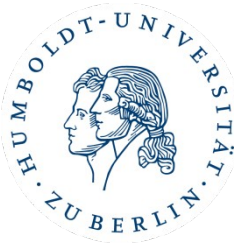


Poisson Surface Reconstruction

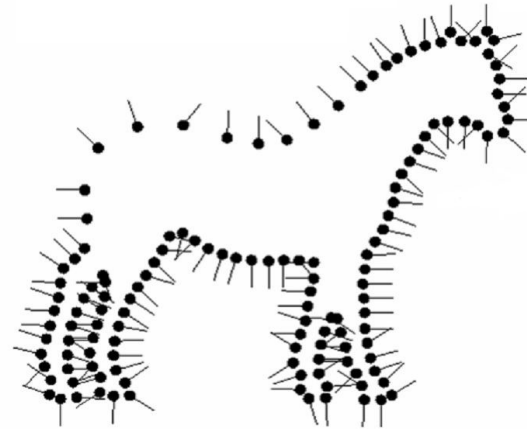
Peter Hagemann
Andreas Meyer

Seminar Visual Computing



Motivation

- Zur 3D Darstellung von Objekten werden meist Scan-Daten erstellt



- Erstellung eines Dreieckmodells aus einer orientierten Punktvolke
 - Zum Effizienten Rendern auf Grafikkarten
 - Ermöglicht Kollisionsabfrage
 - Hilfreich für Weiterverarbeitung der Objekte

Motivation

- Ziel ist es die Punkte so zu verbinden, dass das Objekt dem realen möglichst gut gleicht
- Problem: Daten sind möglicherweise fehlerhaft/verrauscht
- Es existieren bereits einige Verfahren der Realisierung, welche die Punkte interpolieren (z.B. Delaunay Triangulation), sowie lokale und globale Methoden

Vor und Nachteile der Verfahren

- Interpolation:
 - + Dichte Oberfläche
 - anfällig gegenüber rauschen und Ausreißern, benötigt meist zusätzliche Weichzeichnung (Tiefpassfilterung)
- globale Verfahren:
 - + keine lokale Heuristik benötigt
 - Fehler durch Approximation mit Basisfunktionen
- Lokale Verfahren:
 - + durch heuristische Betrachtung in der Umgebung der Punkte glattere Oberfläche
 - hoher Aufwand für jeden Punkt

Poisson Surface Reconstruction

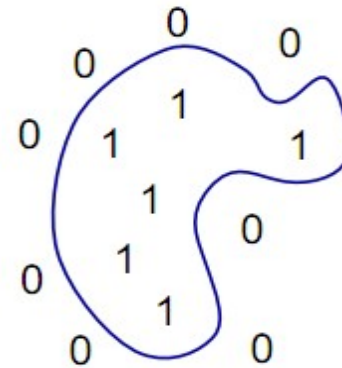
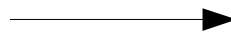
- Kombiniert lokale und globale Vorteile
 - Benötigt keine lokalen Heuristiken
 - Basisfunktionen mit Oberfläche verbunden, statt der Punktdaten

Poisson Surface Reconstruction: Idee

- Berechnen einer Indikatorfunktion, welche das Volumen des Objekts approximiert



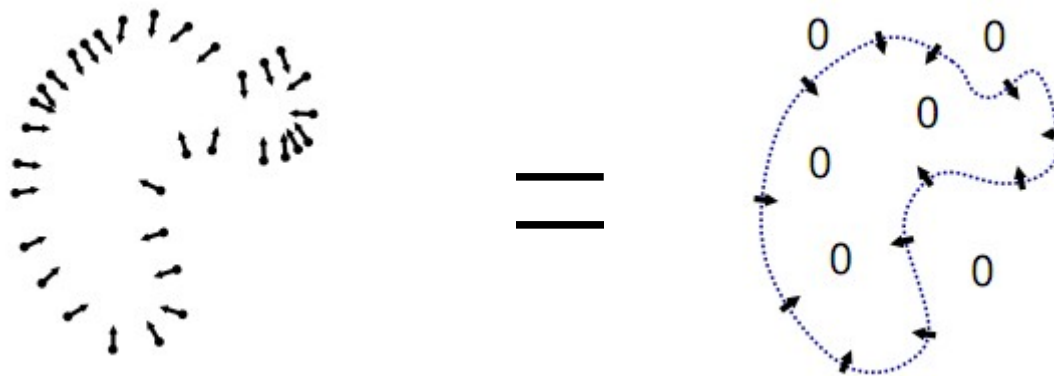
Orientierte Punktwolke (\vec{V})



Berechnete Indikatorfunktion (χ_M)

Poisson Surface Reconstruction: Idee

- Um dies zu erreichen versucht man den Indikatorgradienten den orientierten Punkten anzunähern
- Der Gradient hat die Eigenschaft, dass er nur an der Oberfläche ungleich Null ist



Orientierte Punktvolke (\vec{V})

