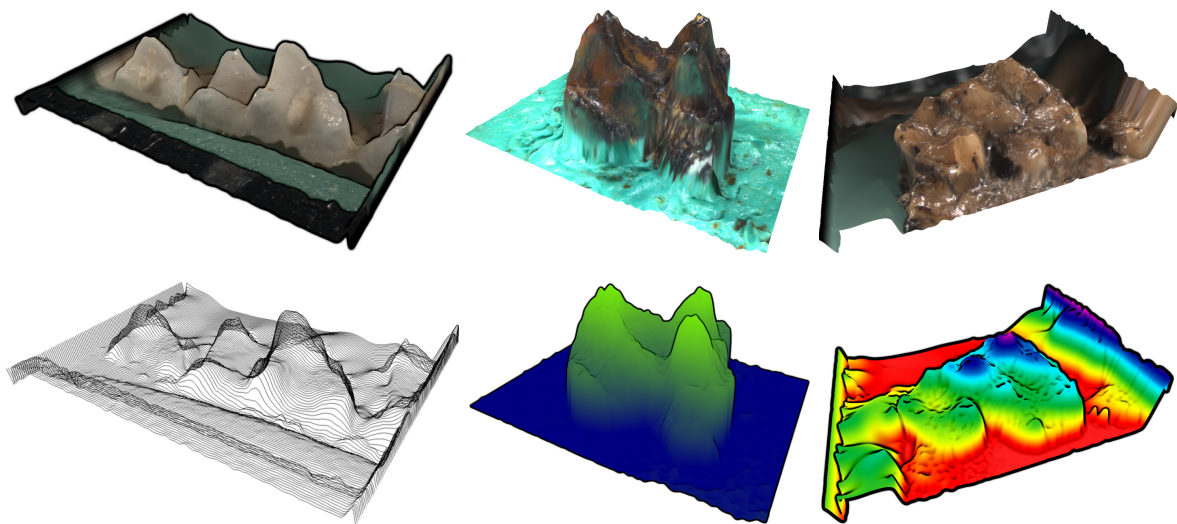


3D-Oberflächen-Rekonstruktion und plastisches Rendern aus Bilderserien

Sebastian Schäfer, Carsten Heep, Detlef Krömker

Institut für Informatik
Goethe-Universität Frankfurt am Main



Zusammenfassung Dieses Paper befasst sich mit der *3D-Rekonstruktion* der Oberfläche von mikroskopisch kleinen Objekten. Dies geschieht in Anlehnung an das von [11] vorgestellte Verfahren, welches in verbesserter Form verwendet wird [17]. Zum Anderen ist es das Ziel, die rekonstruierte Oberfläche mit unterschiedlichen Rendertechniken plastisch zu visualisieren. Dazu zählen fotorealistische, schematische (Drahtgitter-, Linien- oder Punkt-Modelle) und technische (Non-Photorealistic Rendering [9]) Darstellungen. Als besondere Anforderung soll der gesamte Verarbeitungsprozess so weit wie möglich auf die GPU ausgelagert werden.

Als Eingabe dient eine Bilderserie eines abmikroskopierten Objektes. Jedes Bild wird mit gleichen Kameraparametern und unterschiedlichen Objekt-Objekt-Abständen aufgenommen. Daher liegen in jedem Bild der Serie unterschiedliche Bereiche des Objektes im Fokus. Durch die Zuordnung von Höheninformationen zu den fokussierten Pixeln lässt sich ein konvexes Oberflächenmodell rekonstruieren auf dessen Grundlage sich weitere Verfahren zur Hervorhebung von Merkmalen anwenden lassen.

1 Einleitung

In verschiedenen naturwissenschaftlichen Disziplinen wie z.B. der Biologie oder auch der Paläoanthropologie werden kleinste Objekte betrachtet und analysiert. Diese Objekte sind häufig so klein, dass konventionelle Verfahren wie konfokale Laser- oder Streifenlicht Scanner zur Rekonstruktion eines 3D-Modells der Oberfläche nicht angewendet werden können. Eine mögliche Lösung ist der Einsatz von Computertomographen, die jedoch sowohl aus zeitlichen als auch aus finanziellen Gründen selten eine praxistaugliche Alternative bilden. Ein weiterer Nachteil der genannten Methoden ist, dass nur eine reine Objektrekonstruktion durchgeführt werden kann - für eine fotorealistische dreidimensionale Darstellung jedoch noch eine Oberflächentextur benötigt wird.

In dieser Arbeit wird ein Verfahren beschrieben, dass beide Probleme für beliebige konvex geformte Objekte löst. Als Eingabe dient eine Bilderserie, die von einem CCD-Sensor mit Hilfe eines Mikroskops aufgenommen wurde. Durch Verändern des Abstandes zwischen Objekt und Objektiv gelingt es einen *“kleinen”* Ausschnitt des Objektes zu fokussieren, so dass man eine n -elementigen Bilderserie \mathcal{I} mit relative Abstand Δ_i zwischen zwei aufeinanderfolgenden Bildern erhält. Durch *erkennen*, *extrahieren* und *komponieren* dieser fokalen Areale lässt sich ein Höhenbild des Objektes berechnen aus dem sich die Oberfläche rekonstruieren lässt. Neben dem plastischen Eindruck, den eine dreidimensionale Rekonstruktion mit sich bringt, ist ein weiterer Aspekt der Arbeit die Erstellung eines *Bildes gesamter Schärfe*, im Folgenden als *Total-Focus-Image* oder *TFI* bezeichnet. Dieses Bild kann als Textur auf die Oberfläche gelegt und so ein fotorealistisches 3D-Modell gerendert werden, welches als Ausgangspunkt für weitere Visualisierungsmethoden dient.

