

GPU-basierte Verarbeitung von hochaufgelösten Videodaten zur Navigation in der laparoskopischen Prostatektomie

M. Müller¹, A. Seitel¹, M. Baumhauer¹, A. Jalal³, D. Teber³, J. Rassweiler², H.P. Meinzer¹ und Lena Maier-Hein¹

¹Abteilung für Medizinische und Biologische Informatik,
Deutsches Krebsforschungszentrum (DKFZ), Heidelberg
²Urologische Klinik des SLK-Klinikums am Gesundbrunnen Heilbronn
³Universitätsklinik Heidelberg

Kontakt: michael.mueller@dkfz-heidelberg.de

Abstract:

In der medizinischen Bildverarbeitung müssen immer größere Datenmengen verarbeitet werden, während die Echtzeitfähigkeit der eingesetzten Systeme mitunter eine entscheidende Rolle spielt. Um diesen Anforderungen gerecht zu werden, werden zunehmend Grafikkarten-Prozessoren (GPU, graphics processing unit) eingesetzt, die aufgrund ihrer Architektur prädestiniert sind, um parallele Rechenschritte auf großen Datensätzen durchzuführen. Insbesondere für minimal-invasive, laparoskopische Eingriffen, in denen eine Echtzeitanzeige der Videobilddaten unabdingbar ist, birgt eine GPU-basierte Datenverarbeitung großes Potential. In diesem Beitrag stellen wir die GPU-basierte Optimierung von Algorithmen vor, welche innerhalb eines Navigationssystems für die laparoskopische Entfernung der Prostata zum Einsatz kommen. Für die Verarbeitung der hochaufgelösten Laparoskopie-Bilder konnten Geschwindigkeitszuwächse um den Faktor 6 erzielt werden. Dies zeigt das Potential von modernen GPUs für bildverarbeitende Echtzeitsysteme in der Laparoskopie.

Schlüsselworte: Augmented Reality, Navigation, Prostatektomie, GPU, laparoskopisch

1 Problem

Insbesondere für bildbasierte Navigationssysteme in der Medizin ist die Echtzeitfähigkeit ein wichtiges Kriterium, während der Umfang der anfallenden Daten stetig zunimmt. Dabei werden zunehmend Grafikkarten-Prozessoren (GPU, graphics processing unit) eingesetzt, die sich aufgrund ihrer Architektur für die parallele Verarbeitung von großen Datenmengen eignen. So werden GPUs eingesetzt, um in Echtzeit digitale Röntgenrekonstruktionen für ein C-Bogen-basiertes Navigationssystem zu erzeugen [1]. Neben reinen Visualisierungstechniken werden GPUs zunehmend auch für Registrierungsprobleme in der medizinischen Bildverarbeitung eingesetzt, um den Echtzeitanforderungen in intra-operativen Umgebungen gerecht zu werden [2]. Ein Navigationssystem für die laparoskopische Entfernung der Prostata verarbeitet hochaufgelöste, laparoskopische Videodaten, um Strukturen aus einem intra-operativen 3D-Ultraschall auf dem realen Videobild anzeigen zu können (Abb. 1) und wurde in [3] vorgestellt. Bewegungen des Laparoscops und/oder des Organs müssen in Echtzeit erfasst werden, so dass die eingeblendeten Strukturen passgenau ohne Verzögerung adaptiert werden können. Somit ist die Echtzeitfähigkeit für das entwickelte System von höchster Bedeutung. Um diese zu gewährleisten und gleichzeitig den hohen Detailgrad von hochaufgelösten Videodaten nutzen zu können, wurden vorhandene Algorithmen auf die GPU portiert. In diesem Beitrag stellen wir die Ergebnisse dieser Überarbeitung bezüglich Laufzeitverbesserung vor.