Indikatorgradient ($\nabla \chi_M$)

$$\vec{V} = \nabla \chi_M$$

Poisson Surface Reconstruction: Idee

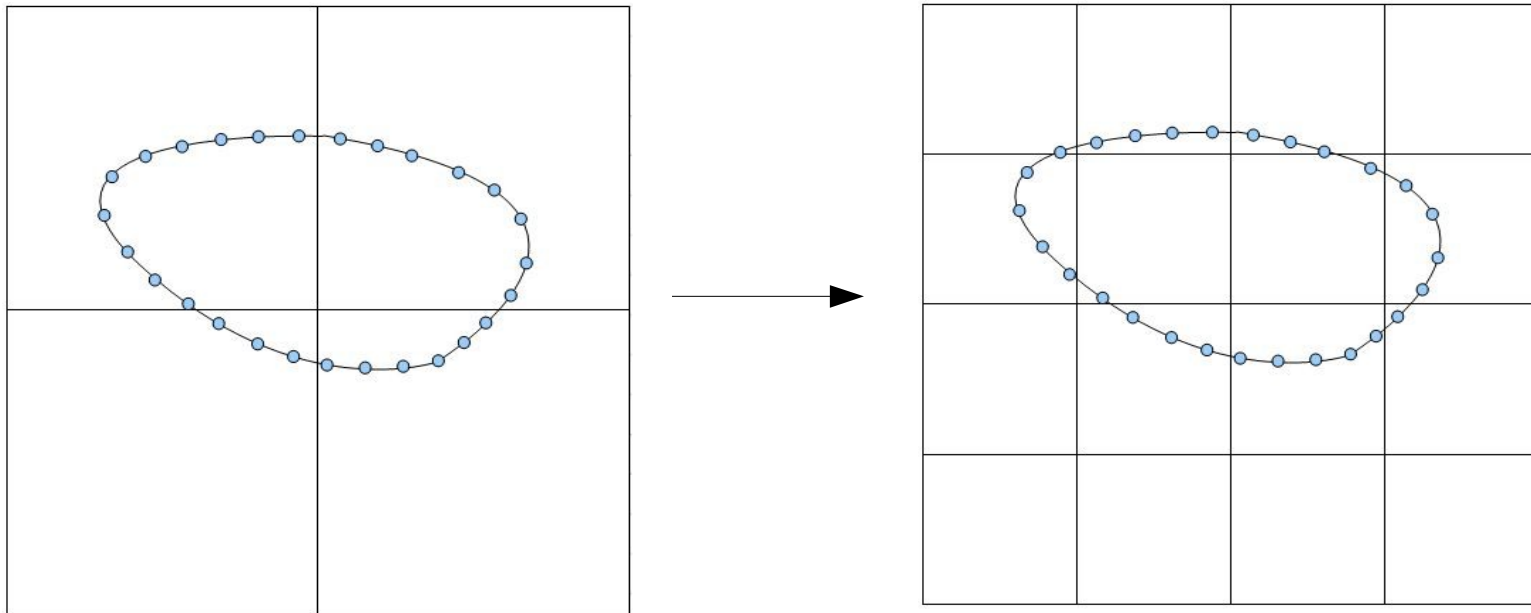
- Da \vec{V} im allgemeinen nicht integrabel ist, wendet man die Divergenz auf beide Seiten der Gleichung an und erhält somit eine Poissongleichung