Der Aufbau dieser Arbeit gestaltet sich wie folgt: In Abschnitt 2 geben wir einen Überblick über vorangegangene Arbeiten, die sich mit der gleichen Thematik befassen haben. Der nachfolgende Abschnitt 3 stellt unser Verfahren kurz vor, so dass im Abschnitt 4 dessen Details behandelt werden. Der 5'te Abschnitt befasst sich mit der quantitativen Bewertung der Ergebnisse. Der letzte Abschnitt 6 geht auf bestehende Probleme und mögliche Lösungen ein.

2 Vorherige Arbeit

Der grundlegende Algorithmus, der Oberflächen aus Mikroskopaufnahmen erzeugt, wurde in [11] vorgestellt. Der Schwerpunkt der Arbeit liegt in der Durchführbarkeit des Verfahrens. Der vorgestellte Algorithmus wurde für diese Arbeit übernommen, angepasst und auf die GPU ausgelagert, wie dies schon in [17] angedeutet wurde. Das Verfahren von Nayar ([11]) bietet noch keine Möglichkeit, die Energie von Kanten als Maskierung zu benutzen und befasst sich daher auch nicht mit dem Problem des Lückenfüllens. Diese Aufgabe wurde in mehreren Arbeiten angegangen: Zum Einen kann die Unschärfe-Informationen der Bilder ausgenutzt werden ([12]) oder ein geschlossenes Surface mittels polynomieller Interpolation zur Rekonstruktion ([2]) benutzt werden. Ein alternatives Verfahren, das aus den gleichen Eingabedaten die 3D-Oberfläche rekonstruiert, wurde in [18] vorgestellt und benutzt ausschließlich die Unschärfe-Informationen der Bilder.

Es existieren verschiedene Ansätze Bilder auf der GPU zu filtern. In [13] werden konkrete Pixel-Shader für verschiedene Faltungsoperationen vorgestellt. In [5] wird ein einfaches Framework gezeigt, das Bildfilter auf der GPU objektorientiert umsetzt.

Unter dem Begriff *Non-Photorealistic Rendering* versteht man das Rendern von Bildern, die keinen fotorealistischen Eindruck erzeugen sollen. Häufig werden dabei verschiedene Zeichenstile wie z.B. technische Skizzen ([6]) oder künstlerische Malstile ([10], [15]) imitiert. Das Verfahren *Image enhancement by unsharp masking the depth buffer* [8] benutzt eine Kontrastverstärkung an Kontur- und Silhouettenkanten um Bilder plastischer erscheinen zu lassen.

3 Übersicht

Das vorgestellte Verfahren orientiert sich an der Methode *Shape from Focus* [11]. Innerhalb einer Bilderserie wird versucht, die fokussierten Bereiche jedes Bildes zu ermitteln, die zusammengesetzt ein komplett scharfes Bild ergeben. Mit den Höheninformationen, die jedem Bild zugeordnet sind, lässt sich so eine *Height-Map* (*HM*) bestimmen, mit deren Hilfe sich ein 3D-Modell erzeugen lässt. Die Height-Map wird in einem Zwischenschritt aus der *Energy-Height-Map* (*EHM*) generiert. In dieser ist für jeden Pixel die maximale Energie und die zugeordnete Höhe über alle Bilder der Bilderserie hinweg vermerkt. Sie kann daher benutzt werden, um schlecht erkannte Bereiche des Bildes zu maskieren.

