden sind.

Erster Vorgang:

- Mesh Ansicht frontal vorne
- Mesh Ansicht seitlich links
- Mesh Ansicht seitlich rechts
- Mesh Ansicht frontal hinten

Wie sich in der Abbildung 9, Bild links erkennen lässt, entstand auf diese Weise eine Punktewolke, welche lediglich ein Loch an der Oberseite des Kopfes und an der Öffnung des Halses besaß. Diese wurde mit dem Filter „Poisson Surface Reconstruction“ und einer „Octree Depth“ von 8 aufgefüllt. Das Bild links zeigt, dass ein nahezu perfektes 3D-Modell entstanden ist und die beiden Löcher problemlos gefüllt werden konnten.

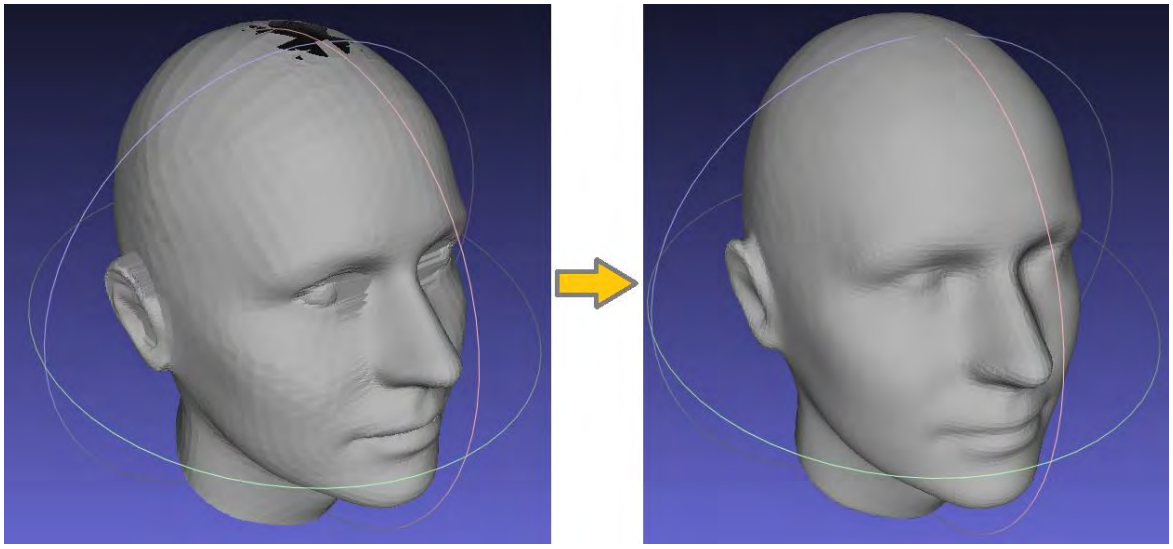


Abbildung 9: Erfolgreiche Rekonstruktion

Zweiter Vorgang:

- Mesh Ansicht frontal vorne
- Mesh Ansicht frontal hinten

Im zweiten Schritt wurden nun die seitlichen Meshes außen vorgelassen und lediglich mit den restlichen Beiden gearbeitet. In diesem Fall ist deutlich erkennbar, dass Löscher über den kompletten Kopf entstanden sind. Das Modell wurde wiederum mit dem Filter „Poisson Surface Reconstruction“ aufgefüllt, dieses Mal allerdings mit einer „Octree Depth“ von 6. Hintergrund dessen ist, dass bei einer Einstellung des Faktors 8 das Programm zum Absturz kam, da das Berechnen zum Auffüllen der Löcher mit dieser Genauigkeit nicht mehr funktioniert hat. Mit dem Faktor 6 konnte allerdings ebenfalls ein erfolgreiches 3D-Modell rekonstruiert werden. Vergleicht man allerdings Abbildung 9 und 10 miteinander, ist die Qualität der Genauigkeit deutlich zu erkennen.

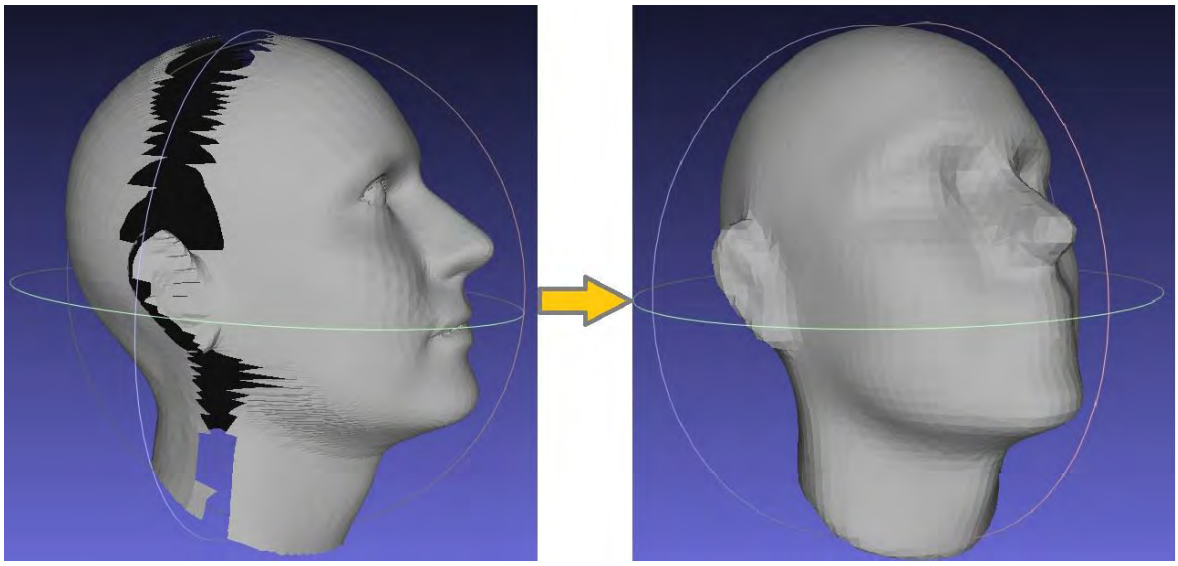


Abbildung 10: Erfolgreiche Rekonstruktion

Nach dieser Erkenntnis konnte mit der Rekonstruktion von realen Modellen begonnen werden. Dazu wurden einige Probanden fotografiert und daraus Meshes erstellt. Anschließend wurde versucht, daraus ein 3D-Modell zu erstellen. Ein dabei entstandenes erfolgreiches Modell zeigt sich in Abbildung 11. In diesem Fall wurden vier Meshes verwendet:

- Mesh Ansicht frontal vorne
- Mesh Ansicht seitlich links

- Mesh Ansicht seitlich rechts
- Mesh Ansicht frontal hinten

Dadurch konnten die Meshes genauer miteinander verbunden und die Löcher auf das Minimalste verringert werden. Das Auffüllen mit Hilfe des Filters „Poisson Surface Reconstruction“ und einer „Octree Depth“ von 8 wurde somit erfolgreich durchgeführt und es entstand eine Büste, welche anschließend sogar mit einem 3D-Drucker gedruckt werden konnte.

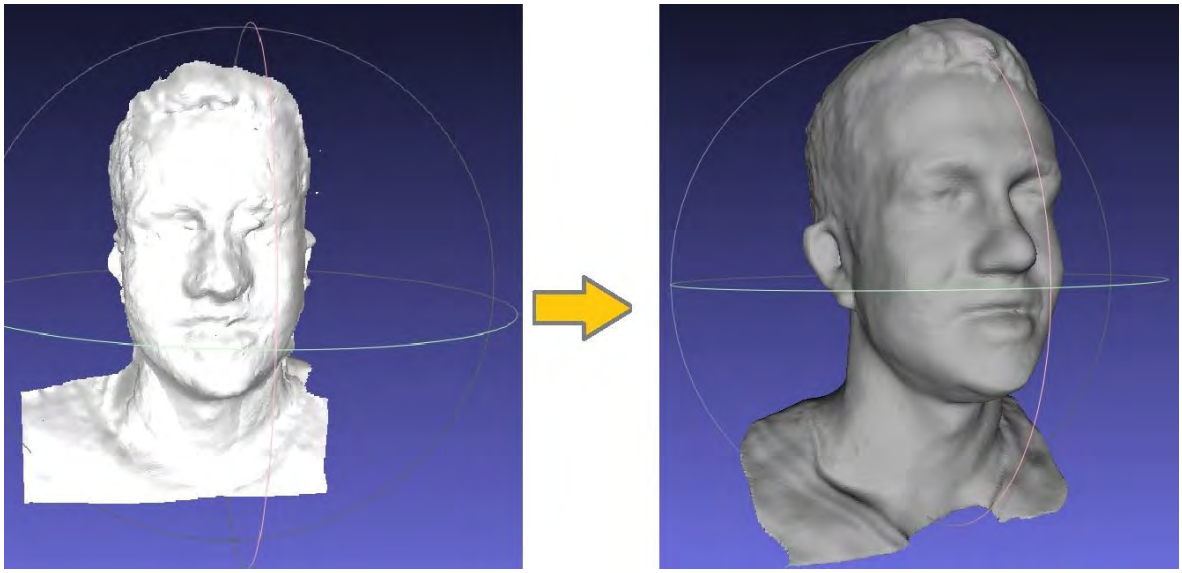


Abbildung 11: Erfolgreiche Rekonstruktion

5.2 Fehlerhafte Rekonstruktionen

Neben den erfolgreichen Rekonstruktionen gab es allerdings auch einige fehlgeschlagene Versuche. Da das Ziel des Projektes war, eine 3D-Büste mit Hilfe von möglichst wenig Bildern zu erstellen, gestaltete sich die Rekonstruktion oft sehr schwierig. Abbildung 12 zeigt eine Rekonstruktion desselben Modelles aus Abbildung 11. Nun wurde jedoch folgendes verwendet:

