

Interaktive diffusionsbasierte Segmentierung von Volumendaten auf Grafikkarte

Caroline Langer¹, Markus Hadwiger¹, Katja Bühler¹

¹Medical Visualization, VRVis Research Center for Virtual Reality and Visualization,
1220 Wien

Email: {langer|msh|buehler}@vrvis.at

Zusammenfassung. Im vorliegenden Artikel wird ein verbesserter Ansatz für eine *interaktive* Segmentierung von Volumendaten auf Standard PCs vorgestellt. Sowohl Visualisierung als auch Segmentierung wurden durch Programmierung handelsüblicher 3D Grafikkarten so stark beschleunigt, dass eine Darstellung des Fortschrittes der Segmentierung in Echtzeit und im Kontext des gesamten Datensatzes möglich ist. Dadurch kann direkt korrigierend in den Ablauf der Segmentierung eingegriffen werden. Der implementierte Segmentierungsalgorithmus kombiniert Region Growing (Bereichswachstumsverfahren) mit diffusionsbasierter Kantenerkennung und läßt sich durch Parametereinstellungen auf eine Vielzahl von Problemstellungen anwenden.

1 Einleitung

Die Segmentierung medizinischer Daten ist in der Praxis auch heute noch in vielen Fällen ein mühsamer und zeitaufwändiger Prozess. Vollautomatische Segmentierungsalgorithmen liefern oft nicht die gewünschten Resultate, da die Komplexität der Aufgabe meistens hochspezialisierte Lösungen erfordert. Halbautomatische Segmentierungsalgorithmen vereinen Erfahrung und Wissen des Radiologen mit der Objektivität und Schnelligkeit einer computergenerierten Lösung. Die Interaktion des Experten mit dem Programm kann dabei grundsätzlich vor, nach oder auch während des automatischen Teils der Segmentierung stattfinden. Existierende halbautomatische Segmentierungsalgorithmen beschränken sich überwiegend auf die Interaktion vor (Initialisierung, Auswahl von Parametern) und nach (Korrekturen) der Ausführung des Algorithmus. Da sowohl Segmentierung als auch Visualisierung von Volumendaten rechen- und speicherintensiv sind, gibt es bis heute kaum Lösungen, die eine Visualisierung des Segmentierungsfortschrittes im Kontext der originalen Volumendaten anbieten. Fehlt eine solche Visualisierung, ist eine direkte Überwachung des Segmentierungsprozesses und ein Eingreifen in den Segmentierungsprozess kaum möglich.

Moderne 3D PC Grafikkarten ermöglichen heute die Implementierung leistungsfähiger Volumenvisualisierer, die eine schnelle und qualitativ hochwertige Darstellung der Daten zulassen [1]. Die neuesten Grafikkarten erlauben eine parallele Ausführung arithmetischer Basisoperationen und Grundfunktionen mit 32bit-Präzision und bieten damit die Funktionalität, auch Segmentierungsalgorithmen

zu parallelisieren und direkt auf der Grafikkarte zu berechnen [2].

Damit ist es erstmals tatsächlich möglich 3D Visualisierung, Segmentierung und Interaktion gleichzeitig auszuführen.

Einer der ersten Ansätze zur *interaktiven Segmentierung im Kontext des Gesamtvolumens* beruht auf deformierbaren impliziten Flächen (level sets), deren Implementierung ebenfalls mit Hilfe von Grafikkarten umgesetzt wurde [3].

Erst kürzlich wurde ein neuer Ansatz und seine Implementierung auf Grafikkarten für die interaktive Segmentierung dreidimensionaler medizinischer Daten vorgestellt [4], in dem Diffusionsprozesse nicht nur, wie in der klassischen Anwendung üblich, zur Bildglättung und Kantenverstärkung verwendet wird [5], sondern auch zur Segmentierung selbst. Der von Sherbondy et al. vorgestellte Algorithmus basiert auf klassischem Region Growing, in dem allerdings während des Segmentierungsprozesses die Ausbreitung der Segmentierungsmaske durch einen Diffusionsprozeß gesteuert wird. Getestet wurde der Algorithmus zur Segmentierung der Aorta, Aneurysmen und peripheren Blutgefäßen in der Lunge. Die im Rahmen dieses Artikels vorgestellte interaktive diffusionsbasierte Segmentierung ist eine Erweiterung des Ansatzes von Sherbondy mit weiteren Diffusionsmodellen, leicht veränderter Diskretisierung und Integration in einen leistungsfähigen Volumenvisualisierer [6]. Der entwickelte Prototyp gestattet es dem Benutzer, die Entwicklung der Segmentierung in zwei- und dreidimensionalen Ansichten in Echtzeit zu verfolgen und direkt in den Evolutionsprozess einzugreifen.