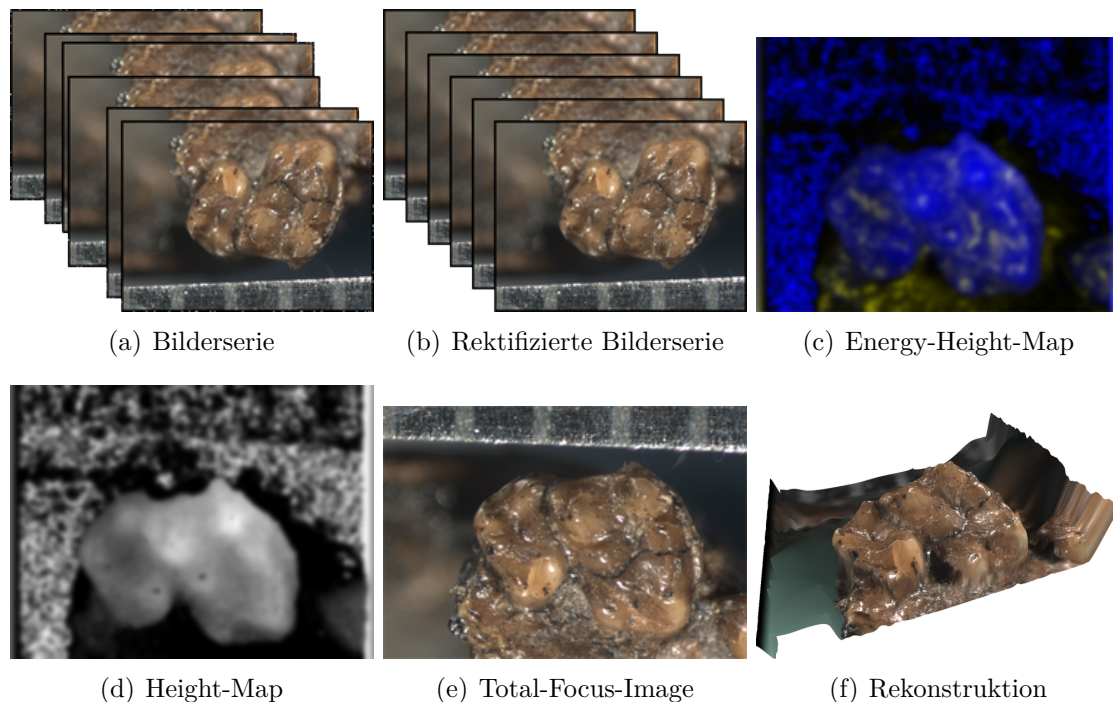


Abbildung 1: Die Bilderserie aus (a) wird im ersten Schritt *rektifiziert*. Im 2. Schritt wird aus den rektifizierten Bildern (b) Die EHM (c) berechnet. Diese wird zur HM (d) weiterverarbeitet und dient als Grundlage für die Erstellung des TFI (e), sowie der 3D-Rekonstruktion (f).

4 Verfahren

4.1 GPU-basierte Bildfilter

Da lokale Bildfilter auf den parallelen SIMD-Vektor-Prozessoreinheiten der GPU sehr schnell ausgeführt werden können, sind alle Schritte so gelöst, dass sie komplett auf der GPU verarbeitet werden. Dies geschieht in Anlehnung an die in [13] vorgestellte Methodik: Ein Pixel-Shader Programm wird für jeden Pixel des zu filternden Bildes ausgeführt und bestimmt so die neue Farbe. Da alle Schritte auf der GPU berechnet werden, hat dies den Vorteil, dass nur die Ausgangsbilder in den Grafikkartenspeicher geladen werden müssen. Alle Operationen darauf und alle Zwischenergebnisse können mittels *Render-To-Texture* realisiert werden. Da der Grafikspeicher sehr schnell an die GPU angebunden ist, kann einer der Flaschenhälse eines CPU-basierten Verfahrens komplett umgangen werden.

4.2 Rektifizierung der Bilderserie

Unser Verfahren verarbeitet die Bilder \mathcal{I}_i einer n -elementigen Bilderserie \mathcal{I} , die mit gleichen Kameraparametern aber unterschiedlichem Abstand zwischen Objekt und Objektiv aufgenommen wurde. Neben dieser Bilderserie wird der relative Betrachtungsabstand Δ_i zwischen zwei Bildern zur Rekonstruktion der Höhe benötigt. Typisch für diese Art von Aufnahmen sind zwei Aspekte: zum Einen liegen in jedem Bild unterschiedliche Bereiche im Fokus. Diese Eigenschaft bildet die Grundlage für die Oberflächenrekonstruktion nach der Methode *Shape from Focus*[11]. Die andere Eigenschaft ist die Änderung des „*Gesichtsfeldes*“. Dies kommt darin zum Tragen, dass das abmikroskopierte Objekt umso größer abgebildet wird, je näher sich dieses am Objektiv des Mikroskops befindet. Dieser Umstand macht es notwendig, dass im ersten Schritt eine Skalierung durchgeführt werden muss. Der Faktor der Skalierung s_i , mit $0 \leq i < n$, zur Beseitigung des ersten Fehlers, errechnet sich aus den Abstandsinformationen Δ_i zwischen den einzelnen Aufnahmen nach folgender Formel:

$$s_i = \frac{|s'_{n-1} - s'_0|}{|s'_{n-1} - s'_i|}$$

Hierbei sind die Höhenwerte der j -ten Aufnahme durch $s'_j := \sum_{i=0}^j \Delta_i$ gegeben.

Ein systematischer Fehler, der nicht zwingend für diese Art der Aufnahmen ist, aber dennoch häufig auftrat, war ein scheinbares „*wandern*“ der Objekte innerhalb der Bilderserie. Dies begründet sich darin, dass die Bewegungsachse des Mikroskops nicht mit dem Normalenvektor der Oberfläche übereinstimmte, auf der das Objekt während der Aufnahme lag. Daher beinhaltet die Rektifizierung neben der reinen Skalierung auch eine Registrierung. Dies geschieht durch Minimierung der Differenzbildung je zwei benachbarter Bilder \mathcal{I}_i und \mathcal{I}_{i+1} mit einer Offsetverschiebung in horizontaler und/oder vertikaler Ausrichtung. Wie in [7] beschrieben, kann dies mit der *Methode der kleinsten Quadrate* durchgeführt werden.