- Mesh Ansicht frontal vorne
- Mesh Ansicht schräg hinten seitlich links
- Mesh Ansicht schräg hinten seitlich rechts

Bei dieser Konstellation gestaltete sich bereits das punktuelle Anheften der einzelnen Meshes schwieriger, weil wenige gemeinsame Punkte gefunden wurden. Wie des Weiteren ersichtlich ist, sind die einzelnen Meshes außerdem nicht ideal

erstellt worden. Die Löcher sind in diesem Fall sehr groß und durch die entstandenen Wände zeigen die Normalen in die verschiedensten Richtungen, weshalb die ersichtlichen Verformungen entstanden sind. Auch der vermehrte Versuch des punktuellen Anheftens und das Entfernen der Wände führte hierbei zu keinem akzeptablem Ergebnis.

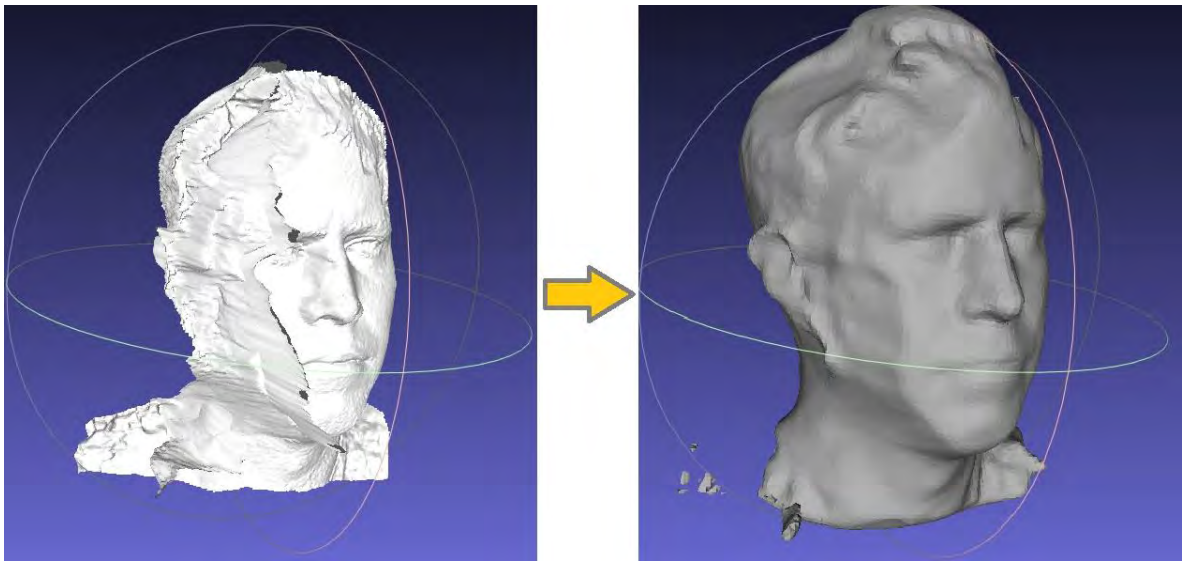


Abbildung 12: Fehlgeschlagene Rekonstruktion

Der letzte hier aufgezeigte Fall zeigte sich als eines der schwierigsten Fälle. Linkes Bild der Abbildung 13 stellt das Modell als eingefärbte Punktewolke dar, damit die Löcher, welche beim Erstellen der Meshes entstanden sind, deutlicher ersichtlich werden. Das große Loch an der Stirn des Probanden konnte mit Hilfe des Filters „Poisson Surface Reconstruction“ nicht passend rekonstruiert werden und führte zu einer Einbuchtung im entstanden 3D-Modell. Als größtes Problem stellten sich allerdings der nicht dargestellte Rücken und die Seiten der Schultern da. Dieses Loch war für die Rekonstruktion zu groß und bei dem Versuch bildete sich eine nicht definierbare Wand in sämtliche Richtungen.

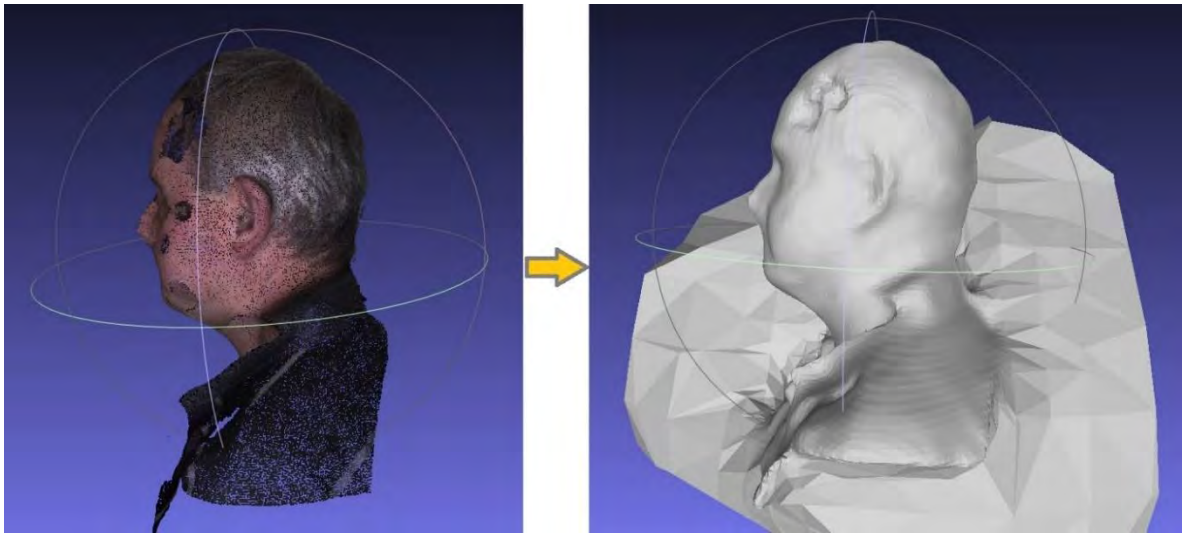


Abbildung 13: Fehlgeschlagene Rekonstruktion

5.3 Vergleich eines Modells mit verschiedenen Einstellungen

Bei den erfolgreichen und nicht erfolgreichen Modellen haben sich diese teilweise an den einzelnen Meshes oder der Verwendung der einzelnen Meshes unterschieden. Ein 3D-Modell entstanden durch dieselbe Punktewolke kann allerdings auch auf unterschiedliche Weise durch verschiedene Einstellungen des Filters „Poisson surface Reconstruction“ gebildet werden.

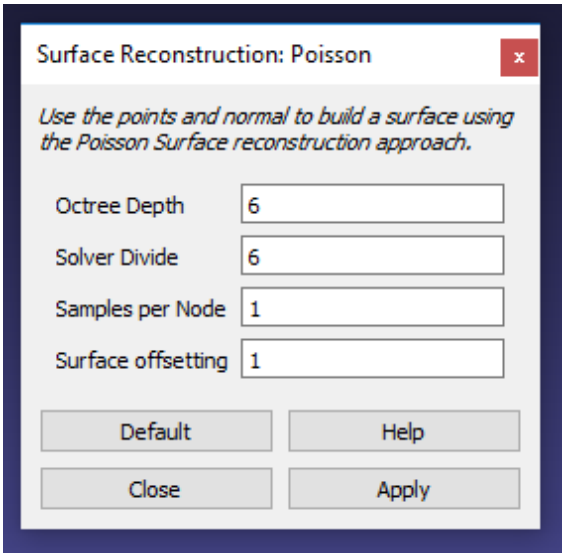


Abbildung 14: Filtereinstellung Poisson surface reconstruction

Dabei gibt *Octree Depth* die Tiefe des Baumes an, *Solver Divide* die Lösungsteilung, *Samples per Node* die Anzahl der Punkte pro Knoten und *Surface offsetting* die Oberflächenabsetzung. Für diesen Vergleich wurden einige Versuche durchgeführt

und die Werte in der folgenden Tabelle festgehalten. Im Anhang gibt es zu jedem Versuch, insofern dieser nicht zum Programmabsturz geführt hat, das entsprechende Ergebnis der Rekonstruktion.