$$\Delta \chi = \nabla \cdot \vec{V}$$

$$\text{mit } \Delta \chi = \nabla \cdot \nabla \chi$$

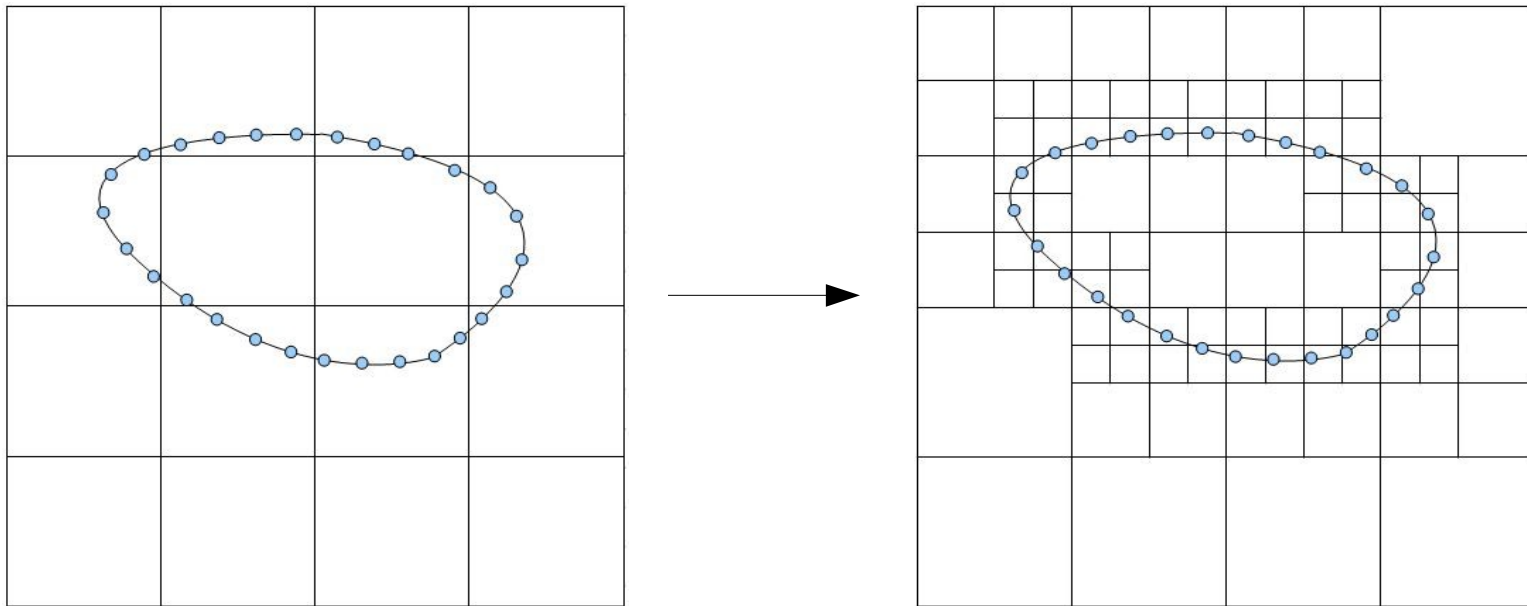
Diskretisierung

- Darstellung des Raums durch adaptiven Octree mit maximaler Tiefe D
- Verfeinerung bis jeder Punkt s in einem Knoten der Tiefe D ist



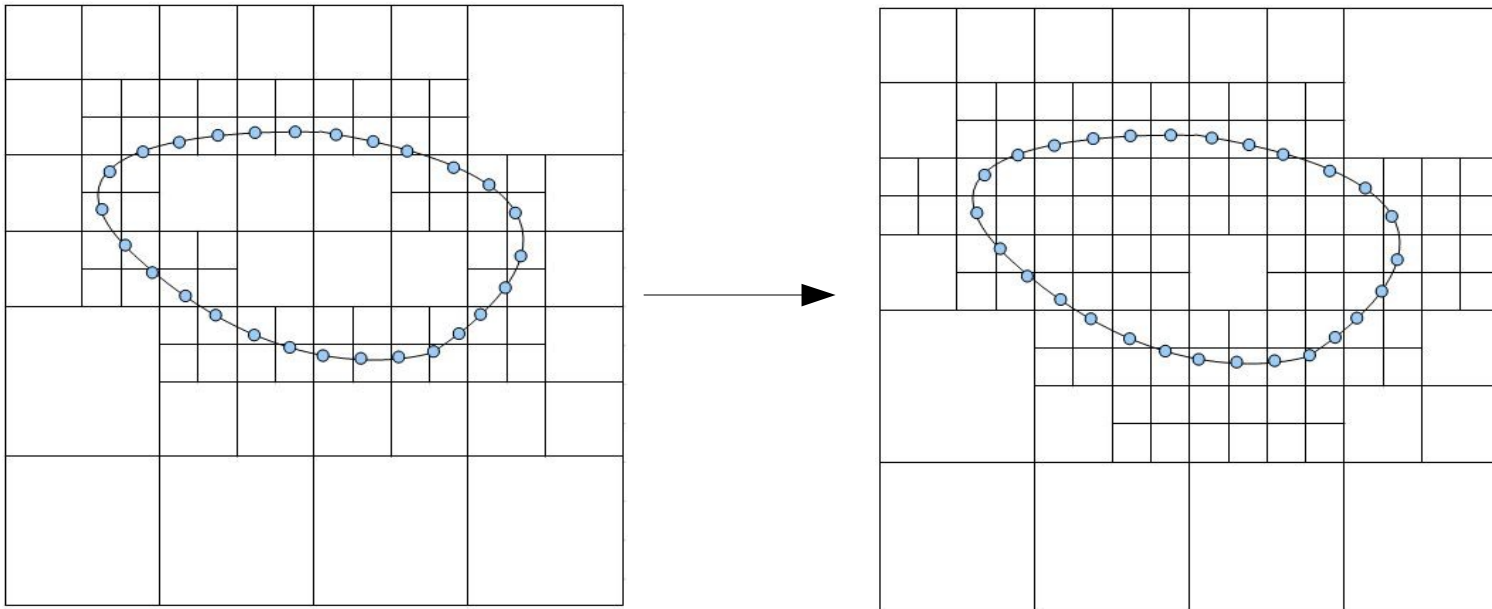
Diskretisierung

- Darstellung des Raums durch adaptiven Octree mit maximaler Tiefe D
- Verfeinerung bis jeder Punkt s in einem Knoten der Tiefe D ist



