

GPU-basiertes Volumenrendering von multimodalen medizinischen Bilddaten in Echtzeit

Ingrid Scholl¹, Nicole Schubert¹, Pascal Ziener¹, Uwe Pietrzyk²

¹Fachbereich für Elektrotechnik und Informationstechnik, FH Aachen

²Institut für Neurowissenschaften und Medizin, Forschungszentrum Jülich
scholl@fh-aachen.de

Kurzfassung. Die vorliegende Arbeit zeichnet sich dadurch aus, dass registrierte unsegmentierte Volumina aus multimodalen Bilddatensätzen (z.B. MRT, PET) gleichzeitig in einer 3D-Rekonstruktion visualisiert werden und in Echtzeit manipuliert werden können. Ermöglicht wird die Echtzeitfähigkeit durch die Programmierung der Algorithmen zur direkten Volumenvisualisierung auf der Grafikkarte mittels der neuen CUDA-Technologie. Die Zuordnung der Farbeigenschaften wird über 1D-Transferfunktionen für jedes Volumen getrennt gesteuert. So können durch die interaktive Veränderung der 1D-Transferfunktion Detailinformationen aus den zwei Bilddatensätzen getrennt kontrolliert werden und die Vorteile der verschiedenen Bildmodalitäten in einer Visualisierung genutzt werden. Mittels dieses interaktiven *Frameworks* können neue Erkenntnisse insbesondere über neurodegenerativen Erkrankungen gewonnen werden.

1 Einleitung

Zur Diagnostik neurodegenerativer Erkrankungen des menschlichen Gehirns sowie weiterer Tumorerkrankungen werden mehrere bildgebende Untersuchungen am gleichen Patienten durchgeführt, um die Vorteile der verschiedenen bildgebenden Verfahren zu nutzen. Die gewonnenen multimodalen Bilddatensätze aus MRT, PET und CT werden in der Regel einzeln betrachtet und vergleichend bewertet. Eine 3D Rekonstruktion der Bilddatensätze liefert dem Mediziner zusätzliche Informationen und kann durch die Verfahren der direkten Volumenvisualisierung gewonnen werden. Wenn eine höhere Qualität bei der 3D Visualisierung gewünscht wird, können einfachere Verfahren der Volumenvisualisierung wie 2D- und 3D-Texture-Slicing nicht verwendet werden. In diesem Fall wird das Verfahren der direkten Volumenvisualisierung mittels Ray Casting für uniforme Gitter verwendet. Hierbei wird die Licht-Transferfunktion entlang von Sehstrahlen durch das Volumen bestimmt. Die abgetasteten Werte entlang des Sehstrahls werden trilinear aus den umgebenden Voxel-Werten interpoliert und aufaddiert.

Um Echtzeitfähigkeit zu erreichen, werden die Algorithmen zur Volumenvisualisierung auf die Grafikkarte ausgelagert [1, 2]. Dazu stehen C-ähnliche Programmiersprachen, wie CUDA von Nvidia zur Verfügung, um beliebige parallele

Algorithmen mit Hilfe der grafischen Prozessoreinheit (GPU/GPGPU) anstatt der CPU zu berechnen.

Schwierig wird es, wenn mehrere Volumendaten gleichzeitig in einem Volumen abgebildet werden sollen, die zunächst durch eine Vorverarbeitung registriert werden müssen. Die Verfahren zur multimodalen Visualisierung durch das Ray Casting unterscheiden sich hierbei in der Literatur, ob nur eine Transferfunktion für alle Volumina verwendet wird, oder ob eine Transferfunktion pro Volumen eingesetzt wird. Durch die Verwendung von 1D-Transferfunktionen können Farbe und Transparenz durch die Kombination der abgetasteten Ray Casting Strahlenwerte gesteuert werden. Verschiedene Techniken der Farb- und Transparenzzuordnung werden in [3] vorgestellt.