	Octree Depth	Solver Divide	Samples per Node	Surface offsetting	Zeit	CPU-Auslastung
1	1	1	1	1	00:00.13	4,7 %
2	2	1	1	1	00:00.47	5,6 %
3	3	1	1	1	00:00.93	9,3 %
4	3	2	1	1	00:00.98	8,4 %
5	4	1	1	1	00:01.31	21,4 %
6	4	2	1	1	00:01.12	14,2 %
7	4	3	1	1	00:00.93	12,1 %
8	5	3	1	1	00:01.37	27,5 %
9	5	4	1	1	00:01.27	21,8 %
10	5	4	3	1	00:01.22	19,6 %
11	6	6	1	1	00:01.93	42,8 %
12	6	5	1	1	00:03.99	46,2 %
13	7	6	1	1	00:10.01	98 %
14	7	5	1	1	Program mabsturz	100 %
15	7	6	5	1	00:06.34	89 %
16	8	6	1	1	Program mabsturz	100 %
17	8	7	1	1	00:58.52	98,5 %
18	10	9	1	1	02:01.32	99,3 %

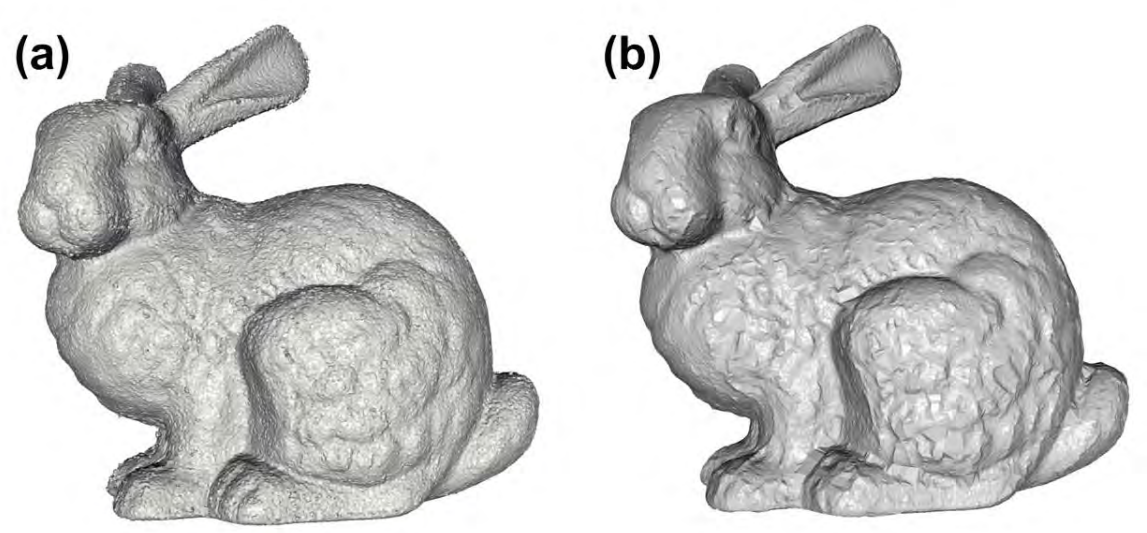
Einstellungen mit geringen Werten bei Octree Depth wie Versuch 1, 2, 3 oder 4 rekonstruieren Modelle in äußerst schneller Zeit und mit einer geringen CPU-Auslastung. Anhand den dabei entstandenen Bildern können die Dreicksflächen sehr gut erkannt werden. Die Rekonstruktion ähnelt einem Tetraeder. Vom ursprünglichem Kopf ist hierbei allerdings noch nichts erkennbar. Versuche 5 bis 10 hingegen lassen bereits die ursprüngliche Punktewolke erkennen und das bei nur geringer Erhöhung von Zeit und Auslastung. Deutlich zeigt sich hier, wie die Dreiecksflächen immer kleiner werden und sich die Formen von Augen, Mund, Nase und Ohren mit Anheben der Werte verdeutlichen. Versuch 11 (Default-Einstellungen) und Versuch 12 weisen nun einen großen Sprung in der CPU-Auslastung aus und auch die Zeit benötigt jetzt deutlich länger. Erschreckend hierbei ist jedoch, dass die Rekonstruktion in diesem Fall fehlgeschlagen und an

der Seite des Gesichtes ein nicht erklärbares Artefakt entstanden ist. Versuch 14 und 16 führten zu einem Programmabsturz und die restlichen Versuche lieferten sehr gute Ergebnisse. Deutlich erkennbar ist allerdings, dass sich an den Bildern bei den beiden letzten Versuchen kein Unterschied mehr feststellen lässt, allerdings nahezu die doppelte Zeit dafür benötigt wird. Daher zeigt sich also insgesamt, dass die Einstellung 8,6,1,1 mit einem leistungsstarken Rechner die besten Ergebnisse in angemessener Zeit liefert und sich Octree Depth und Solver Dived lediglich um 1 unterscheiden sollen. Allerdings sind die Werte hier mit einer optimalen gleichmäßig verteilten Punktewolke entstanden. Unregelmäßigere Modelle oder Modelle aus zusammengesetzten Meshes können dementsprechend länger dauern und nicht die gewünschten Erfolge liefern.

6. Vergleich mit anderen Methoden

Um eine Aussage über die Qualität der 3D-Rekonstruktion mit Hilfe des Poisson-Verfahrens machen zu können, wurde diese Methode mit anderen Ergebnissen verglichen, die durch verschiedene bereits bestehende Methoden zur 3D-Rekonstruktion entstanden sind. Dabei war das Testobjekt der sogenannte „Standford bunny“. Eine rohe Datenmenge von 362.000 Punkten, die durch zehn bewegten Fotos zusammengesetzt wurde. Diese Daten wurden so verarbeitet, dass jeder Algorithmus damit arbeiten konnte. Abbildung 14 zeigt das Ergebnis dieses Vergleiches. Dabei wurden folgende Rekonstruktionsmethode angewandt:

- Bild (a): Power Crust
- Bild (b): Robust Cocone
- Bild (c): Fast RBF
- Bild (d): MPU
- Bild (e): Hoppe et al’s reconstruction
- Bild (f): VRIP
- Bild (g): FFT-basierte Rekonstruktion
- Bild (h): Poisson Rekonstruktion



