

Mosaickingalgorithmus zur schnellen Panoramabilderstellung in der Fluoreszenzendoskopie

Alexander Behrens¹, Michael Bommers¹, Thomas Stehle¹, Sebastian Gross¹,
Steffen Leonhardt², Til Aach¹

¹Lehrstuhl für Bildverarbeitung, RWTH Aachen University

²Philips Lehrstuhl für Medizinische Informationstechnik, Helmholtz-Institut,
RWTH Aachen University

`alexander.behrens@lfb.rwth-aachen.de`

Kurzfassung. Bisherige Mosaickingalgorithmen zur Erstellung von Panoramabildern der Harnblase aus endoskopischen Fluoreszenzbildern erreichen aufgrund ihres hohen Rechenaufwands nur einen langsamen sukzessiven Panoramabildaufbau. Die Verwendung eines Panoramas als interagierende Orientierungs- und Navigationshilfe erfordert jedoch eine echtzeitnahe Bilderstellung während der Operation. Hierzu wurde ein PDD-Mosaickingalgorithmus entwickelt, der eine schnelle Erstellung und Darstellung von endoskopischen Panoramabildern durch parallelisierte und synchronisierte Multi-Thread Prozesse ermöglicht.

1 Einleitung

Die Sichtung und Therapie von multi-fokalem Harnblasenkrebs erfolgt durch eine Zystoskopie, in der ein starres Endoskop durch die Harnröhre in die Harnblase des Patienten eingeführt wird. Bei Verwendung eines Videoendoskopiesystems wird hierbei Licht durch ein Lichtleiterkabel in die Blase eingestrahlt und ein Videobild durch einen am Okular arretierten Kamerakopf aufgenommen. Während unter Weißlicht der Kontrast zwischen kleinen flachen Tumoren und dem umliegenden gesunden Gewebe gering ist, liefert die photodynamische Diagnose (PDD) eine deutliche Kontrastverstärkung. Hierbei wird unter Verwendung einer Markersubstanz und einer schmalbandigen blauen Lichteinstrahlung (380–450nm) Tumorgewebe zum rötlichen Fluoreszieren angeregt. Zur Erzeugung von gut belichteten und kontrastreichen PDD-Bildern ist aufgrund der schmalbandigen Beleuchtung das Endoskop stets nah ($\leq 1\text{cm}$) an der Blasenwand entlangzuführen. Dies führt zu Ansichten mit einem sehr geringen „Field Of View“ (FOV), welches die Orientierung und Navigation innerhalb der Harnblase erschwert.

Panoramabilder iterativer Mosaickingalgorithmen [1, 2, 3] liefern dagegen ein erweitertes Sichtfeld. Obwohl offline berechnete Panoramabilder und deren Darstellung zur Befundsprotokollierung und Operationsplanung [1] verwendet werden können, ist für eine interagierende Navigationshilfe eine echtzeitnahe Verarbeitung sicherzustellen. Hierzu wird im Folgenden ein neuer PDD-Mosaickingalgorithmus vorgestellt, der durch ein Multi-Thread Softwaredesign innerhalb des

RealTimeFrame [4] eine schnelle Panoramabilderstellung ermöglicht. Mit Hilfe eines herkömmlichen PC-Systems wird hierbei bei voller Bildauflösung (768x576) das sukzessiv wachsende lokale Blasenpanorama live berechnet und direkt während der Blasenpiegelung dargestellt.

2 Methoden

2.1 RealTimeFrame

Für die Entwicklung des Mosaickingalgorithmus wurde die Demonstratorplattform *RealTimeFrame* [4] verwendet. Basierend auf einem Model-View-Controller Konzept stellt das Framework Laufzeitinformationen und ein Hauptfenster zur Livebildanzeige und GUI-Elemente für Benutzereingaben bereit. Der Controller verwaltet Threads zur Bildakquisition und zur sequentiellen Abarbeitung einer konfigurierbaren Prozesskette. Algorithmen kompiliert als Dynamic Link Libraries können in die Prozesskette eingefügt und direkt ausgeführt werden. Dies ermöglicht ein schnelles Umsetzen von Echtzeit-Bildverarbeitungsalgorithmen [5], wie z.B. Kontrastverstärkung, zeitliche Filterung und Gefäßverstärkung.

2.2 Mosaickingalgorithmus

Die Erstellung eines lokalen PDD-Blasenpanoramas erfolgt durch eine Weiterentwicklung des iterativen Mosaickingalgorithmus aus [1, 2]. Hierbei wird zunächst der elliptische Vordergrundbereich in den Endoskopbildern extrahiert. Aufgrund der deutlich abgebildeten Gefäßstruktur der Blasenwand, werden danach einzelne charakteristische Merkmalspunkte durch „Speeded Up Robust Features“ (SURF) [6] extrahiert, welche eine sehr große Wiederholbarkeit in den Bildsequenzen aufweisen. Das Matchen von Punktkorrespondenzen zweier sequentieller Bilder erfolgt mittels der Featuredeskriptoren \mathbf{d} . Als Distanzfunktion wird der minimale euklidische Abstand

$$\min_j \|\mathbf{d}_i - \mathbf{d}_j\|_2 = \min_j \sqrt{\sum_{k=1}^n (d_{i,k} - d_{j,k})^2}, \quad (1)$$

zwischen den n -dimensionalen Deskriptoren $\mathbf{d}_i, \mathbf{d}_j$ herangezogen. Die anschließende Berechnung der Bild-zu-Bild-Homographie