Die Skalierungs- und Offsetwerte werden nicht direkt auf die Bilderserie angewandt, sondern zwischengespeichert und beim Verarbeiten der Bilder mit der Texturkoordinate verrechnet: Jedes Bild \mathcal{I}_i wird direkt passend verschoben und skaliert ausgelesen.

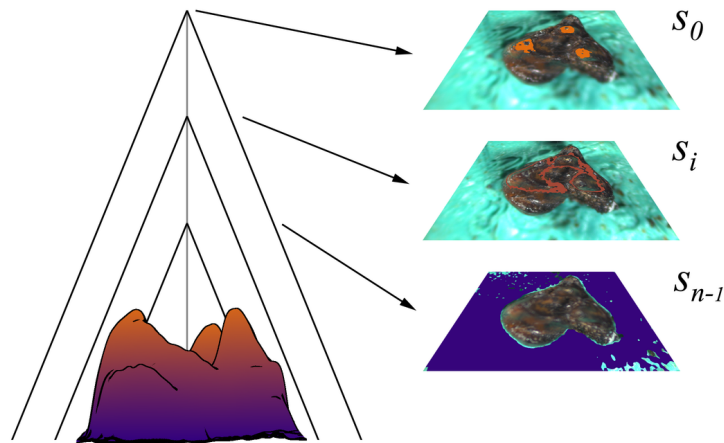


Abbildung 2: Schematisch Darstellung des Aufnahmeprinzips. Auf der linken Seite das zu rekonstruierende Objekt. Auf der rechten Seite sind exemplarisch drei Aufnahmen mit den erkannten fokalen Bereichen markiert dargestellt.

4.3 Berechnung der Energy-Height-Map

In der EHM (Abbildung 1(c)) wird für jeden Pixel der maximale Gradient (= Energie) aus der gesamten Bilderserie und sein zugeordneter Höhenwert in den Farbkanälen des Bildes vermerkt. Die Grundidee ist, dass die Energie eines Pixels mit dessen *Schärfe* korrespondiert: In einem scharfen Bildbereich sind die gesehenen Kanten stärker als in einem unscharfen. Die Erstellung der EHM erfolgt iterativ. Jedes Bild wird mit einem Laplace-Operator gefaltet und das Ergebnis mit der aktuellen EHM verglichen. Ist der Gradient des Bildes größer, so wird der Energie- und Höhenwert in der EHM mit den neuen Werten aktualisiert. In der Praxis konnten bessere Ergebnisse erzielt werden, wenn die Ausgabe des Laplace-Operators zusätzlich mit einem Tiefpass geglättet wurde.

4.4 Erzeugung der Height-Map

Ein inhärentes Problem des Verfahrens zur Berechnung der EHM ist, dass nicht mit absoluter Sicherheit alle fokussierten Bereiche immer exakt bestimmt werden. So ist die Energie nur in den Bildbereichen groß, in denen eine Kante vorkommt. Befindet sich z.B. ein Bildbereich im Fokus, der keine Kanten aufweist so ist die auftretende Energie sehr niedrig. Diese Bereiche beinhalten keine oder falsche Informationen in Bezug auf die Höhe der Oberfläche an einem Pixel beinhaltet.

Wir schlagen daher vor, diese Lücken mittels eines Schwellwertes (Thresholds) zu maskieren und in einem anschließenden Schritt zu füllen. Der Threshold definiert die minimal notwendige Energie, die eine Kante haben muss, um bei der Erstellung der HM berücksichtigt zu werden. Liegt die maximale Energie eines Pixels unterhalb des Schwellwertes, wird das Pixel maskiert und sein Höhenwert muss aus der Umgebung errechnet werden.

Die durch die Maskierung entstandenen Lücken werden mittels iterativer Interpolation geschlossen. Dazu wird ein selektiver Gauß-Operator auf den maskierten *Randpixel* angewandt, die in ihrer direkten Umgebung unmaskierte Pixel besitzen. Ein so interpoliertes Pixel gilt anschließend selbst als unmaskiert. Der Filter wird so oft angewandt, bis alle