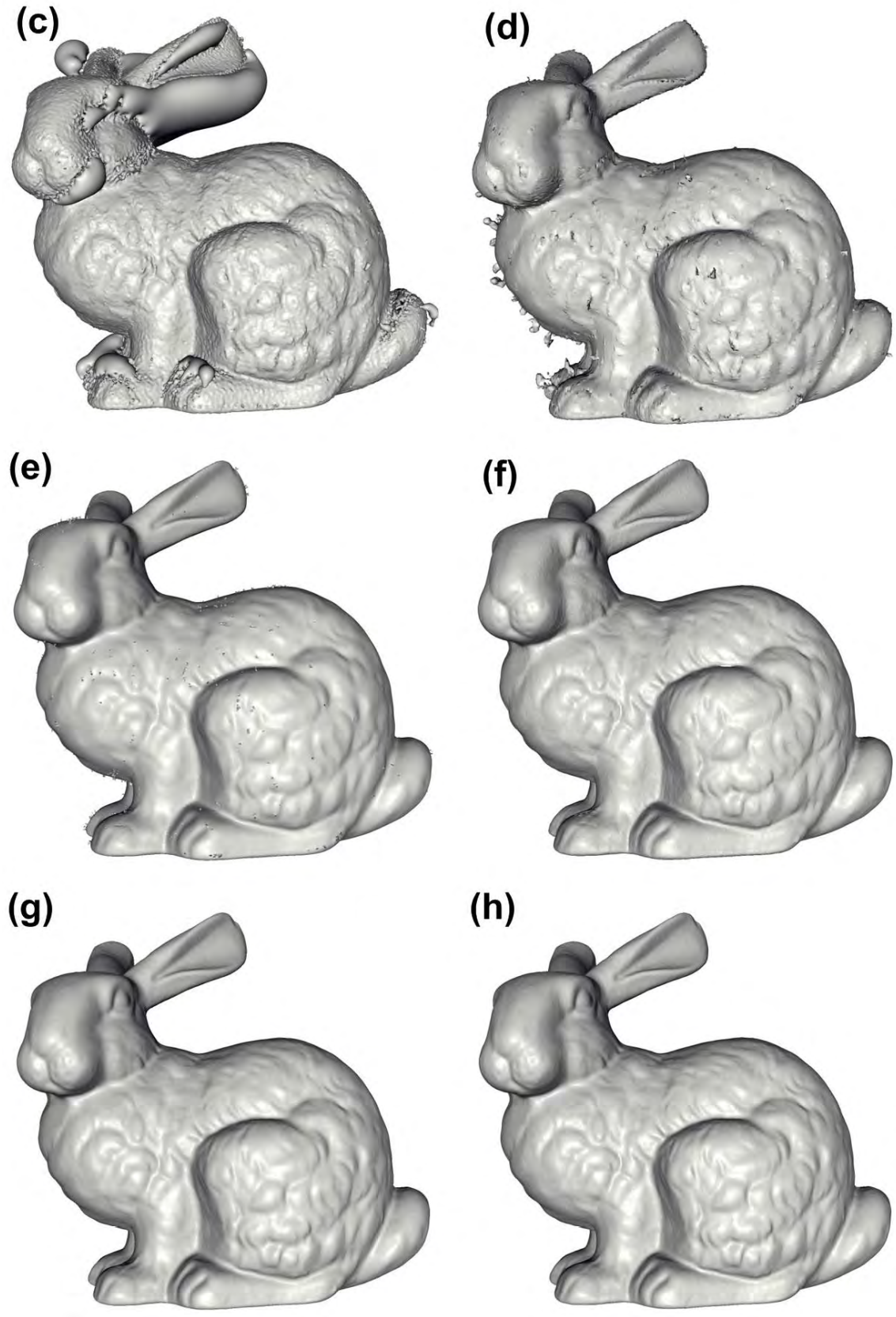


Abbildung 15: Rekonstruktion des "Standford Bunnys"

6.1 Vergleich mit VRIP

Eine Herausforderung von Oberflächenrekonstruktion ist die Wiederherstellung von scharfen Eigenschaften. Für einen Vergleich bezüglich dieser Herausforderung wurde die Poisson-Methode mit der VRIP-Methode anhand des Modells „Happy Buddah“ in Abbildung 15 verglichen. Hier ist deutlich ersichtlich, dass die VRIP-Methode zu einem Überlappungsphänomen neigen. Dies lässt sich dadurch erklären, dass die VRIP’s Funktion senkrecht zu der Sichtrichtung wächst und nicht zu den Oberflächennormalen. Im Gegensatz dazu ist die Poisson-Methode unabhängig zur Sichtrichtung und rekonstruiert daher den Kern des Fragmentes sorgfältiger.

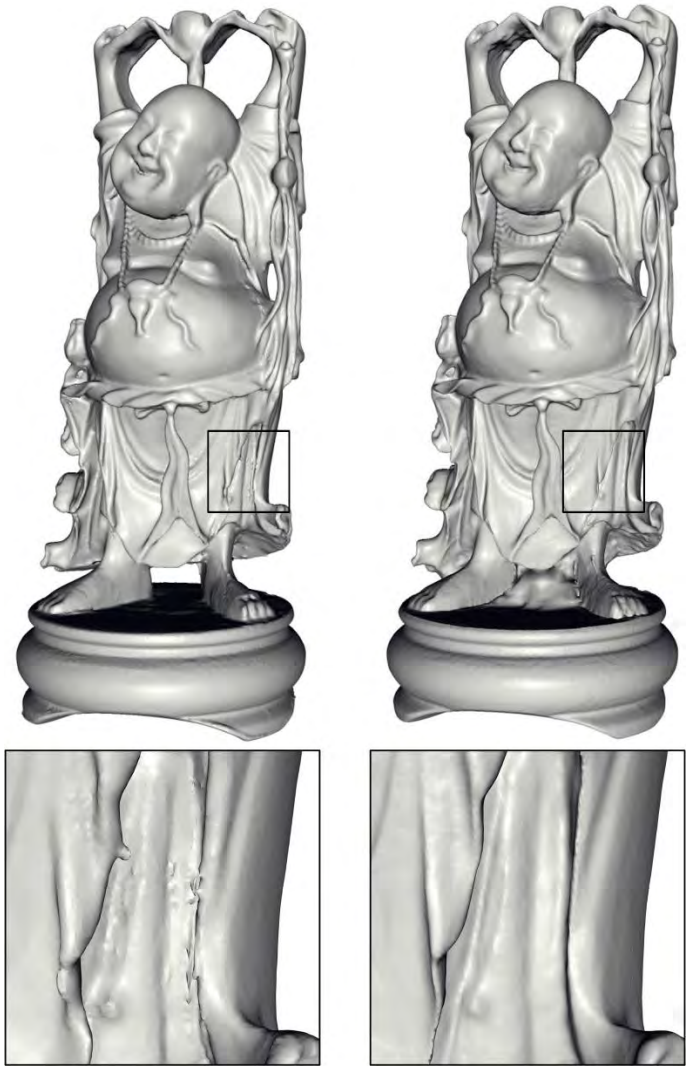


Abbildung 16: Rekonstruktion des "Happy Buddah"

6.2 Vergleich mit FFT-basierter Rekonstruktion

In Abbildung 14 ist sehr deutlich erkennbar, dass die Rekonstruktion mit Hilfe der Poisson-Methode derer, welche durch die Rekonstruktion mit Hilfe der FFT-basierten Methode entstanden ist, am ehesten gleicht. Die Gemeinsamkeit ist eine Bestätigung, dass die Poisson-Methode herausstechende, hochfrequente Informationen nicht verwirft. Außerdem wurde somit bekräftigt, dass diese Methode eine hohe Anpassungsfähigkeit aufweist. Dass die Rekonstruktionen bei gleichmäßig verteilten Punktproben dasselbe Ergebnis liefern ist ein Zeichen dafür, dass die Benutzung des Anpassungsfähigen Weiten-Filters bei der Poisson-Methode bessere Rekonstruktionsergebnisse bei nicht gleichmäßig verteilten Punktproben liefert. Dies wurde bereits in Abbildung 8 bestätigt [7].

7. Fazit und Ausblick

7.1 Fazit

Ein Fazit, welches ich aus dem Projekt zur 3D-Rekonstruktion ziehen kann ist das Modell unter bestimmten Gesichtspunkten zu wählen.

Zum einen sollten die Modelle im Falle von Brillenträgern diese für die Erstellung der Bilder zur Rekonstruktion abnehmen. Brillen liegen zwar an den Kopfseiten eng an der Haut eines Menschen auf, allerdings sind um die Augen rum Löcher vorhanden. Diese Löcher können nicht rekonstruiert werden. Entweder es entstehen unerwünschte Artefakte oder die Brille wird so an den Kopf angeheftete Rekonstruiert, dass nach unnatürlichen Augenumrandungen aussieht, jedoch nicht als Brille erkannt werden kann. Des Weiteren sollten bei der Modellwahl die Lage der Haare beachtete werden. Viele vom Kopfe abstehende Strähnen oder Pferdeschwänze bei Frauen führen ebenfalls zu den bereits als kritisch bekannten Löchern. Aus diesem Grund sollte versucht werden, die Haare so gut wie möglich am Kopf anliegen zu lassen. Das zählt auch für die Kleidung des Modelles. Je nachdem wie viel vom Körper mit rekonstruiert werden soll, sollte auch hier darauf geachtete werden, dass die Kleidung nicht zu sehr vom Körper absteht.

Insgesamt kann also zusammengefasst werden, dass bei der Modellwahl darauf zu achten ist, dass die sogenannten Löcher so gut wie möglich vermieden werden.

7.2 Ausblick

Auch wenn es für viele nicht den Anschein erweckt, spielt die 3D-Rekonstruktion heutzutage eine große Rolle. Es wird stetig daran gearbeitet, dieses Gebiet weiter zu entwickeln. Modernste 3D-Laserscanner vereinfachen die Arbeit in einigen Gebieten jedoch enorm. So können zum Beispiel Menschen komplett eingescannt werden, es entstehen keine Teil-Meshes mehr, sondern ein vollständiger Mesh ohne Löcher, welcher wiederum ohne Probleme aufgefüllt werden kann. Das Rekonstruieren von Menschen stellt jedoch nur einen Teil dieser Arbeit da. Die 3D-Rekonstruktion wird beispielweise auch in der Medizin verwendet. Dabei werden aus Teilröntgenbildern die entsprechenden Organe rekonstruiert. Auch hier wird mittlerweile versucht, aus so wenig Bildern wie möglich eine weitestgehend genaue Rekonstruktion zu erhalten. Dies hat den Vorteil, dass dadurch die

Röntgenstrahlung verringert werden kann. Trotz der modernsten Technik ist allerdings die Methode der 3D-Rekonstruktion aus 2D-Bildern immer noch ein aktuelles Thema. Ein dabei bekanntes Gebiet ist das Restaurieren von alten historischen Gebäude. Mit Hilfe von Bildern des ursprünglichen Gebäudes können 3D-Modelle erstellt werden. Anhand dieser wiederum können im Anschluss die verfallenen Gebäudeteile nachgebaut werden. Aber auch Bergwerke, welche schwer zugänglich oder nicht scannbar sind, können so für die tägliche Arbeit und Sicherheit der Arbeiter nachgebaut werden.
Die 3D-Rekonstruktion mit Hilfe des Poisson-Verfahrens wird also auch in Zukunft noch eine Rolle spielen.