223

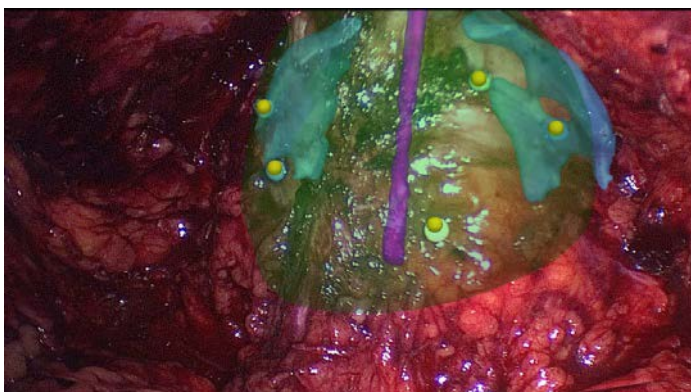


Abb. 1 Darstellung von semi-transparenten Oberflächen auf dem laparoskopischen Videobild (Erweiterte Realität, engl. Augmented Reality, AR, aus [4]).

2 Methoden

Zusammengefasst besteht das Navigationssystem aus folgenden Schritten (für weitere Informationen siehe [3]):

1. Präoperative Kalibrierung der laparoskopischen Kamera.
2. Einbringen von nadelförmigen Navigationshilfen mit farbigen, kugelförmigen Enden in die Prostata.
3. Durchführung eines intraoperativen 3D Transrektal-Ultraschalls.
4. Lokalisation der Nadel-Enden im Ultraschall-Bild und Segmentierung der relevanten Strukturen

(Prostata- Kapsel, Urethra, Neuro-vaskuläre Bündel).

5. Einblenden der relevanten Strukturen in das Videobild durch Echtzeitanalyse der eingehenden Videobilddaten. Dabei kann die Echtzeitanalyse grob in die Teile *Farbraumsegmentierung*, *Registrierung* und *Bildentzerrung* unterteilt werden.

Die Registrierung von Videobild und Ultraschallbildvolumen erfolgt mit Hilfe der eingebrachten Navigationshilfen. Die farbigen Köpfe müssen hierzu in Echtzeit in den laparoskopischen Videodaten detektiert werden. Hierfür werden die ankommenden Videobilder in den HSV-Farbraum konvertiert. Jeder Farbkanal wird dann anhand von empirisch festgelegten HSV-Werten gefiltert. Aus den so gewonnenen Binärbildern werden die Konturen extrahiert und die wahrscheinlichsten Kandidaten als Navigationshilfe festgelegt. Die Konvertierung der Farbraums, die Zerlegung in die einzelnen Farbkanäle, sowie die Anwendung einer Segmentierung mit zweiseitigem Schwellwert wurden hier als parallelisierbare Einzelschritte auf die Grafikkarte portiert.

Aus den Korrespondenzen der so extrahierten 2D-Bildpunkte und den einmalig lokalisierten 3D-Punkten im Ultraschallvolumen wird eine Registrierung mittels projektiver Geometrie errechnet [5]. Hier wird ein Levenberg-Marquardt Algorithmus angewendet, der durch Minimierung des Rückprojektionsfehlers eine geeignete Registrierung errechnet. Für die Eliminierung von Ausreißern wird ein RANSAC-Ansatz (engl. „*random sample consensus*“, [6]) verwendet. Die einzelnen, voneinander unabhängigen Iterationen werden nach der Umstellung auf die GPU parallel ausgeführt.

Damit die zuvor segmentierten Strukturen korrekt dargestellt werden, müssen die einzelnen Videobilder schließlich entzerrt werden. Bei der Entzerrung werden die Pixel der eingehenden Bilder einzeln anhand der kalibrierten Verzerrungsparameter im Bild verschoben und durch Interpolation ein neues verzerrungsfreies Bild ermittelt. Die Verschiebung und Interpolation kann für einzelne bzw. eine Untermenge von Pixeln unabhängig auf der GPU berechnet werden.

Für große Teile der Implementierung wurde die Computer-Vision Bibliothek OpenCV verwendet, die in der Version 2.3. die Möglichkeit bietet, mittels Nvidia CUDA Technologie („*Compute Unified Device Architecture*“), Algorithmen auf Grafikkarten-Prozessoren auszuführen. CUDA bietet eine Schnittstelle, um eigene Algorithmen auf Nvidia-Grafikkarten zu implementieren und von der Leistung moderner GPUs zu profitieren.

Für die Experimente wurde folgende Hardware verwendet: Intel Quad CPU Q6600 2.40GHz, 4,00 GB RAM, NVIDIA GeForce GTX 460. Die Evaluierung der portierten Algorithmen wurde auf aufgezeichneten, hochauflösten Videodaten aus bereits durchgeführten Prostata-Resektionen durchgeführt.