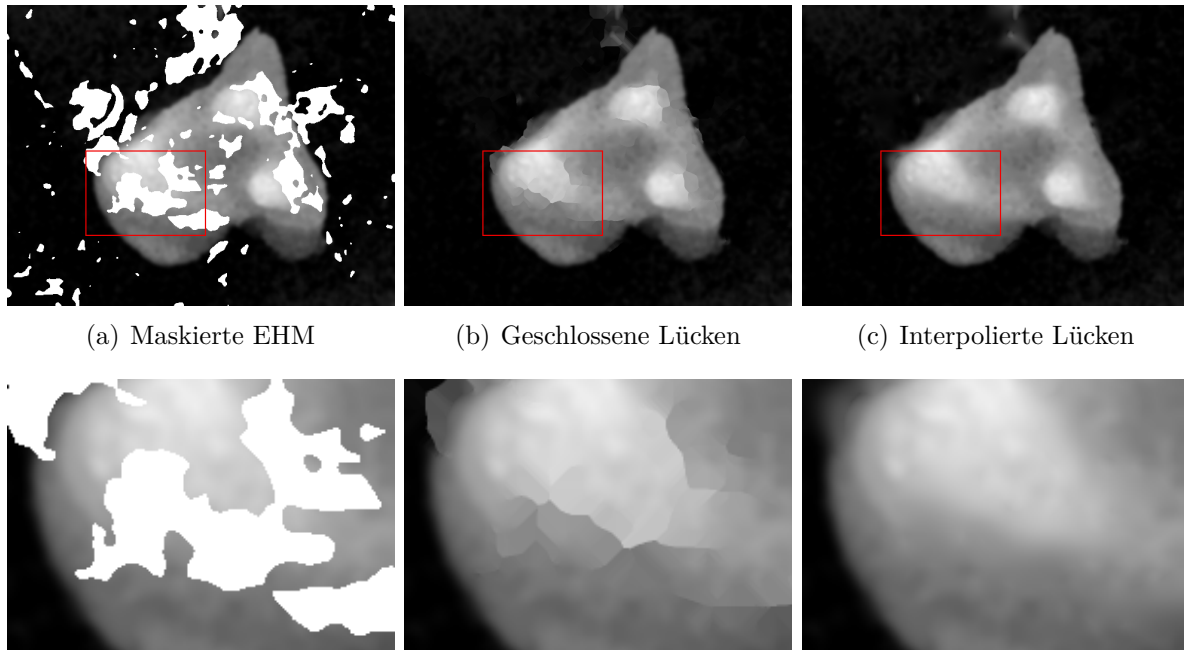


Abbildung 3: Bilder des Glättungsprozesses: oben die komplette EHM, unten ein Ausschnitt.

Lücken geschlossen sind. Die nun gefüllten Lücken beinhalten allerdings z.T. noch sehr harte Übergänge (Abbildung 3). Die iterativ vom Lückenmittelpunkt nach außen geglättet werden.

Das Ergebnis der Glättung ist eine HM (Abbildung 1(d)), die in jedem Pixel die entsprechende Höhe der rekonstruierten 3D-Oberfläche gespeichert hat. Es ist auch möglich, eine Maske zu laden und so komplette Teile der Bilderserie von der Rekonstruktion auszuschließen.

4.5 Berechnung des Total-Focus-Image

Aus der Height-Map und der Bilderserie lässt sich iterativ das *Total-Focus-Image* berechnen (Abbildung 1(e)). Dieser Vorgang lässt sich als Abbildung auffassen, da die Höhenwerte der Height-Map in direkter Relation zu den scharfen Bildbereichen der Bilderserie \mathcal{I} stehen, aus denen sich das Total-Focus-Image kumulativ zusammensetzt.

4.6 Rendern der 3D-Oberfläche

Es wird ein polygonales Modell aus der Height-Map rekonstruierten. Jedem Pixel ist ein Vertex zugeordnet, dessen Position durch die Heightmap bestimmt ist. Das Modell kann nun für verschiedene Darstellungsarten unterschiedlich gerendert werden. Der einfachste Fall ist eine fotorealistische Darstellung, in der das Total-Focus-Image als Textur auf das Modell gelegt wird (Abbildungen 4(a), 5(a), 6(a), 7(a)).

Um das Modell schematisch anzuzeigen, genügt es, verschiedene Füllmodi des Rasterisierers auszuwählen. So kann ein Drahtgitter-, Punkt- oder auch Linienmodell angezeigt werden (Abbildungen 5(c), 7(b)).

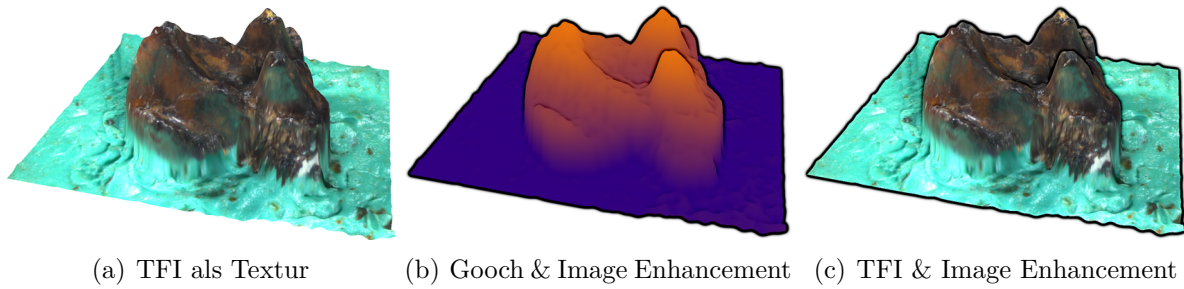


Abbildung 4: Rekonstruktion des Molaren M/2 von *Leptictidium nasutum*.

4.6.1 Non-Photorealistic Rendering

Die rekonstruierte 3D-Oberfläche kann auch in der Art einer technischen Zeichnung gerendert werden. Diese Darstellung erleichtert es unter anderem die Morphologie der Oberfläche besser zu erkennen. Innerhalb des *Non-Photorealistic Renderings* gibt es verschiedene etablierte, miteinander kombinierbare Techniken, von denen drei ausgewählte implementiert werden:

- Hervorheben von räumlichen Zusammenhängen
- Einfärbung der Oberfläche mit Falschfarben
- Darstellung der Modellkontur und Silhouette (*CEL*)

Das in [8] vorgestellte Verfahren *Image enhancement by unsharp masking the depth buffer (IE)* betont die räumlichen Zusammenhänge der Darstellung. Dazu wird neben einem Bild der Oberfläche das Tiefenbild der virtuellen 3D Kamera benötigt. Weist es Diskontinuitäten auf, so wird das gerenderte Bild abgedunkelt bzw. aufgehellt und der räumlichen Unterschied verdeutlicht (Abbildungen 4(b), 4(c), 5(c), 6(b), 6(c), 7(b)).