8. Anhang

Anhang 1: Bilder zum Vergleich von Rekonstruktionen

Anhang 2: Quellverzeichnis [7] als PDF auf CD

Anhang 3: Bachelorarbeit als PDF auf CD

8.1 Anhang:

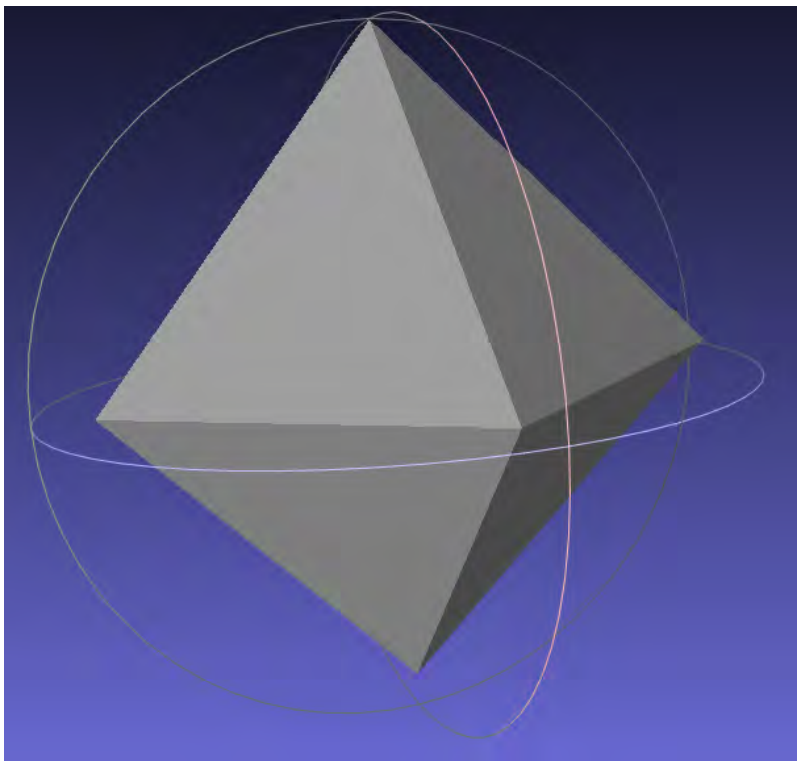


Abbildung 17: Versuch 1

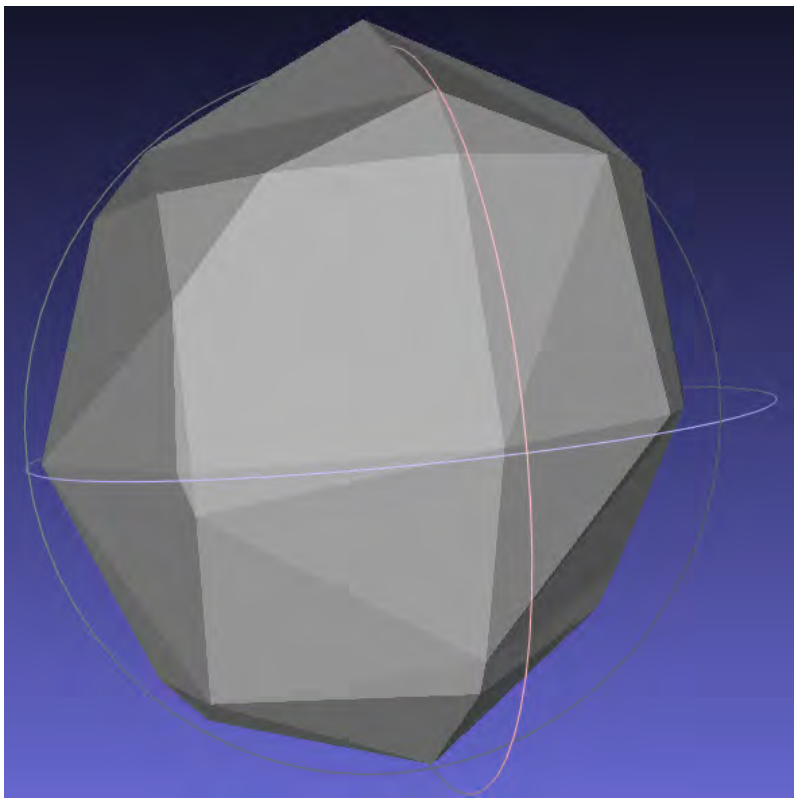


Abbildung 18: Versuch 2

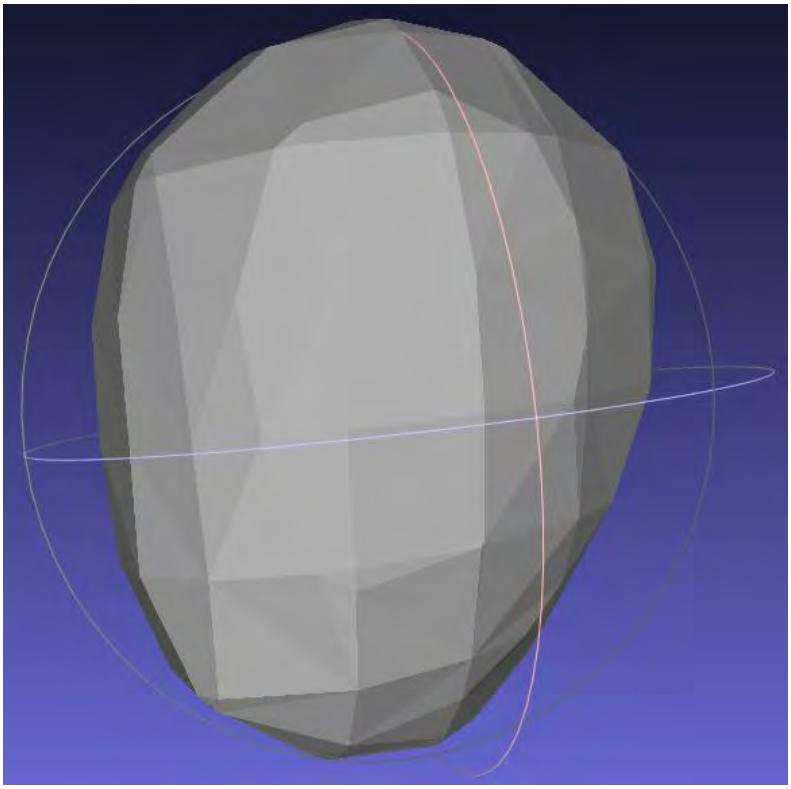


Abbildung 19: Versuch 3

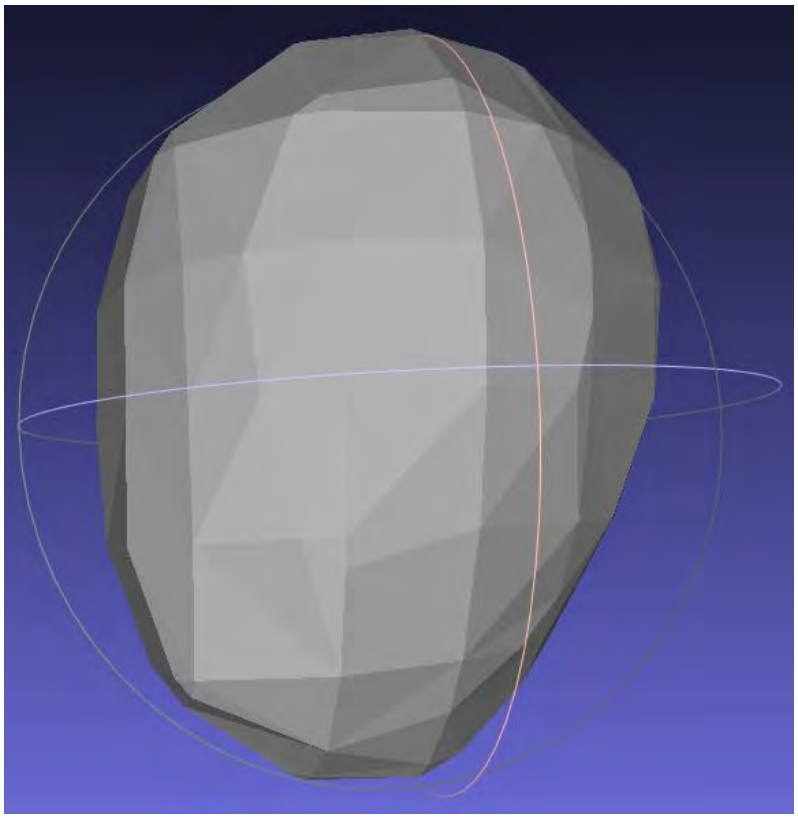


Abbildung 20: Versuch 4

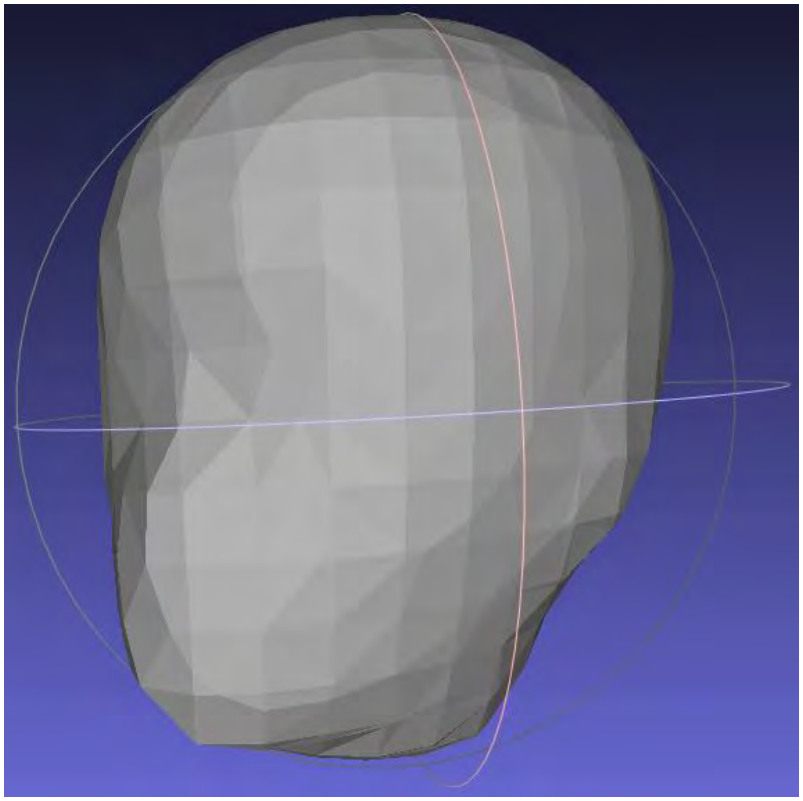


Abbildung 21: Versuch 5

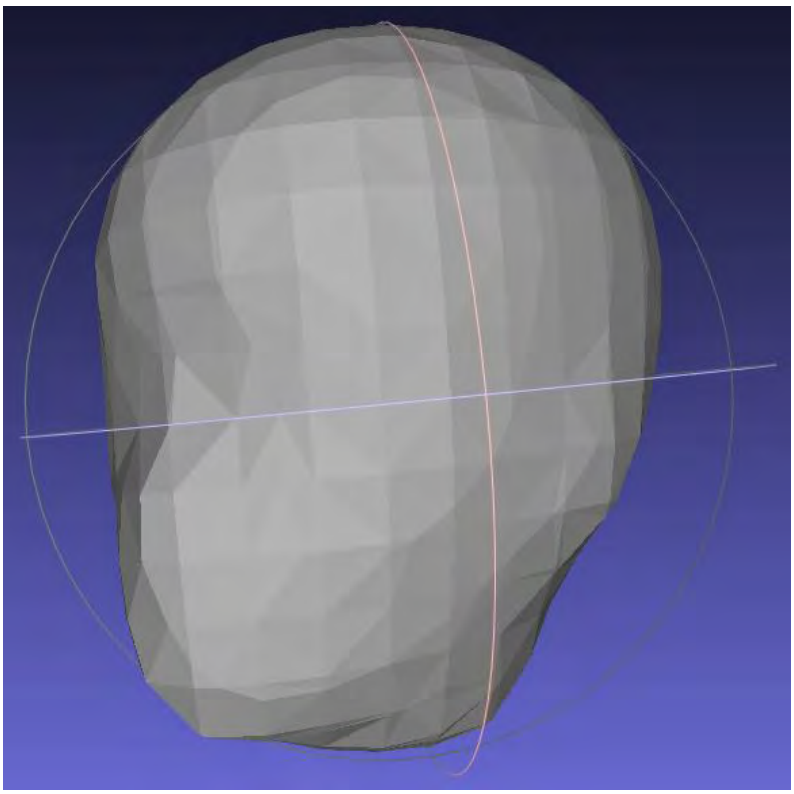


Abbildung 22: Versuch 6

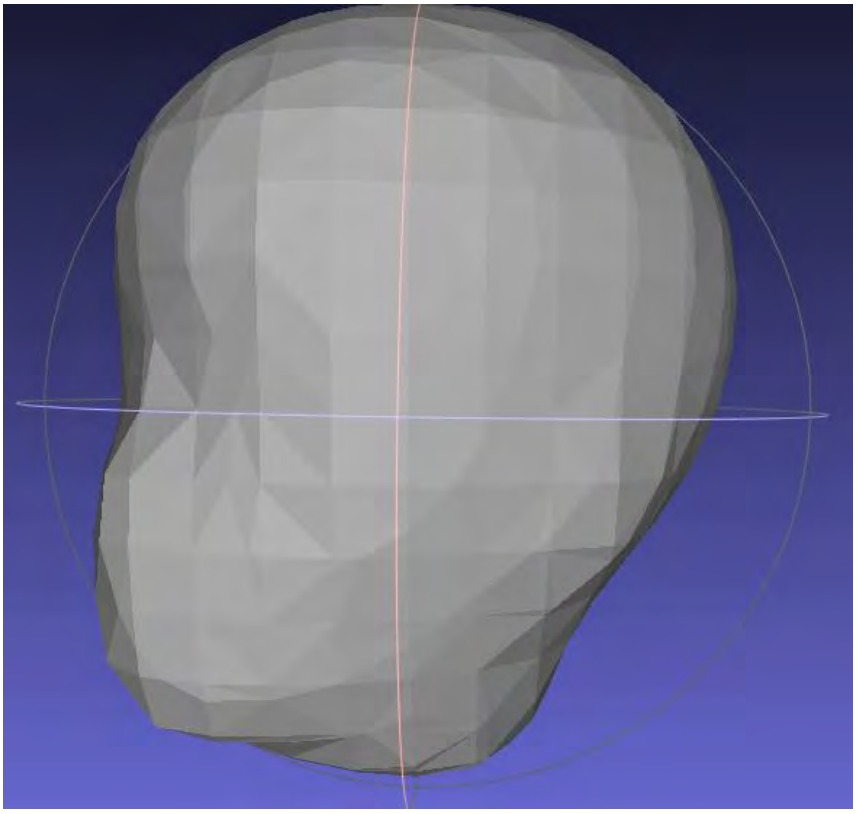


Abbildung 23: Versuch 7

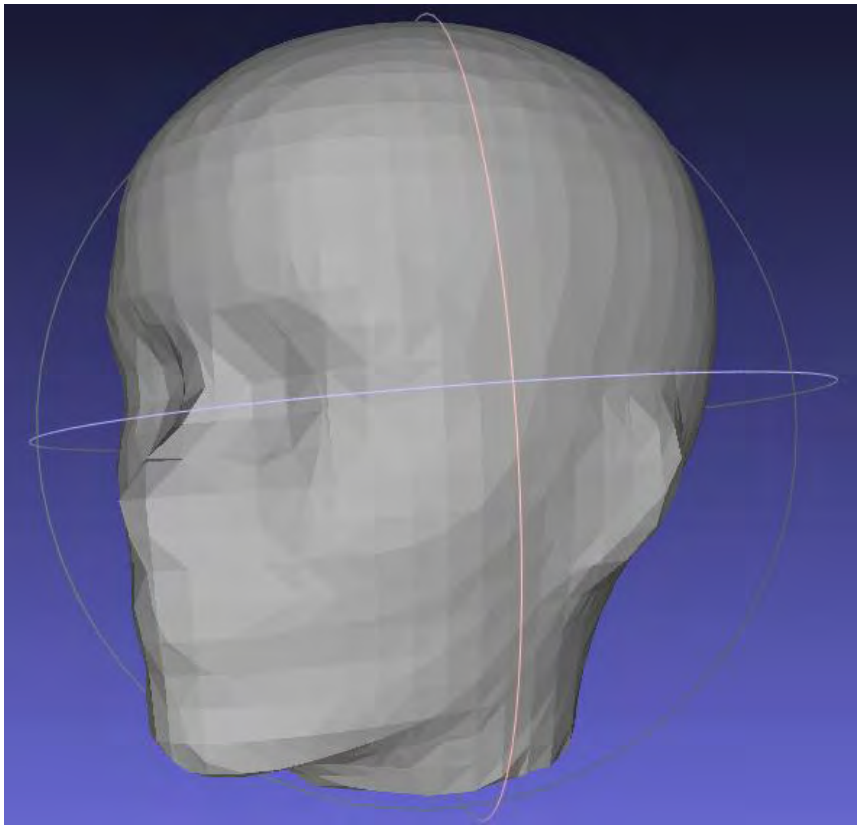


Abbildung 24: Versuch 8

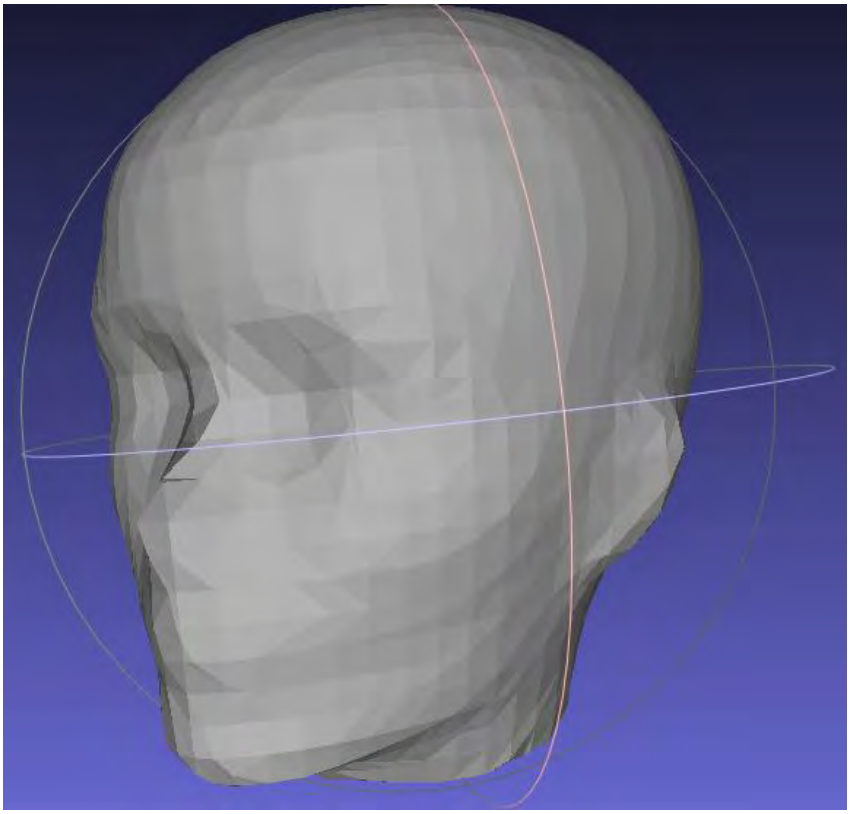


Abbildung 25: Versuch 9

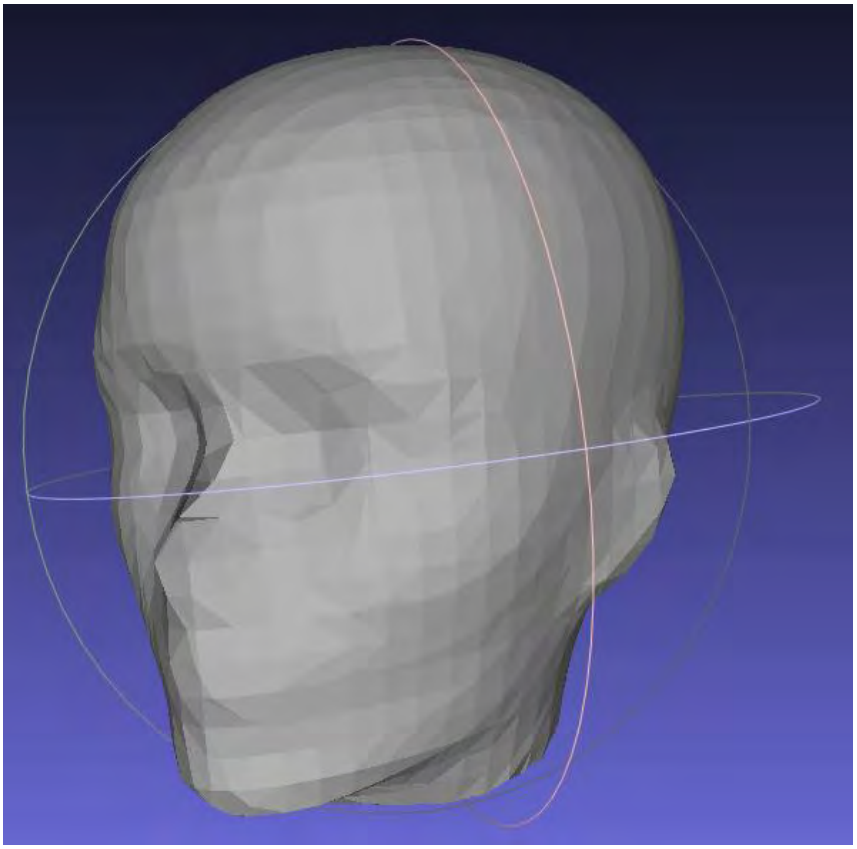


Abbildung 26: Versuch 10

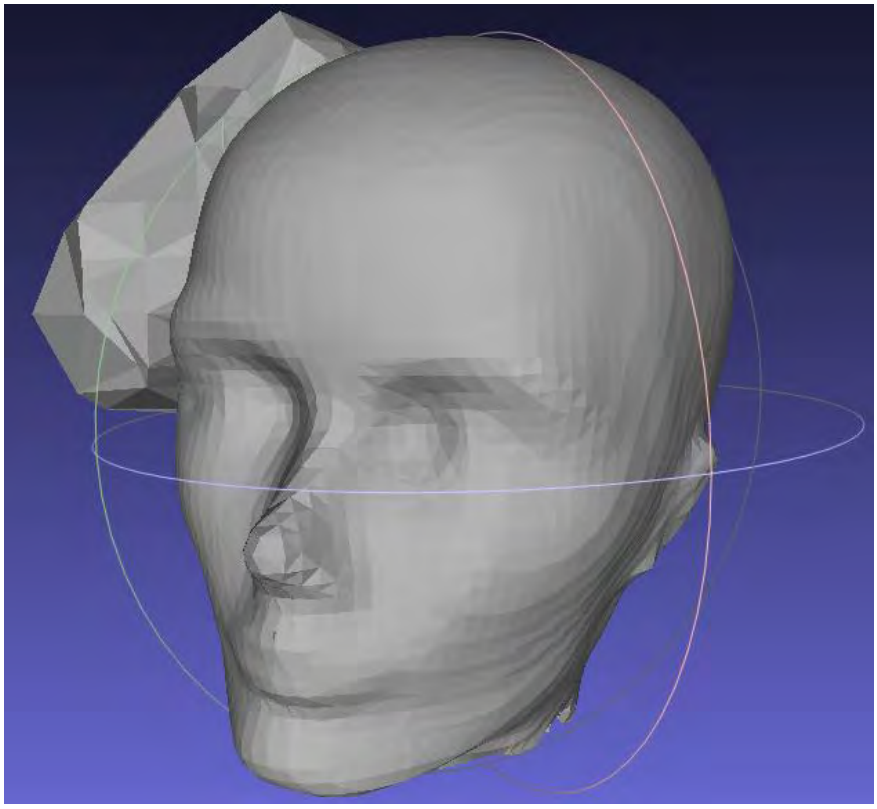