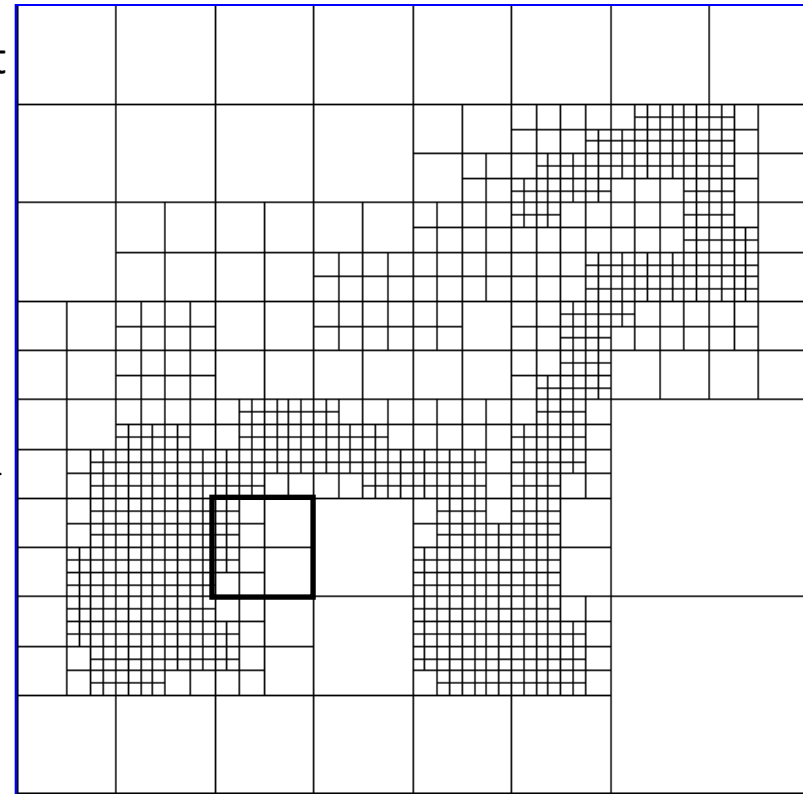
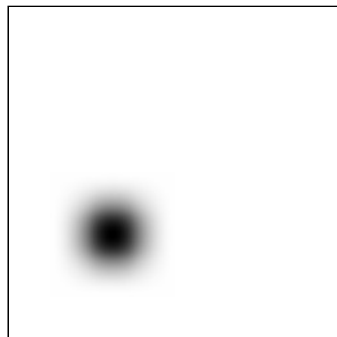
Diskretisierung

- trilineare Interpolation über die benachbarten Knoten für Sub-Knoten Präzision



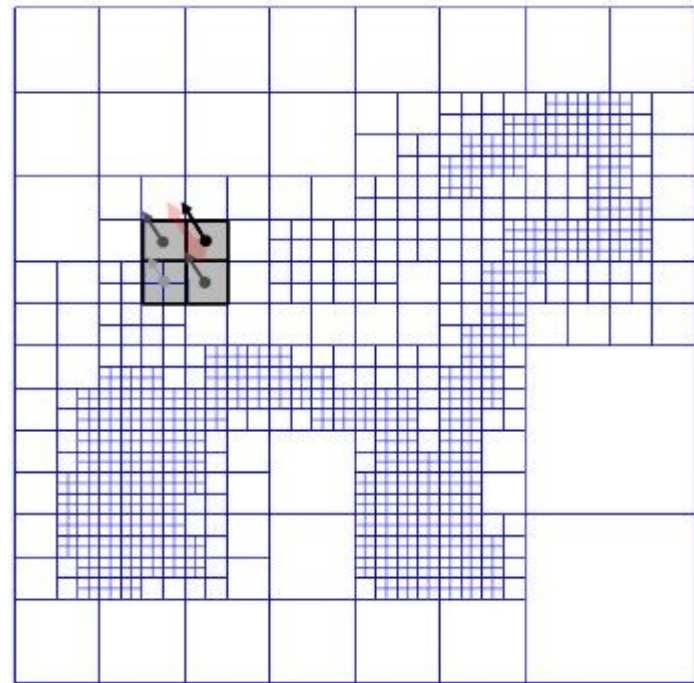
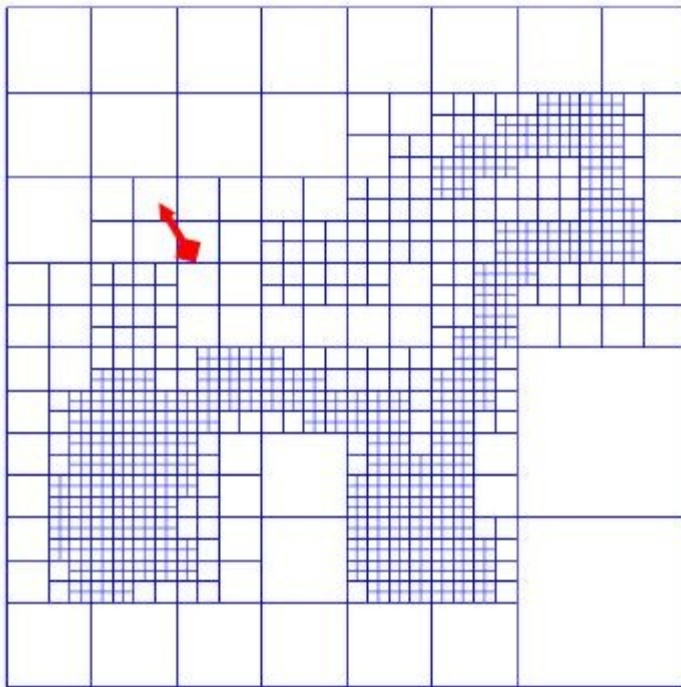
Bestimmung des Vektorfeldes