3 Ergebnisse

Die drei folgenden Diagramme fassen die Ergebnisse der GPU- und CPU-basierten Implementierung für die jeweiligen Algorithmen zusammen. Abb. 2 zeigt die Geschwindigkeitsveränderung bei der Farbraumsegmentierung, Abb. 3 und Abb. 4 stellen die Ergebnisse der Re-Implementierung des Registrierungsalgorithmus und der Bildentzerrung grafisch dar.

224

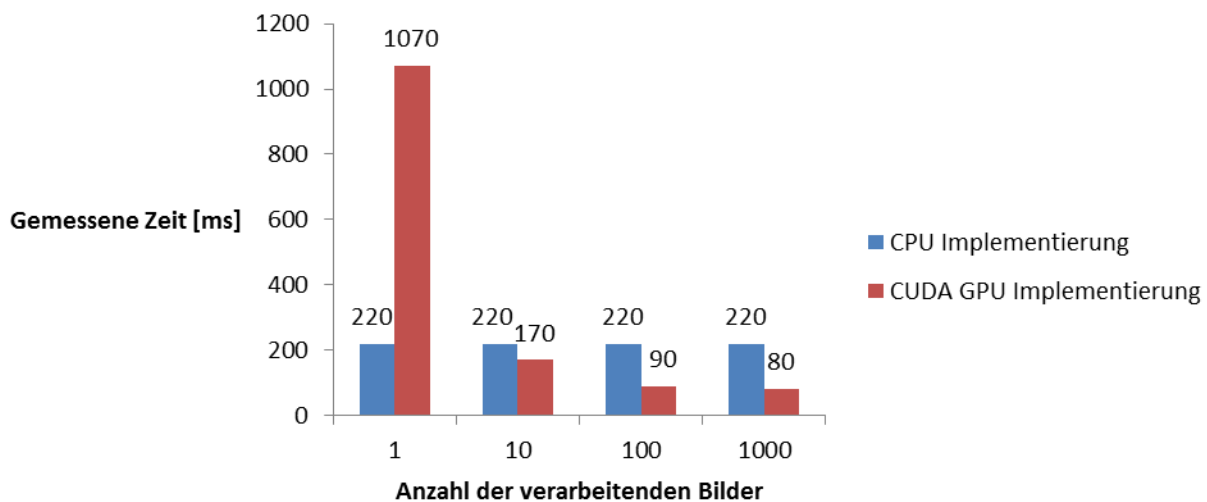


Abb. 2 Laufzeiten von CPU- und GPU-Implementierung der *Farbraumsegmentierung* in der Gegenüberstellung.