Abbildung 27: Versuch 11

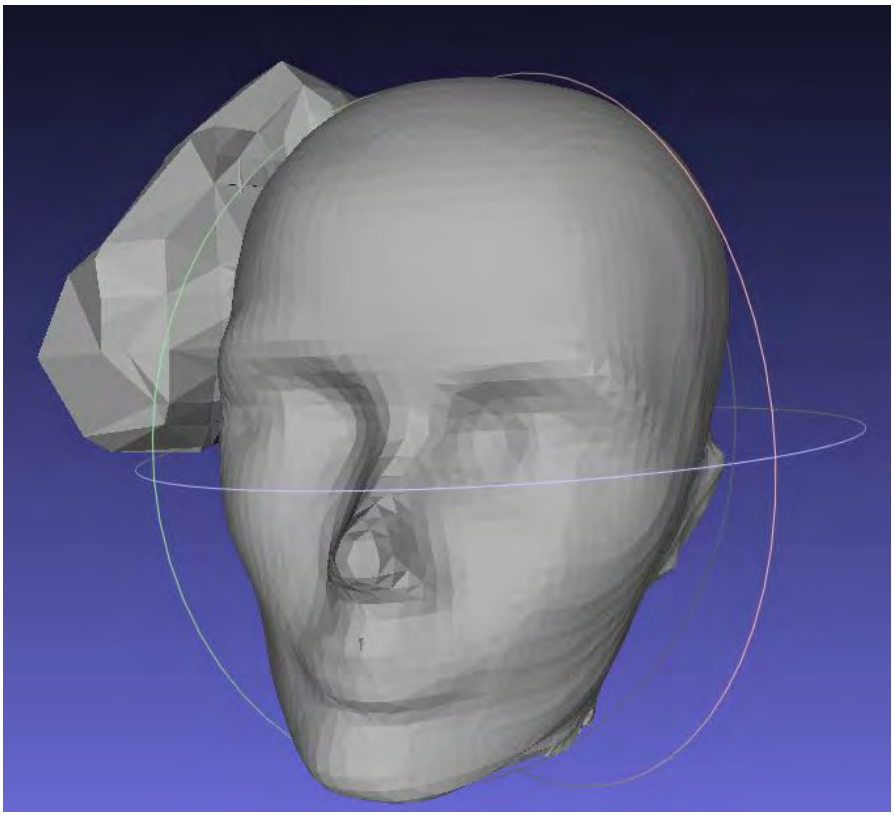


Abbildung 28: Versuch 12

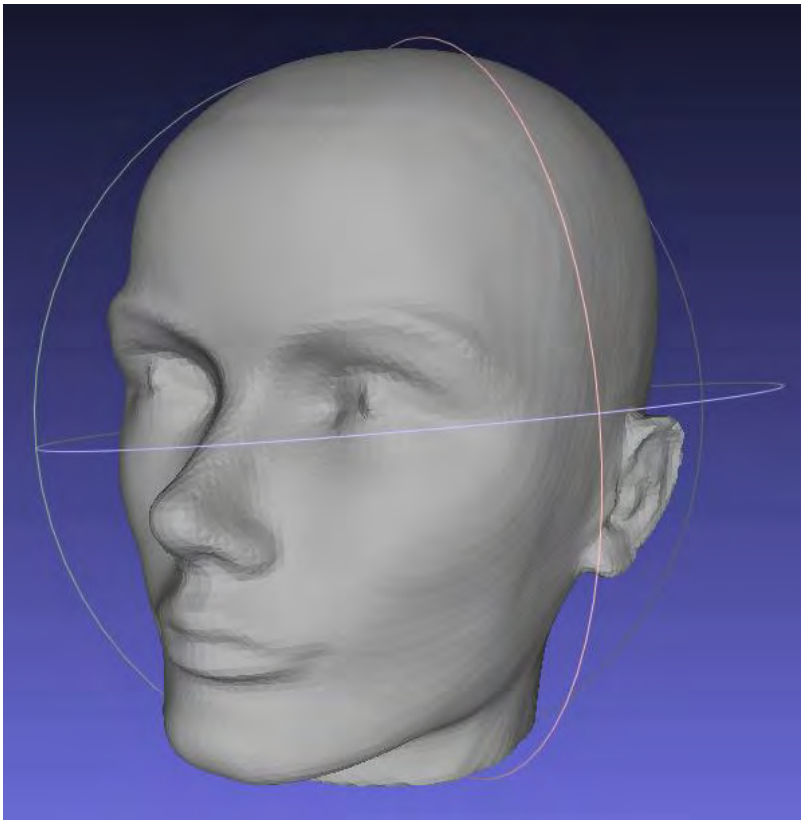


Abbildung 29: Versuch 13

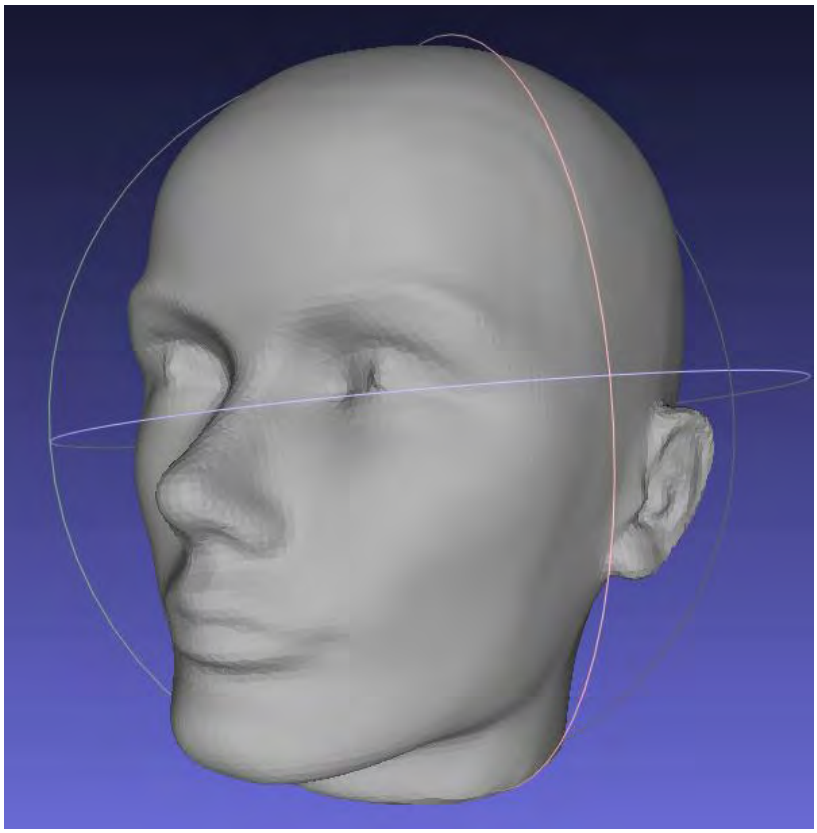


Abbildung 30: Versuch 15

Abbildung 31: Versuch 17

Abbildung 32: Versuch 18

9. Abkürzungsverzeichnis

M

MPU ..... Multi-Level Partition of Unity Implicits

O

OFF ..... Object File Format

P

PLY ..... Polygon File Format

R

RBF ..... Radial Basis Functions

S

STL ..... STereoLithography

V

VGG-Bibliothek ..... Visualization and Computer Graphics Library
VRIP ..... Volumetric Range Image Processing

10. Abbildungsverzeichnis

Abbildung 1: Aus Bildern eine Büste erstellen 5
Abbildung 2: Octree-Struktur 7
Abbildung 3: Einheitswürfel 7
Abbildung 4: Programm Meshlab 9
Abbildung 5: Grundsätzliche Idee des Poisson-Verfahrens 12
Abbildung 6: Octree θ über Punktprobe S 14