2 Methoden

Ablauf. Nach einer optionalen Bildglättung durch einen Diffusionsprozess beginnt der Anwender mit der Segmentierung, indem er Startbereiche (Entwicklungspunkte) einzeichnet. Die Ausbreitung der Entwicklungspunkte wird durch einen Diffusionsprozess gesteuert. Die Diffusion der Entwicklungspunkte wird von der Länge der Gradienten der Daten im Ursprungsbild beeinflusst und stoppt dadurch an Kanten. Nach jeder Iteration wird die Objektzugehörigkeit der Voxel im Ursprungsbild aktualisiert und das identifizierte Objekt im Kontext des Gesamtvolumens gezeichnet (Abb. 1). Da eine Ausbreitung der Bereiche durch die Diffusion nur in die unmittelbare Nachbarschaft erfolgt, kann man den Bereich, auf dem die Diffusion berechnet wird, entsprechend reduzieren.

Das identifizierte Objekt kann im Anschluss abgespeichert, mit anderen bereits abgespeicherten Objekten durch binäre Operationen verglichen und bei einer neuen Segmentierung im Kontext mit angezeigt werden. Im Gegensatz zu Sherbondy et al. [4] werden bereits vorher segmentierte Objekte und das Gesamtvolumen in unterschiedlichen Modi und mit verschiedenen Transferfunktionen während der Segmentierung dargestellt. Der Benutzer kann in die laufende Segmentierung eingreifen, indem er Parameter ändert, weitere Entwicklungspunkte einzeichnet oder Bereiche löscht und damit eine Diffusion in diese verhindert.

Diffusion. Für die Diffusion der Entwicklungspunkte wird eine Ausbreitungsmaske über das Gesamtvolumen gelegt, in welche die Entwicklungspunkte ein-



Abb. 1. Ablauf

getragen werden. Die Diffusion beschreibt die Änderungen in der Ausbreitungsmaske $s = s(t; x, y, z)$ aufgrund der Daten im Ursprungsbild $u = u(x, y, z)$ in jedem Zeitschritt t :

$$\partial s / \partial t = \text{div} (g(|\nabla u|) \nabla s) \quad (1)$$

$g(|\nabla u|)$ skaliert die Gradienten der Daten im Ursprungsbild, damit der Diffusionsfluß an Kanten gestoppt wird. Als effektiv und brauchbar haben sich zwei nichtlineare, inhomogene, isotrope Modelle erwiesen. Ein Modell wird schon bei Sherbondy et al. verwendet, stammt von Perona und Malik [7] und ist definiert in Formel (2). Das andere von uns implementierte Modell stammt von Tukey [8] und hat seinen Ursprung in der robusten Statistik (Formel (3)).

$$g(w) = e^{-(w^2/\lambda^2)} \quad (2) \quad g(w) = \begin{cases} \frac{1}{2} [1 - (\frac{w}{\lambda})^2]^2, & |w| \leq \lambda \\ 0 & \text{sonst.} \end{cases} \quad (3)$$

Der Parameter λ skaliert den Gradienten und beeinflusst so die Kantendetektion. In der Praxis hat sich gezeigt, dass Tukey schärfere Kanten erzeugt und ein passender Wert für den Parameter λ leichter einstellbar ist.

Um bei leichten Intensitätsübergängen ein Ausrinnen der Entwicklungspunkte zu verhindern, haben wir die von Sherbondy verwendete Diskretisierung erweitert, indem wir in $g(w)$ für w die Länge des Gradienten in Abhängigkeit aller Dimensionen (z.B. Zentraldifferenzen) verwenden.