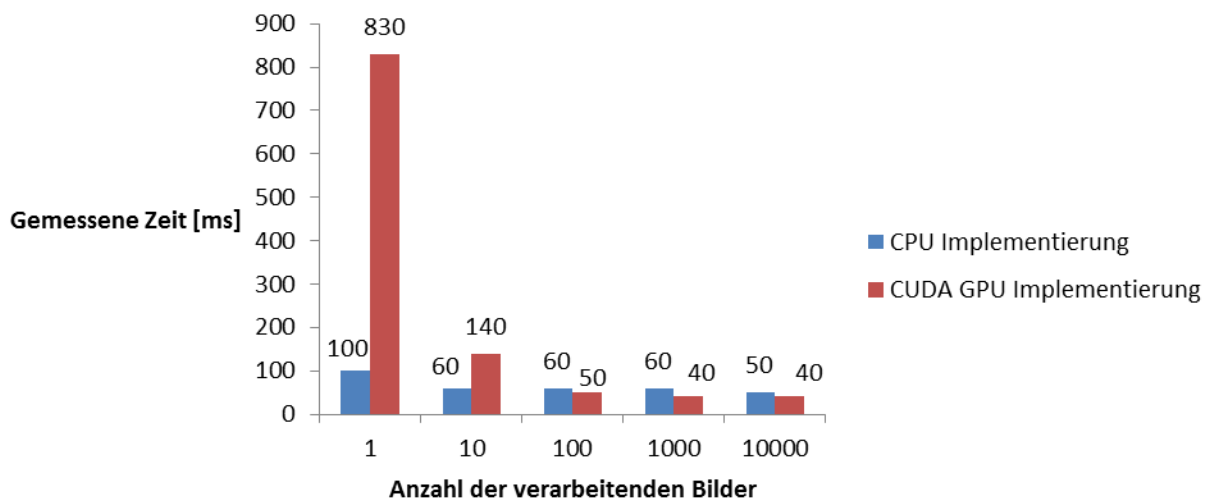


Abb. 3 Laufzeiten von CPU- und GPU-Implementierung des *Registrierungs-Algorithmus* in der Gegenüberstellung.

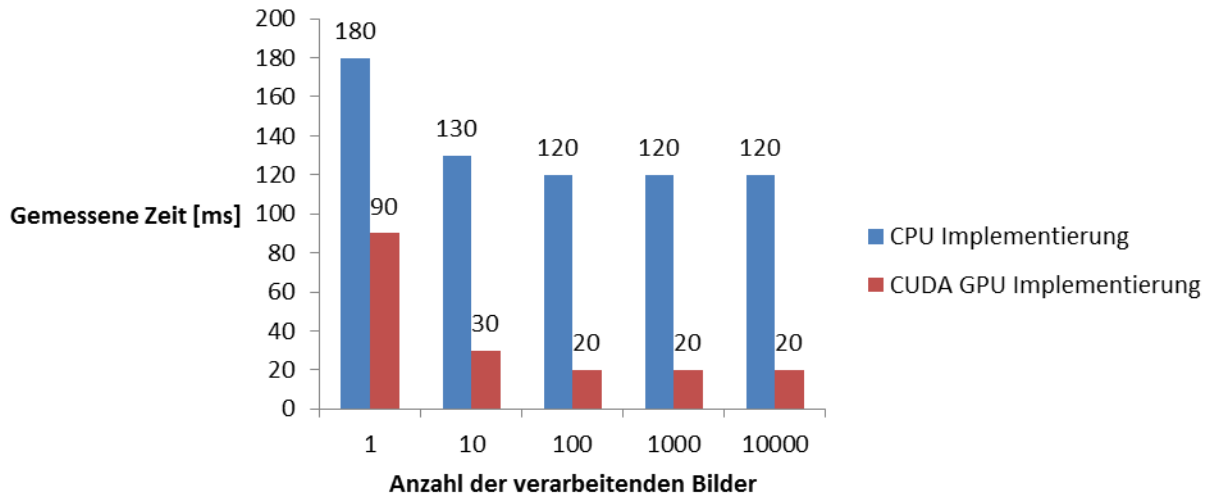


Abb. 4 Laufzeiten von CPU- und GPU-Implementierung der *Bildentzerrung* in der Gegenüberstellung.