Abbildung 7: Triangulierung aus Marching-Cube-ähnlichem Algorithmus 17
Abbildung 8: Punktprobe (a) ; FFT-Annäherung (b) ; Poisson-Rekonstruktion (c) 19
Abbildung 9: Erfolgreiche Rekonstruktion 21
Abbildung 10: Erfolgreiche Rekonstruktion 22
Abbildung 11: Erfolgreiche Rekonstruktion 23
Abbildung 12: Fehlgeschlagene Rekonstruktion 24
Abbildung 13: Fehlgeschlagene Rekonstruktion 25
Abbildung 14: Filtereinstellung Poisson surface reconstruction 25
Abbildung 15: Rekonstruktion des "Standford Bunnys" 29
Abbildung 16: Rekonstruktion des "Happy Buddah" 30
Abbildung 17: Versuch 1 35
Abbildung 18: Versuch 2 35
Abbildung 19: Versuch 3 36
Abbildung 20: Versuch 4 36
Abbildung 21: Versuch 5 37
Abbildung 22: Versuch 6 37
Abbildung 23: Versuch 7 38
Abbildung 24: Versuch 8 38
Abbildung 25: Versuch 9 39
Abbildung 26: Versuch 10 39
Abbildung 27: Versuch 11 40
Abbildung 28: Versuch 12 40
Abbildung 29: Versuch 13 41
Abbildung 30: Versuch 15 41
Abbildung 31: Versuch 17 42

Abbildung 32: Versuch 18..... 42

11. Quellverzeichnis

[1] „Duden,“ [Online]. Available: http://www.duden.de/suchen/dudenonline/gradient. [Zugriff am 25 07 2016].

[2] „WU-wien,“ [Online]. Available: http://statistik.wu-wien.ac.at/~leydold/MOK/HTML/node123.html. [Zugriff am 25 07 2016].

[3] „Wikipedia,“ [Online]. Available: https://de.wikipedia.org/wiki/Isofl%C3%A4che. [Zugriff am 25 07 2016].

[4] „Wikimedia,“ [Online]. Available: https://commons.wikimedia.org/wiki/Category:Octrees?uselang=de. [Zugriff am 25 07 2016].

[5] „Interpolation methods,“ [Online]. Available: http://paulbourke.net/miscellaneous/interpolation/. [Zugriff am 02 09 2016].

[6] „MeshLab,“ [Online]. Available: http://meshlab.sourceforge.net/. [Zugriff am 27 07 2016].

[7] A. S. Konrad Polthier, „Poisson Surface Reconstruction,“ Eurographics Symposium on Geometry Processing, 2006.