Neben einer fotorealistischen Oberfläche kann das Modell auch mit Falschfarben eingefärbt werden, wie es in [6] vorgestellt wird. Als Schlüssel für den Farbverlauf bietet sich hier die Höhe der Vertices an (Abbildung 4(b), 5(b), 5(c), 6(c), 7(b), 7(c)).

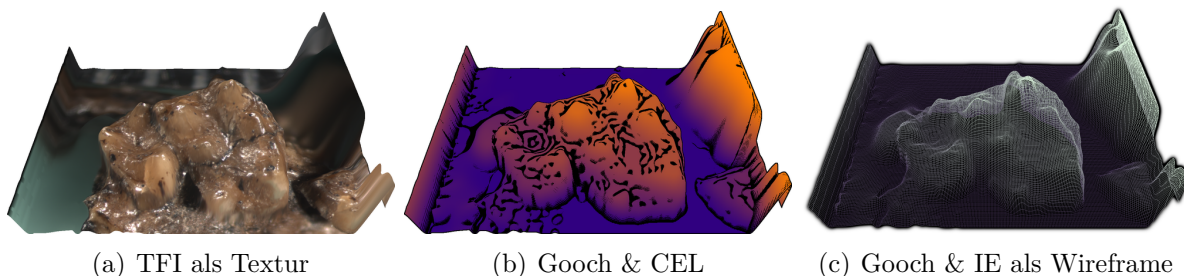


Abbildung 5: Rekonstruktion des Molaren 2-3 von *Eochenus*.

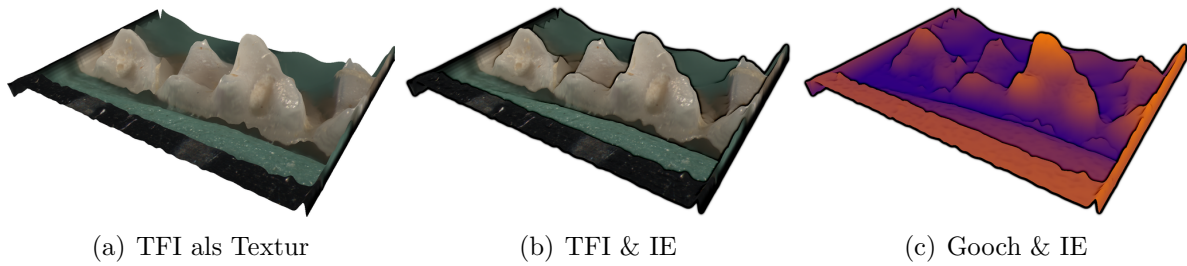


Abbildung 6: Rekonstruktion des Molaren m2 einer *Spitzmaus*.

Die Silhouette eines Modells kann nach [14] durch das Bestimmen einer Diskontinuität im Tiefenbild mit einem Sobel-Operator für jeden Pixel berechnet werden. Ist die Energie des Gradienten im Tiefenbild größer als ein festgelegter Schwellwert, so befindet sich an dieser Stelle im Bild eine Silhouettenkante. Konturen, die keine Silhouetten bilden, können nach [4] und [3] als Diskontinuität der Oberflächennormale aufgefasst werden. Auch hier findet der Sobel-Operator kombiniert mit einem Schwellwert Verwendung (Abbildungen 5(b), 7(c)).

Um das Tiefenbild der virtuellen Kamera und die Normalen zu rendern, werden *Multiple-Render-Targets (MRT)* eingesetzt, so dass das Mesh nur einmal gerendert werden muss, aber mehrere unterschiedliche Texturen erzeugt werden. In unserem Fall brauchen wir drei verschiedene Texturen:

1. Das gerenderte Modell mit der gewünschten Oberflächendarstellung (z.B. fotorealistisch oder Falschfarben),
2. Das Tiefenbild der virtuellen Kamera,
3. Eine farbliche Darstellung der Oberflächennormale für jeden Pixel, in der der XYZ-Wert der Normalen in die RGB-Kanäle geschrieben wird.

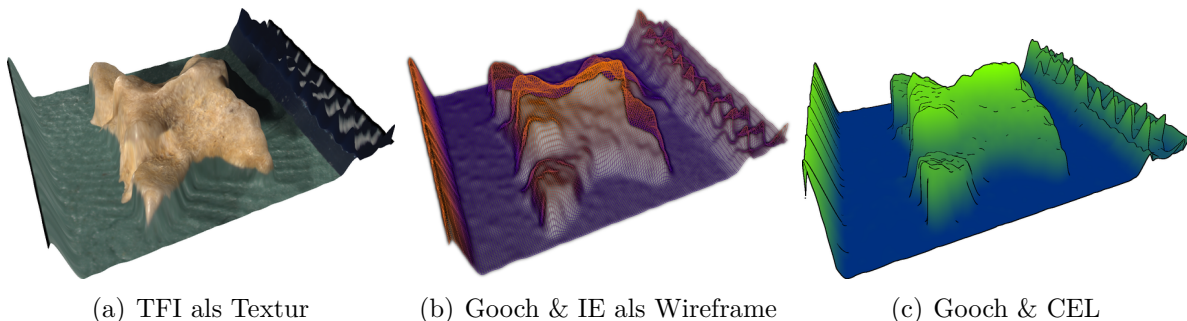


Abbildung 7: Rekonstruktion eines Wirbels (Vertebra) von *Ogmophis*.