- Da die Indikatorfunktion Sprünge enthält muss sie gefiltert werden, damit der Gradient endlich ist
 - Die Filterung der Indikatorfunktion entspricht der Filterung der einzelnen Punkte
- die Filterfunktion ähnelt einer Gaussverteilung



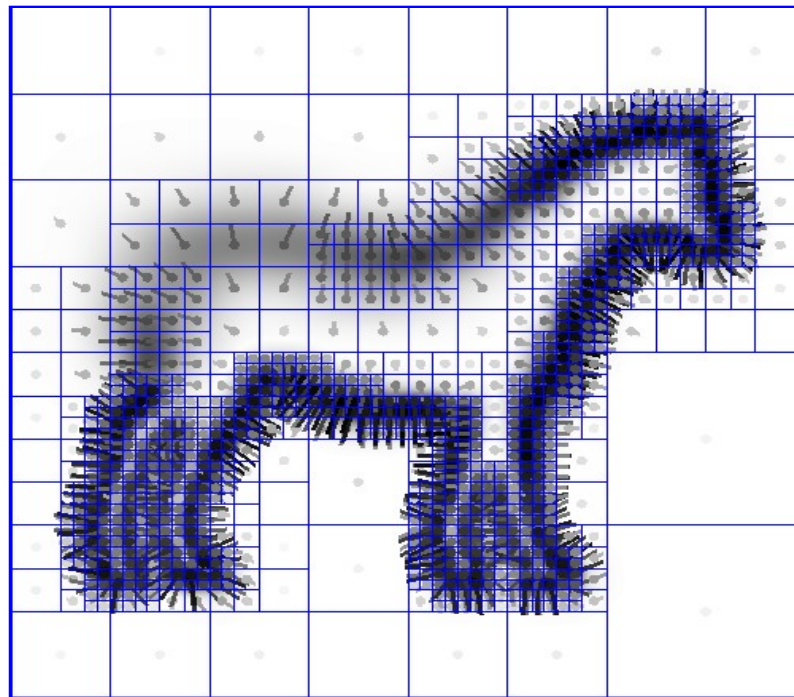
Erweiterung auf nicht gleichverteilte Punkte

- Abschätzen der Punktdichte, durch Verteilen der Punkte auf benachbarte Baumknoten
- Vergrößerung des Octrees in Abhängigkeit von der Punktdichte



Erweiterung auf nicht gleichverteilte Punkte

- mit Hilfe der geschätzten Punktdichte wird das Vektorfeld angepasst
- Anpassung des Filters an die lokale Baumtiefe



Lösung des Poissonproblems

- Nachdem \vec{V} definiert, soll χ so gelöst werden, dass gilt:
 $\nabla \chi = \vec{V}$ bzw. $\Delta \chi = \nabla \cdot \vec{V}$

- Funktionen bilden i.A. keine Orthonormalbasis

→ direkte Lösung teuer

- Vereinfachung möglich :

$$\sum_{o \in O} \|\langle \Delta \tilde{\chi} - \nabla \cdot \vec{V}, F_o \rangle\|^2$$



Lösung des Poissonproblems

$$\sum_{o \in O} \|\langle \Delta \tilde{x} - \nabla \cdot \vec{V}, F_o \rangle\|^2 = \sum_{o \in O} \|\langle \Delta \tilde{x}, F_o \rangle - \langle \nabla \cdot \vec{V}, F_o \rangle\|^2$$