Implementierung. Die Implementierung erfolgt auf einer handelsüblichen PC Grafikkarte (ATI 9800XT mit 256MB Speicher). Sämtliche Daten (das Ursprungsbild und die Ausbildungsmaske) werden in Texturen gespeichert. Die Berechnungen finden in Fragmentshadern statt. Details sind in [9] beschrieben. Um die volle Geschwindigkeit der Grafikkarte nützen zu können, müssen die Ursprungsdaten und die Ausbreitungsmaske in den Speicher der Grafikkarte passen.

3 Ergebnisse

Interaktion Der Fortgang der Segmentierung im Kontext des Gesamtvolumens und bereits segmentierter Objekte können in Echtzeit dargestellt werden. Das sofortige visuelle Feedback unterstützt eine interaktive Einstellung der wenigen Parameter und damit eine Anwendung auf eine Vielzahl unterschiedlicher Datensätze. Des weiteren ermöglichen die unterschiedlichen Diffusionsmodelle gekoppelt mit den zwei Arten der Gradientenberechnung eine Anpassung an das zu identifizierende Objekt. Der einfache interaktive Arbeitsablauf ermöglicht auch das Zuweisen unterschiedlicher Transferfunktionen und Darstellungsmöglichkeiten während das Fließen der Diffusion beobachtet werden kann.

Segmentierung Der Algorithmus wurde auf CT Datensätzen zur Segmentierung des Dickdarms (Luft), von Handknochen, Blutgefäßen (Kontrastmittel) und des Herzmuskels (Kontrastmittel) getestet. Gute Ergebnisse konnten bei der Segmentierung des Dickdarms erzielt werden (Abb. 2 links). Die Größe des Datensatzes wurde von $512 \times 512 \times 544$ auf $256 \times 256 \times 128$ reduziert, damit die Daten in den Speicher der Grafikkarte passen. Um den Dickdarm vollständig zu segmentieren waren im schlechtesten Fall ca. 750 Iterationen notwendig (ein Initialisierungspunkt am Ende des Darms), was auf einem Intel P4 2.8GHz, 1GB Arbeitsspeicher etwa 100 Sekunden entspricht. Die Geschwindigkeit der Segmentierung kann durch zusätzliches Setzen von Ausbreitungspunkten signifikant erhöht werden. Im Handdatensatz wurden dickere Blutgefäße, Knochen und Spongiose schnell identifiziert, Probleme treten jedoch bei Elementen auf, die schmaler als drei Voxel sind.

Ebenfalls gute Ergebnisse lieferte der Algorithmus bei der Segmentierung des linken Herzventrikels. Obwohl das Gradientenbild keine Struktur aufweist, ermöglicht ein Filtern des Bildes mit dem selben Diffusionsalgorithmus eine anschließende diffusionsbasierte Segmentierung (Abb. 2 Mitte und rechts).

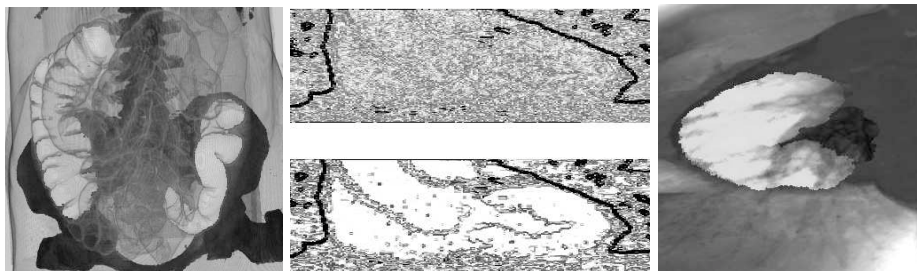


Abb. 2. LINKS: Dickdarmsegmentierung ohne Schattierung; MITTE: Gradientenlänge in CT von einem Herz (oben) und nach Bildglättung mit Perona-Malik Diffusion (unten); RECHTS: Segmentierung des Myokardium des linken Ventrikels in der 3D Ansicht

4 Diskussion und Zusammenfassung

Der hier vorgestellte Ansatz ist ein gradientenbasiertes Verfahren und erbt damit auch alle Nachteile solcher Methoden. Der Algorithmus neigt zum Überlaufen an nicht geschlossenen Konturen. Sobald aber so ein Ausfließen bemerkt wird, kann diese Region speziell gekennzeichnet und der Fluß der Diffusion gestoppt werden. Noch besser funktioniert ein Vorverarbeitungsschritt, in der die Diffusion zur Bildglättung und Kantenverstärkung verwendet wird. Der selbe Algorithmus wird statt auf der Ausbreitungsmaske als Vorverarbeitungsschritt am Bild selber ausgeführt und läuft ebenfalls in Echtzeit.