4 Diskussion

Wie die Ergebnisse zeigen, erfüllt Re-Implementierung der für die Navigation verwendeten Algorithmen die Erwartungen an den Geschwindigkeitszuwachs und garantiert so eine nahezu echtzeitfähige Computerassistenz während der laparoskopischen Prostatektomie. Wie die Ergebnisse für die Re-Implementierung der Farbraumsegmentierung und des Registrierungsalgorithmus zeigen, kostet die initiale Speicherallokation auf der Grafikkarte einen beträchtlichen Zeitanteil. Solange aber die Auflösung der ankommenden Videobilder konstant ist, amortisiert sich die zeitaufwändige Speicherallokation bereits nach wenigen Zeitschritten. Im Fall der Bildentzerrung ist die Grafikkarte bereits im ersten Durchlauf schneller. Hier lässt sich der größte Geschwindigkeitsunterschied feststellen. Dies ist dadurch zu erklären, dass neben der reinen Speicherallokation beim ersten Durchlauf auch bereits statische Entzerrungsmatrizen aus den Kalibrierungsdaten erstellt werden, die dann im weiteren Programmablauf eine einfache Entzerrung der neu ankommenden Bilder erlauben. Da dieser rechenaufwendige Vorgang bereits auf der Grafikkarte ausgeführt werden kann, ist bereits im ersten Durchlauf eine höhere Geschwindigkeit bei der GPU-Implementierung festzustellen. Demgegenüber ist der geringste Geschwindigkeitsunterschied bei der Registrierung festzustellen. In den Versuchen wurde für die Ausreißer-Eliminierung die Zahl der Iterationen auf 100 festgelegt. Diese geringe Zahl und der vergleichsweise niedrige Rechenaufwand in jeder Iteration sind dafür ausschlaggebend, dass die CPU-Implementierung ähnliche Ausführungszeiten erreicht. Natürlich lässt sich die Zahl der Iterationen beliebig steigern. Allerdings verbessert sich die Registrierungsqualität dabei kaum. Daraus lässt sich folgern, dass nur hoch parallelisierbare Probleme für die Ausführung auf einer GPU lohnenswert sind. Es ist anzumerken, dass der Geschwindigkeitszuwachs in großem Maß von der eingesetzten Hardware abhängt. Die in den Experimenten verwendete NVIDIA GeForce GTX 460 ist ebenfalls nur eine Grafikkarte der mittleren Preisklasse und bietet dennoch einen ausreichenden Geschwindigkeitsgewinn. Grafikkarten in höheren Preisklassen würden noch bessere Ergebnisse erzielen. Dasselbe gilt natürlich auch für die neusten Generationen von CPUs. Insgesamt war der Implementierungs- und Umstellungsaufwand überschaubar. Viele Bildverarbeitungsbibliotheken bieten bereits Implementierungen ihrer Algorithmen auf Basis von GPUs an bzw. stellen einfache Tools zur Verfügung, um eigene Funktionen auf einfache Art und Weise auf der Grafikkarte zu implementieren. Aus diesem Grund lohnt sich die Re-Implementierung von Algorithmen auf Grafikkarten überall da, wo Echtzeitfähigkeit ein wichtiges Kriterium darstellt.

Kommende Arbeiten werden sich vor allem auf die GPU-basierte Auswertung der für die Navigation verwendeten Bilddaten konzentrieren. Insbesondere die Lokalisierung der Navigationshilfen im Ultraschallbild in Echtzeit ist hinsichtlich der Kompensation von Organdeformationen interessant.

5 Referenzen

- [1] Reungamornrat S., Otake Y., Uneri A., Schafer S., Mirota DJ., Nithianathan S., Stayman JW., Kleinszig G., Khanna AJ., Taylor RH., Siewerdsen JH.. An on-board surgical tracking and video augmentation system for C- arm image guidance. *International Journal of Computer Assisted Radiology and Surgery*, 2012
- [2] Shams, R., Sadeghi, P., Kennedy, R., Hartley, R. A Survey of Medical Image Registration on Multicore and the GPU, *Signal Processing Magazine, IEEE*, 2010
- [3] Simpfendorfer, T., Baumhauer, M., Müller, M., Gutt, C. N., Meinzer, H. P., Rassweiler, J. J., Güven, S. & Teber, D. Augmented reality visualization during laparoscopic radical prostatectomy. *Journal of Endourology, Mary Ann Liebert Inc., USA-New Rochelle, NY*, 2011
- [4] Mueller, M., Groch, A., Baumhauer, M., Maier-Hein, L., Teber, D., Rassweiler, J., Meinzer, H.-P. & Wegner, I. Robust and efficient fiducial tracking for augmented reality in HD-laparoscopic video streams. *Proceedings of SPIE*, 2012
- [5] Grest, D., Petersen, T. & Krüger, V. A Comparison of Iterative 2D-3D Pose Estimation Methods for Real-Time Applications. *Proceedings of the 16th Scandinavian Conference on Image Analysis, Springer-Verlag, Berlin, Heidelberg*, 2009
- [6] Richard Hartley, Andrew Zisserman. *Multiple View Geometry in Computer Vision. 2. Auflage. Cambridge University Press, Cambridge*, 2004

6 Danksagung

Die vorliegende Arbeit ist im Rahmen des Graduiertenkollegs 1126 „Intelligente Chirurgie“ der Deutschen Forschungsgemeinschaft (DFG) entstanden. Die entwickelte Software ist Teil der frei verfügbaren Bibliothek „Medical Imaging and Interaction Toolkit“ (MITK), die am Deutschen Krebsforschungszentrum entwickelt wird.