ersetze: $v_o = \langle \nabla \cdot \vec{V}, F_o \rangle$

sei $\tilde{x} = \sum_o x_o F_o$ und L eine $|O| \times |O|$ Matrix

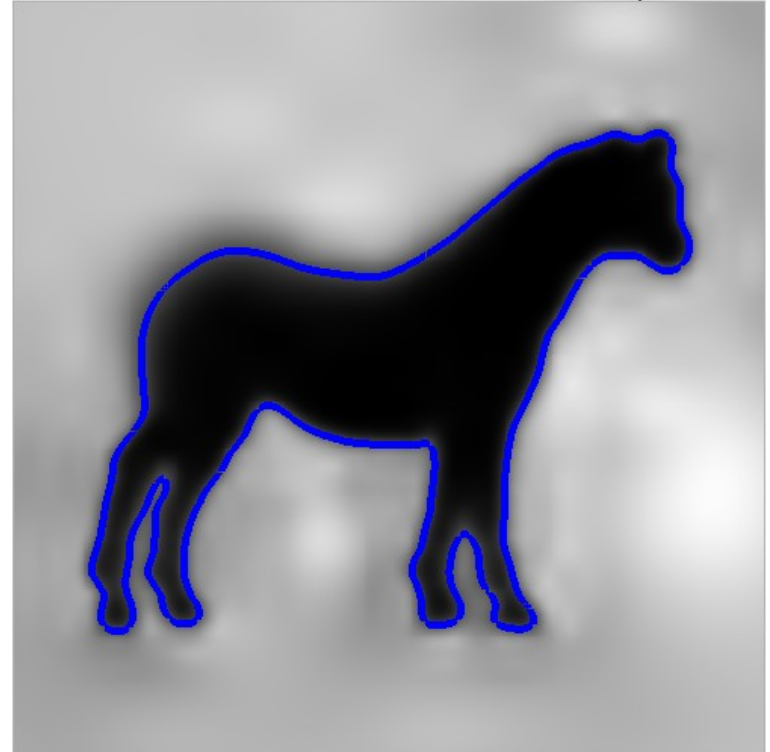
mit $L_{o,o'} \equiv \langle \frac{\delta^2 F_o}{\delta x^2}, F_{o'} \rangle + \langle \frac{\delta^2 F_o}{\delta y^2}, F_{o'} \rangle + \langle \frac{\delta^2 F_o}{\delta z^2}, F_{o'} \rangle$

dann gilt $\min_{x \in \mathbb{R}^O} \|Lx - v\|^2$

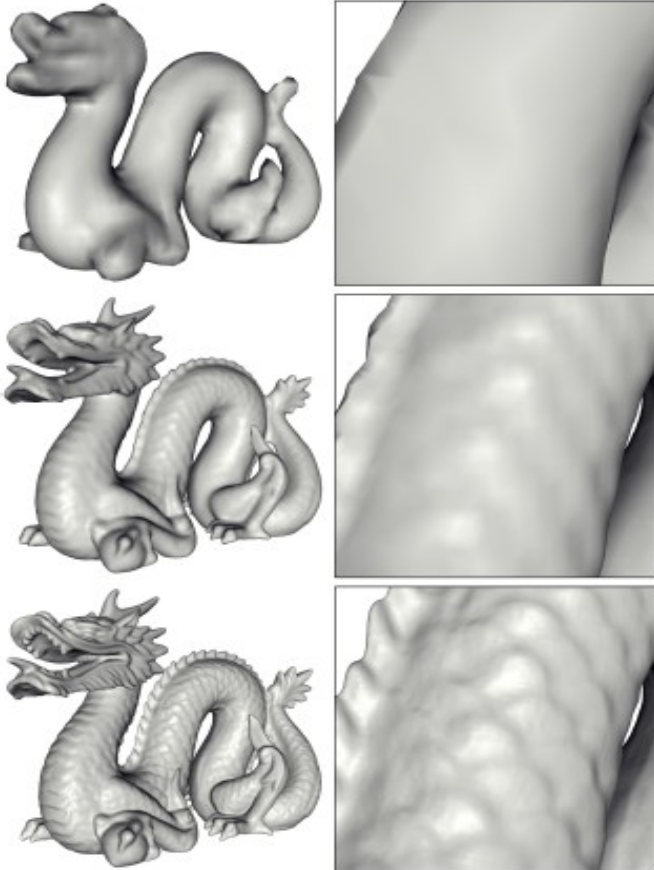
- Lösung mit Gauss-Seidel-Verfahren

Extraktion der Oberfläche

- Test aller Blätter im Octree, anhand der Indikatorfunktion, ob die Oberfläche durch das Blatt geht
- Überführung der Oberflächenfunktion durch den marching cubes Algorithmus



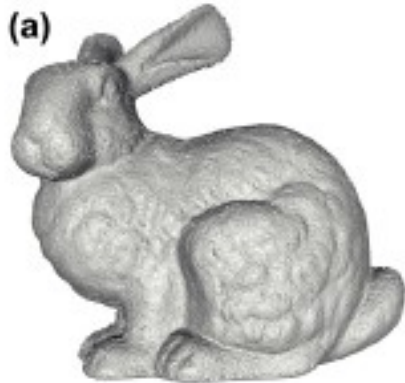
Auflösung und Skalierung mit Baumtiefe



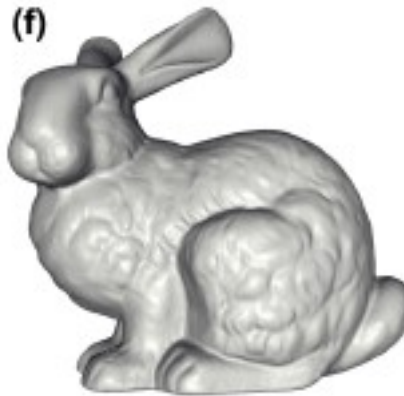
Baumtiefe	Zeit	max. Speicher- verbrauch	Anzahl Dreiecke
7	6	19	21 000
8	26	75	90 244
9	126	155	374,868
10	633	699	1516806