5 Ergebnisse

5.1 Quantität

Das hier vorgestellte GPU-basierte Verfahren wird hinsichtlich der Verarbeitungsgeschwindigkeit mit einer rein auf der CPU arbeitenden Implementierung, die aus gleichen Filter aufgebaut ist, verglichen. Dazu wurden zwei Bilderreihen ausgewählt, die aus 30 und 59 Einzelbildern bestehen. Jede Reihe wurde in drei verschiedenen Auflösungen in die Programme geladen. Um die Skalierbarkeit der GPU zu vergleichen wurden zusätzlich zwei verschiedene NVidia GPU's eingesetzt: eine aktuellen NVidia GTX 260 und eine 3 Jahre alte 8800 GTX. Der Testrechner ist mit einer Intel i5 CPU mit 3,47GHz und 8GB Arbeitsspeicher ausgerüstet. Die Laufzeiten sind in Sekunden angegeben.

	2720x2048 px		1360x1024px		680x512 px	
	HM	TFI	HM	TFI	HM	TFI
CPU	373.123	12.605	90.855	3.073	22.628	0.756
8800 GTX	32.909	27.051	4.002	4.080	0.734	0.741
GTX 260	14.913	9.813	3.349	3.713	0.593	0.570

Abbildung 8: 30-er Bilderserie eines *Ogmophis* Vertebra.

	1360x1024px		680x512 px	
	HM	TFI	HM	TFI
CPU	176.047	5.897	43.672	1.599
8800 GTX	7.402	8.252	1.935	1.919
GTX 260	5.499	6.193	1.521	1.583

Abbildung 9: 59-er Bilderserie eines *Spitzmaus*-Gebißes.

5.2 Qualität

Leider war ein direkter Vergleich zur Bestimmung der Güte bislang nicht möglich, da die uns zur Verfügung stehende Vergleichsprobe des *Eochenus* aus einem anderen Winkel abmikroskopiert wurde (Abbildung 10). Da an dieser Probe keine Trackingpunkte angebracht sind, konnten beide Aufnahmen auch nicht direkt überlagert werden. Als subjektives Maß der Güte unseres Verfahrens, können wir uns daher nur auf die positive Resonanz unserer Kooperationspartner (Senckenberg Forschungsinstitut und Naturmuseum) beziehen.

6 Zusammenfassung

In diesem Paper haben wir ein auf [11] basierendes Verfahren zur Rekonstruktion konvexer Oberflächen mikroskopischer Objekte vorgestellt. Zur Beschleunigung unserer Implementierung wurde die Grafikkarte herangezogen. Wie in Abschnitt 5 gezeigt führt dies zu einem Geschwindigkeitszuwachs um einen Faktor ≥ 20 .

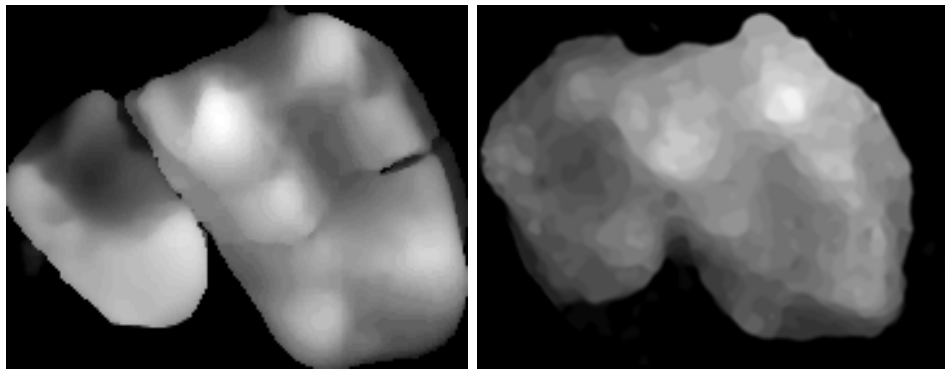


Abbildung 10: Vergleichsbilder des oberen rechten Molaren 2-3 eines *Eochenus*. Links: Segmentierte CT-Aufnahme. Rechts: Ergebnis der Rekonstruktion.

Trotz der guten Ergebnisse treten unter gewissen Umständen Probleme auf. Zu diesen zählen durch Spiegelung der Lichtquellen erzeugte Artefakte, die sich mit einer “besseren” Ausleuchtung reduzieren lassen. Da sie sich nicht komplett vermeiden lassen, müssen andere Verfahren in Betracht gezogen werden, um die per Maskierung Lücken zu erzeugten die in einem anschließenden Schritt zu schließen sind. Mögliche Ansätze sind in [1], [19] oder [16] zu finden.