2 Material und Methoden

Gegeben sind *rigide* registrierte MRT und CT sowie MRT und PET Bilddatensätze von mehreren Untersuchungen des menschlichen Gehirns und Körpers der Größe 512^3 oder 256^3 .

Die Abbildung 1 zeigt den von uns implementierten Ansatz: Nachdem die Bilddaten aufeinander registriert wurden, werden für jedes Volumen 1D-Transferfunktionen bestimmt. Die Volumendaten sowie deren Transferfunktionen werden auf die GPU transferiert. Dort wird pro Volumen ein direktes Volumenrendering mittels Ray Casting durch das Front-to-Back-Verfahren in Echtzeit berechnet, wobei C_{src} der abgetastete Farbwert auf dem Sehstrahl entspricht, α_{src} der dazugehörige Opazitätswert nach der 1D-Transferfunktion und C_{dst} sowie α_{dst} die entsprechenden akkumulierten Werte entlang des Sehstrahls [4]:

$$C_{dst} \leftarrow C_{dst} + (1 - \alpha_{dst})\alpha_{src}C_{src} \quad (1)$$

$$\alpha_{dst} \leftarrow \alpha_{dst} + (1 - \alpha_{dst})\alpha_{src} \quad (2)$$

Das Ray Casting wurde mit dem Verfahren der Early Ray Termination optimiert, so dass die Abtastwerte nicht weiter in die Berechnung einfließen, wenn der akkumulierte α_{dst} -Wert einen vorgegebenen Schwellwert t erreicht. Den Ergebnissen pro Pixel werden über die volumenspezifische Transferfunktion eine

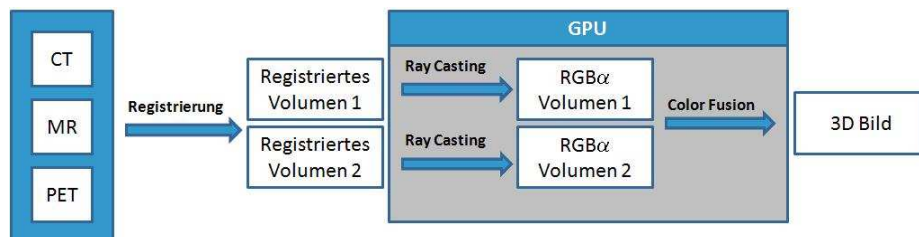


Abb. 1. Workflow der Implementierung.

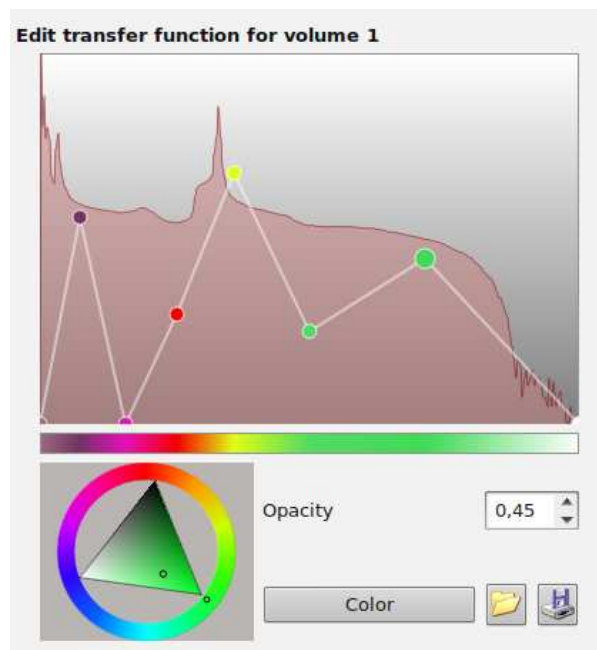
Farbe und eine Transparenz zugeordnet, so dass man pro Volumen und Pixel jeweils einen $RGB\alpha$ -Wert erhält. Durch das Verfahren der Color Fusion wird ein kombinierter Farbwert aus den einzelnen Farbwerten der multimodalen Volumen pro Pixel zugeordnet, der letztendlich als gewichtete Summe aus diesen für die Farbe und die Transparenz berechnet und dargestellt wird [3]:

$$C_{dst} = \beta C_{dst,V1} + (1 - \beta) C_{dst,V2} \quad (3)$$

$$\alpha_{dst} = \beta \alpha_{dst,V1} + (1 - \beta) \alpha_{dst,V2} \quad (4)$$

Die Transferfunktionen können interaktiv mit Hilfe der GUI aus Abb. 2 editiert werden. Die Änderungen werden in Echtzeit auf die Datensätze übertragen und in der 3D Visualisierung direkt sichtbar.

Abb. 2. GUI zur Definition der 1D-Transferfunktion pro Volumen. Im Hintergrund ist das Histogramm der Ray Casting Strahlen zu einem Volumen abgebildet. Die Farben wie auch die Opazität können vom Benutzer interaktiv gesetzt werden und bewirken eine direkte Auswirkung auf das multimodale Visualisierungsergebnis.



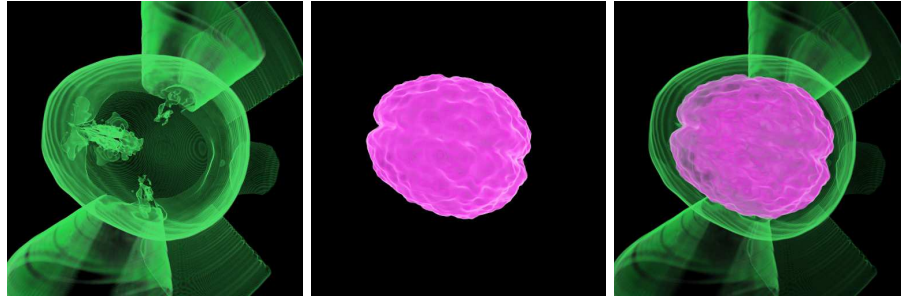
3 Ergebnisse

Der Algorithmus wurde auf einem System mit Intel Core i7 920 Prozessor mit 2.67 GHz, einer Nvidia Quadro FX 5800 Grafikkarte mit 4 GB Grafikspeicher und 12 GB RAM implementiert. Das Programm ist unter Ubuntu 9.04 64 bit und Windows XP 64 bit lauffähig.

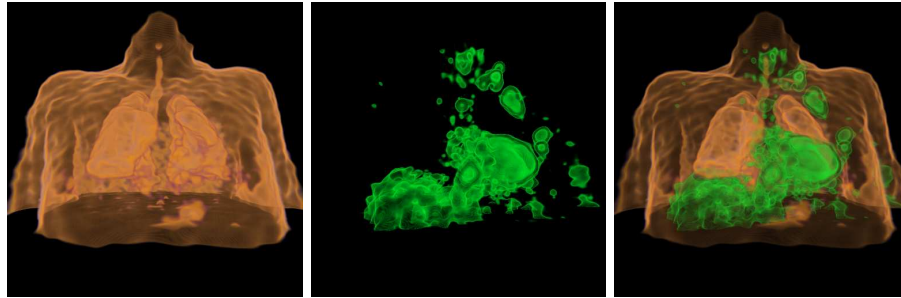
Die Visualisierungsergebnisse sind in Abb. 3 zu sehen. In allen Darstellungen sind die Datensätze einzeln und zusammen veranschaulicht mit unterschiedlichen Transferfunktionen mit $\beta = 0.5$ nach Gl. 4. Es ist sehr gut zu sehen, dass

der Informationsgehalt aus den Datensätzen in der gemeinsamen Visualisierung zunimmt (Abb. 3b,c).