Im Bild verschiedene Baumtiefen
oben 6, in der Mitte 8 unten 10

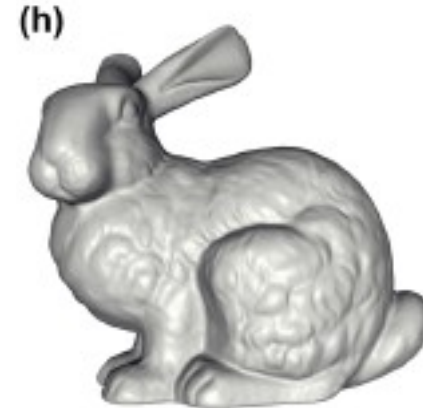
Vergleich mit anderen Verfahren



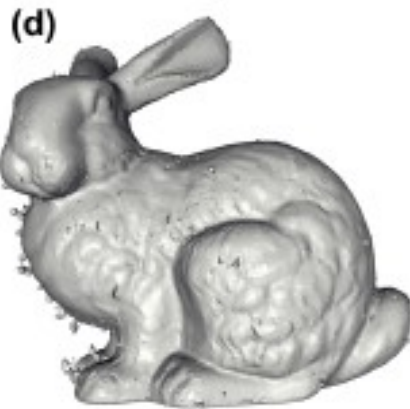
a) Power Crust



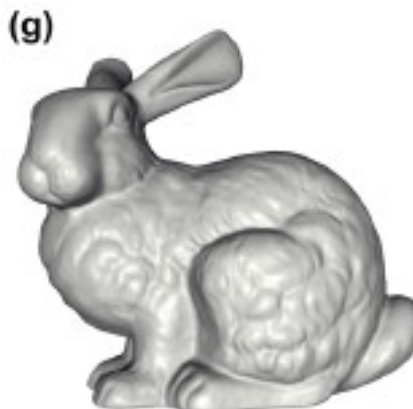
f) VRIP



h) Poisson Reconstruction



d) MPU



g) FFT-based Reconstruction

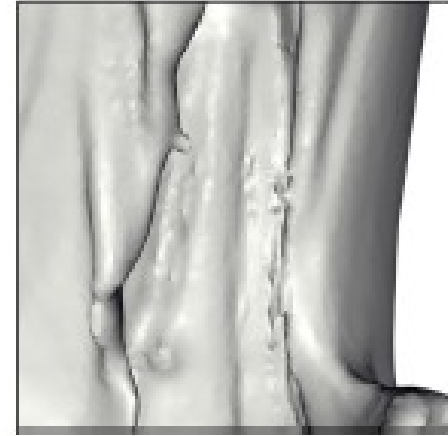
Vergleich mit anderen Verfahren VRIP



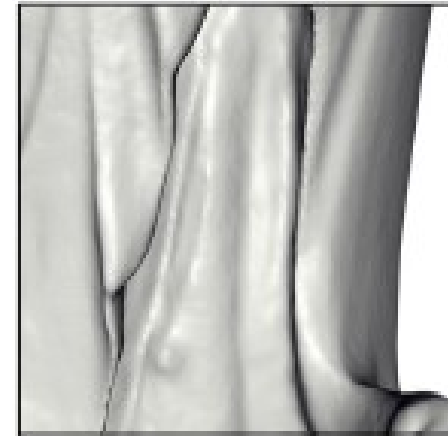
VRIP



Poisson

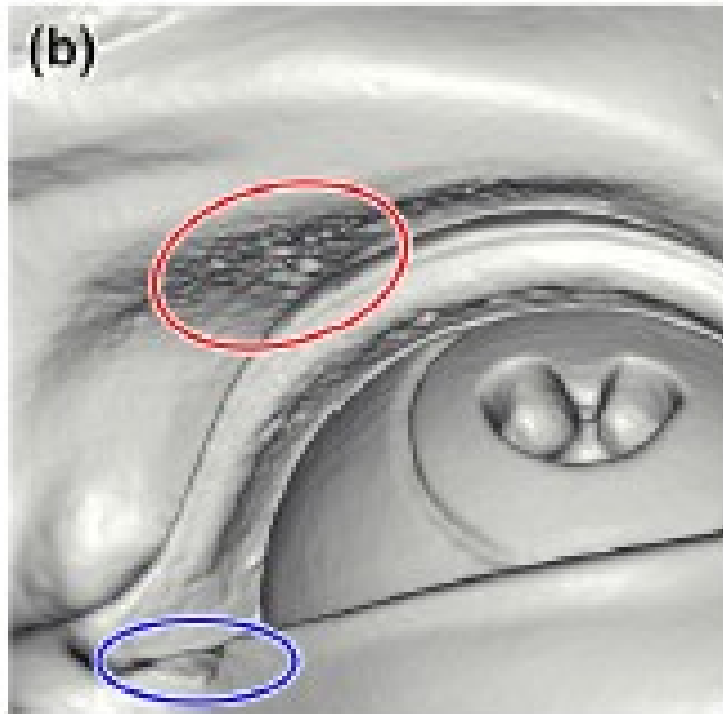


VRIP

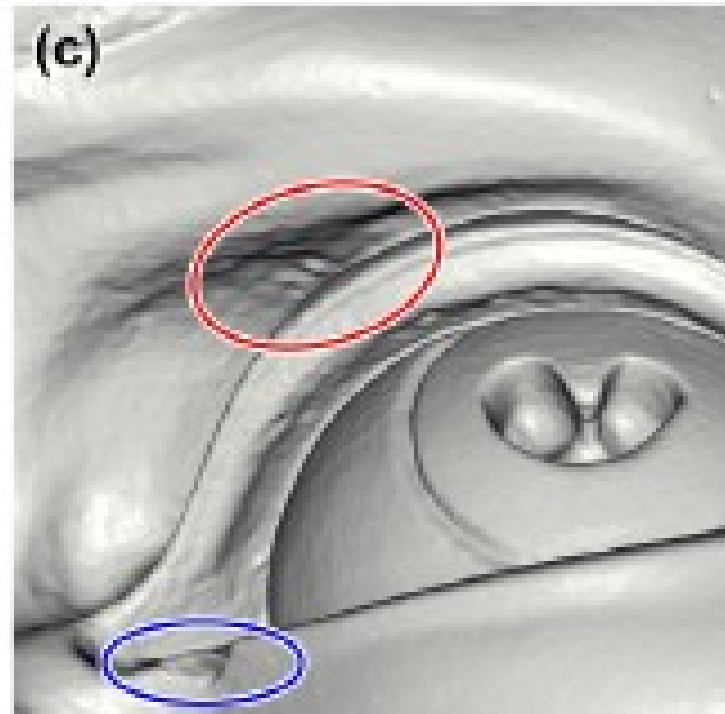


Poisson