Im Bereich des *Non-Photorealistic Rendering*, genauer gesagt im Kontext der Strukturzeichnung, lässt sich die Güte mit einem zweiphasigen Ansatz steigern. In der ersten Phase werden anatomisch relevante Punkte extrahiert. Diese dienen in der zweiten Phase als Eingabe für die Hervorhebung relevanter Objektareale.

Statt einer Height-Map kann das Verfahren auch angepasst werden, so dass von einem transluzenten Objekt ein Volumen rekonstruiert wird. Mögliche Anwendungsfälle wären die 3D-Darstellung von Zellen oder Bernsteineinschlüssen.

Danksagung

Unser Dank gilt Dr. Thomas Lehmann und Dr. Krister Smith vom Senckenberg Forschungsinstitut und Naturmuseum für die Bereitstellung des Bildmaterials und der Unterstützung während der Arbeit.

Literatur

- [1] BERTALMIO, MARCELO, GUILLERMO SAPIRO, VINCENT CASELLES und COLOMA BALLESTER: *Image inpainting*. In: *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, Seiten 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [2] CHOI, T. S., M. ASIF und JOUNGIL YUN: *Three-dimensional shape recovery from focused image surface*. In: *ICASSP '99: Proceedings of the Acoustics, Speech, and*

- Signal Processing, 1999. on 1999 IEEE International Conference*, Seiten 3269–3272, Washington, DC, USA, 1999. IEEE Computer Society.
- [3] DECAUDIN, PHILIPPE: *Cartoon Looking Rendering of 3D Scenes*. Research Report 2919, INRIA, Juni 1996.
 - [4] DECAUDIN, PHILIPPE: *Modélisation par Fusion de Formes 3D pour la Synthèse d’Images – Rendu de Scènes 3D imitant le Style ‘Dessin Animé’*. Thèse de doctorat, Université de Technologie de Compiègne, France, Dezember 1996.
 - [5] FERNANDO, RANDIMA: *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Pearson Higher Education, 2004.
 - [6] GOOCH, BRUCE und AMY A. GOOCH: *Non-Photorealistic Rendering*. A K Peters, Ltd., Natick, 2001.
 - [7] JÄHNE, BERND: *Digitale Bildverarbeitung*. Springer, Berlin, 6. Auflage, 2005.
 - [8] LUFT, THOMAS, CARSTEN COLDITZ und OLIVER DEUSSEN: *Image enhancement by unsharp masking the depth buffer*. ACM Transactions on Graphics, 25:1206–1213, 2006.
 - [9] MARKOSIAN, LEE, MICHAEL A. KOWALSKI, DANIEL GOLDSTEIN, SAMUEL J. TRYCHIN, JOHN F. HUGHES und LUBOMIR D. BOURDEV: *Real-time nonphoto-realistic rendering*. In: *SIGGRAPH ’97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, Seiten 415–420, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
 - [10] MARKOSIAN, LEE, BARBARA J. MEIER, MICHAEL A. KOWALSKI, LORING S. HOLDEN, J. D. NORTHRUP und JOHN F. HUGHES: *Art-based rendering with continuous levels of detail*. In: *NPAR ’00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, Seiten 59–66, New York, NY, USA, 2000. ACM.
 - [11] NAYAR, S. K. und Y. NAKAGAWA: *Shape from Focus*. IEEE Trans. Pattern Anal. Mach. Intell., 16(8):824–831, 1994.
 - [12] PRADEEP, K. S. und A. N. RAJAGOPALAN: *Improving Shape from Focus Using Defocus Information*. In: *ICPR ’06: Proceedings of the 18th International Conference on Pattern Recognition*, Seiten 731–734, Washington, DC, USA, 2006. IEEE Computer Society.
 - [13] ROST, RANDI J.: *OpenGL(R) Shading Language*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.
 - [14] SAITO, TAKAFUMI und TOKIICHIRO TAKAHASHI: *Comprehensible Rendering of 3-D Shapes*. Seiten 197–206, 1990.
 - [15] SALISBURY, MICHAEL P., SEAN E. ANDERSON, RONEN BARZEL und DAVID H. SALESIN: *Interactive pen-and-ink illustration*. In: *SIGGRAPH ’94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, Seiten 101–108, New York, NY, USA, 1994. ACM.

- [16] SANTIAGO SALAMANCA, PILAR MERCHÁN, EMILIANO PÉREZ ANTONIO ADÁN CARLOS CERRADA: *Filling Holes in 3D Meshes using Image Restoration Algorithms*. In: *The Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.
- [17] SEBASTIAN SCHÄEFER und FRANK NAGEL: *3D-SurReAL - 3D Surface Reconstruction of Arbitrary (Image) Layer*. In: *Visual Computing for Biology and Medicine 2010, Eurographics Workshop*. Poster, 2010.
- [18] SUBBARAO, MURALI und GOPAL SURYA: *Depth from defocus: a spatial domain approach*. *Int. J. Comput. Vision*, 13(3):271–294, 1994.
- [19] SUN, JIAN, LU YUAN, JIAYA JIA und HEUNG YEUNG SHUM: *Image completion with structure propagation*. In: *ACM Transactions on Graphics*, Seiten 861–868, 2005.