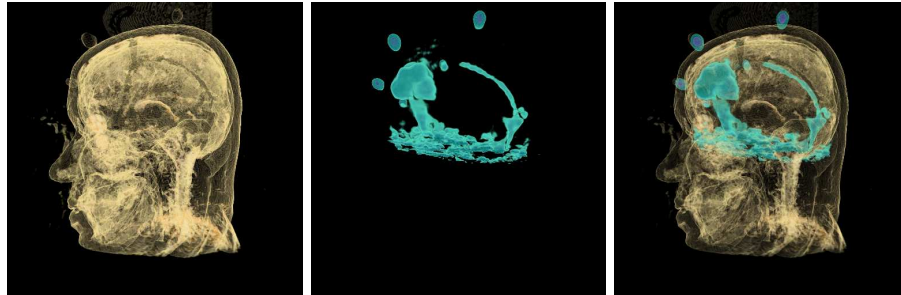
Der Algorithmus wurde daraufhin untersucht, wie sich die Bildwiederholrate pro Sekunde (fps) in Abhängigkeit der Auflösung des gerenderten Ausgabebildes und der Anzahl der maximalen Abtastwerte für alle Ray Casting Strahlen verändert. In Tab. 1 ist das Ergebnis dargestellt. Es ist deutlich, dass mit steigender Anzahl der Abtastwerte die Bildwiederholrate pro Sekunde abnimmt.



(a) Visualisierung von CT und PET Datensätzen des menschlichen Kopfes



(b) Visualisierung von 2 PET Datensätzen des menschlichen Thorax



(c) Visualisierung von MRT und PET Datensätzen des menschlichen Kopfes

Abb. 3. Visualisierung von multimodalen Datensätzen der Größe 512×512 . Links: erster Datensatz; Mitte: zweiter Datensatz; rechts: multimodaler Datensatz.

Tabelle 1. Messwerte: Framerate pro Sekunde für die maximale Anzahl von Abtastwerten (berechnet aus Schrittweite der Abtastwerte des Sehstrahls und Volumengröße).

Schrittweite	Bildgrösse	Max. Abtastwerte	fps
0.01	512×512	52.428.800	82
0.005	512×512	104.857.600	42
0.01	1024×1024	209.715.200	26
0.005	1024×1024	419.430.400	16
0.001	512×512	524.288.000	12
0.001	1024×1024	2.097.152.000	5

4 Diskussion

Der vorgestellte Algorithmus bietet eine neue Applikation für die multimodale Visualisierung medizinischer Bilddaten in Echtzeit. Durch die interaktive Steuerung der Transferfunktionen der Volumen können die Eigenschaften beider Volumina visualisiert werden. Ohne Echtzeitfähigkeit ist das richtige Setzen der Transferfunktionen ein mühsamer Prozeß, der hier durch die neue Technik der GPU Programmierung umgangen werden konnte. Um die Methode weiter zu optimieren, ist die Nutzung von zusätzlichen Grafikkarten vorgesehen, um auch große Volumen darzustellen. Dazu wird auch die Nutzung von Komprimierungstechniken, wie z.Bsp. Wavelet Komprimierung, notwendig, um so den Speicherbedarf der Volumen zu reduzieren. Desweiteren werden andere Möglichkeiten implementiert, die die beiden Volumen fusionieren, wie Depth Peeling oder Opazität gewichtetes Mischen der Volumen [5].

Danksagung. Wir danken dem Bundesministerium für Bildung und Forschung für die Unterstützung dieser Arbeit.

Literaturverzeichnis

1. Stegmaier S, Strengert M, Klein T, et al. A simple and flexible volume rendering framework for graphics-hardware-based raycasting. Proc Vol Graph. 2005; p. 187–95.
2. Weiskopf D. GPU-based Interactive Visualization Techniques. Berlin: Springer; 2006.
3. Beyer J, Hadwiger M, Wolfsberger S, et al. High-quality multimodal volume rendering for preoperative planning of neurosurgical interventions. Proc IEEE Vis. 2007; p. 1696–703.
4. Engel K, Hadwiger M, Kniss J, et al. Real-Time Vol Graph. Peters; 2006.
5. Brecheisen R, i Bartroli AV, Platel B, et al. Flexible GPU-based multi-volume ray-casting. Proc VMV. 2008; p. 303–12.