Vergleich mit anderen Verfahren: FFT-basiert



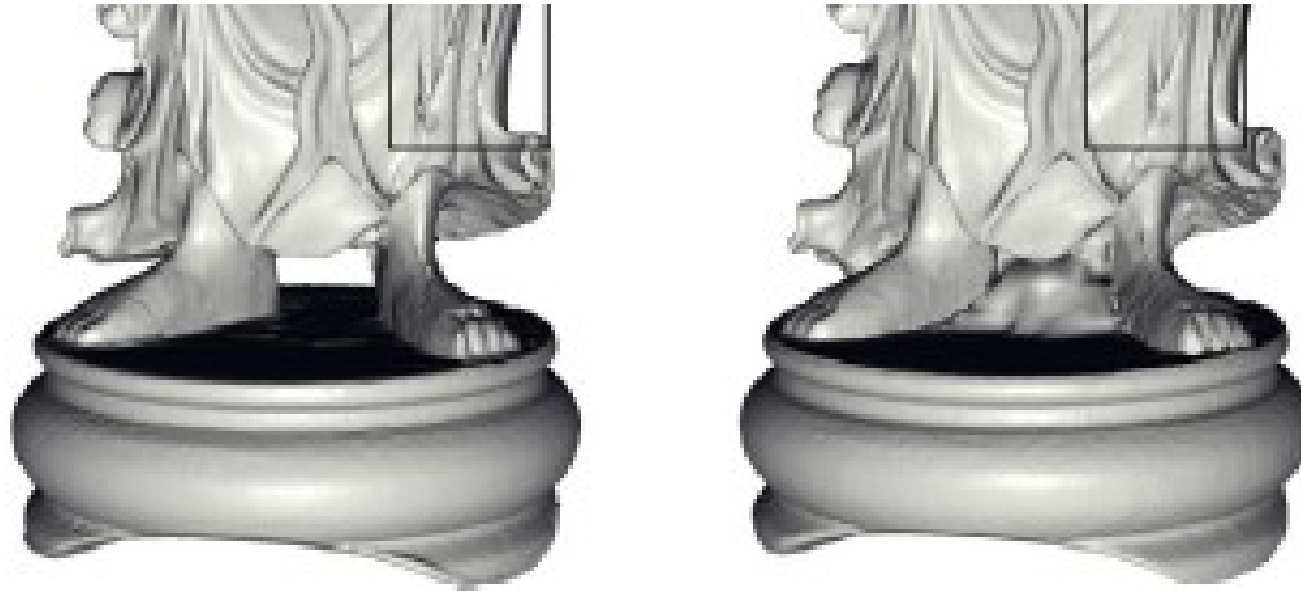
FFT-basiert



Poisson

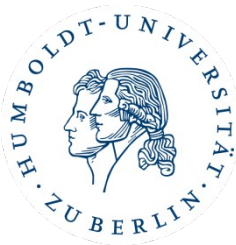
- FFT-basiertes Verfahren ist etwas anfälliger gegen Rauschen

Grenzen des Rekonstruktionsverfahrens



- fehlende Abtastpunkte an Oberflächen kann dazu führen, dass vorhandene Löcher, als zusammenhängendes Objekt erkannt werden

**Vielen Dank
für die Aufmerksamkeit!**



Quellen

- Poisson Surface Reconstruction
 - Paper: <http://www.cs.jhu.edu/~misha/MyPapers/SGP06.pdf>
 - Slides: <http://www.cs.jhu.edu/~misha/MyPapers/SGP06.ppt>