Bewährt haben sich auch die Erweiterungen zu Sherbondy. Das Diffusionsmodell nach Tukey stoppt besser an Kanten. Die andere Art der Gradientenberechnung unterstützt die Kantenerkennung bei leichten Intensitätsunterschieden

in einer Dimension. Da die Ausbreitungsbereiche nur wachsen sollen, kann man die Segmentierung beschleunigen, indem man den Betrag der Änderung (rechte Seite in der Diffusionsformel (1)) verwendet.

Die Diskretisierung der Diffusion nach Perona und Malik erlaubt nur eine Ausbreitung in eine 6er Nachbarschaft in dreidimensionalen Datensätzen. Dadurch wird die Segmentierung bei dünnen Strukturen (schmale Blutgefäße, dünne, schräge Knochen) abgebrochen. Eine Erweiterung auf eine 27er Nachbarschaft ist denkbar, kann aber auf der Grafikkarte ATI 9800XT wegen der eingeschränkten Länge der Fragmentshader nicht implementiert werden. In der neuen Grafikkartengeneration (seit August 2004 erhältlich) ist das schon möglich.

Die Parametereinstellungen stellen sich trotz interaktiven Feedbacks manchmal schwieriger heraus, als vermutet, da schon kleine Änderungen in den Parametern große Auswirkungen auf das Ergebnis haben können.

Weitere Anwendungsbeispiele, Informationen zum Algorithmus und seiner Implementierung befinden sich in [9]. Wir haben bestätigt, dass ein Diffusionsprozess erfolgreich auch zur Volumensegmentierung verwendet werden kann und eine Implementierung auf Grafikkarten einen interaktiven Prozess ermöglicht. Außerdem haben wir gezeigt, dass weitere Diffusionsmodelle und Gradientenberechnungen bei einigen Problemstellungen bessere Ergebnisse liefern. Die Nachteile des gradientenbasierten Ansatzes könnten in der leistungsstärkeren neuen Grafikkartengeneration durch Integration anderer Einflußfaktoren (z.B. Krümmung) in die Diffusionsmodelle überwunden werden.

Danksagung. Dieses Forschungsprojekt wurde im Rahmen des KPlus Förderprogrammes und mit Hilfe von TIANI medgraph finanziert.

Literaturverzeichnis

1. Engel, K., Hadwiger, M., Kniss, J., Lefohn, A., Rezk-Salama, C., Weiskopf, D.: Real-Time Volume Graphics. Course Notes Course#48, ACM SIGGRAPH (2004)
2. Harris, M.: General Purpose Computations on GPUs (GPGPU) web page. <http://www.gpgpu.org/>. (2003)
3. Lefohn, A., Cates, J., Whitaker, R.: Interactive, GPU-based level sets for 3D brain tumor segmentation. In: Proceedings of MICCAI 2003. Volume 1. (2003) 564–572
4. Sherbondy, A., Houston, M., Napel, S.: Fast volume segmentation with simultaneous visualization using programmable graphics hardware. In: Proceedings of IEEE Visualization 2003. (2003) 171–176
5. Weickert, J.: A review of nonlinear diffusion filtering. Scale Space Theories in Computer Vision (1997) 3–28
6. Hadwiger, M.: High-Quality Visualization and Filtering of Textures and Segmented Volume Data on Consumer Graphics Hardware. PhD thesis, TU Wien (Vienna University of Technology) (2004)
7. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990)
8. Black, M.J., Sapiro, G., Marimont, D., Heeger, D.: Robust anisotropic diffusion. IEEE Trans. on Image Processing **7** (1998) 421–432
9. Langer, C.: Interactive diffusion-based volume segmentation on graphics hardware. Master's thesis, TU Wien (Vienna University of Technology) (2004)