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad \text{mit} \quad \mathbf{A} = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (2)$$

beschreibt eine affine Transformation mit vier Freiheitsgraden (Rotationswinkel α , Skalierung s_x, s_y , Scherung a) und eine Translation \mathbf{t} . Zur robusten Homographieschätzung wird ein adaptiver RANSAC (RANDOM SAMPLE CONSENSUS) Algorithmus [7] verwendet, der falsche Punktkorrespondenzen verwirft. Nach dem Ausführen der Bildtransformation wird abschließend der entstandene Überlappungsbereich mittels einer linearen Interpolation überblendet und ein Kompositionsbild erstellt. Iterativ werden so nachfolgende Videobilder sukzessiv dem Panoramabild hinzugefügt.

2.3 Softwaredesign

Zur schnellen Berechnung und Darstellung des entstehenden Panoramabildes während der Zystoskopie verfügt der Mosaickingalgorithmus, eingebettet in dem *RealTimeFrame*-Framework, eine Multi-Thread Softwarearchitektur und ein adaptives Bufferkonzept. Diese und eine Ablaufbeschreibung der Threads sind in Abb. 1 skizziert. Im ersten Prozessschritt wird das aktuelle Endoskopbild F_n dem *CentralBuffer* übergeben. Dieser verwaltet dafür K *PictureBuffer* bestehend aus jeweils N freien Speicherslots, welche sequentiell gefüllt werden. Vollständig gefüllte Buffer werden von der *EmptyBufferList* in die *ActiveBufferList* übertragen und zur Verarbeitung freigegeben. Da während der Blasensichtung die Freihandbewegung des Endoskops annähernd gleichmäßig und überwiegend langsam ist, weisen zwei direkt aufeinanderfolgende Bilder meist einen unnötig großen Überlappungsbereich auf. Dagegen führen größere und adaptiv gesteuerte Schrittweiten zwischen zwei zu registrierenden Bildern zu einem geringeren Rechenaufwand ohne das erweiterte FOV des Panoramabildes zu beeinflussen. Andererseits ergibt sich eine erhöhte Latenz durch das Buffern von Zwischenbildern, wobei die Anzahl der Speicherslots N gleich der Schrittweite ΔF ist.

Zunächst werden nur das erste und das N -te Endoskopbild zur Verarbeitung markiert und in die *NextElementList* eingereiht. Sobald diese Liste gefüllt wird, aktiviert sich der *DescriberThread*, extrahiert SURF Features und übergibt das Bild samt Deskriptoren d an den *ProcessedPictureBuffer*. Nun berechnet der *MatcherThread* Punktkorrespondenzen und schätzt daraus eine Bild-zu-Bild-Homographie H nach Gl. 2. Diese wird abgespeichert und der zu den Bildern F_1, F_N zugehörige *PictureBuffer* zum erneuten Füllen in die *EmptyBufferList* verschoben, da die Zwischenbilder F_2, \dots, F_{N-1} nicht weiter berücksich-

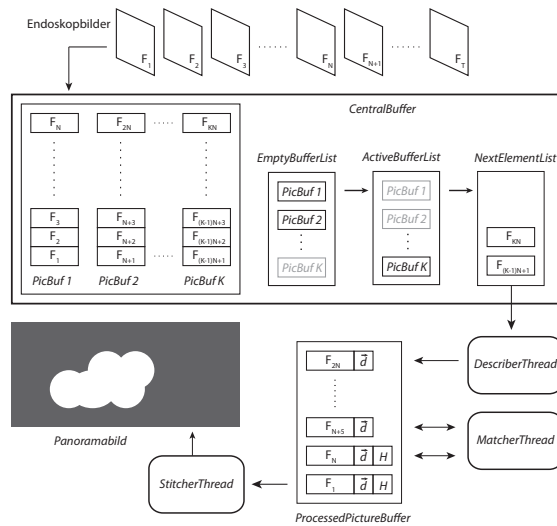


Abb. 1. Softwarearchitektur, Bufferdesign und Thread-Ablaufbeschreibung.

tigt werden müssen. Werden weniger als drei Punktkorrespondenzen bestimmt, so schlägt die Berechnung der Homographie fehl und der *MatcherThread* fordert ein neues Zwischenbild F_n mit $F_1 < F_n < F_N$ aus dem *PictureBuffer* an. Sobald eine Homographie vorliegt, fügt der *StitcherThread* das zugehörige Bild an das sukzessiv wachsende Panoramabild an und interpoliert die überlappenden Bildbereiche.

3 Ergebnisse

Das erzeugte lokale Panoramabild einer realen Zystoskopievideosequenz (609 Frames) ist in Abb. 2(a) dargestellt. Dieses bietet ein vergrößertes FOV der Blasenwand und zeigt zwei papillare Tumore an der linken oberen Blasenwand vom linken Ureterostium. Aufgrund der parallelisierten Prozesse (Abb. 1), bestimmt die Threadssynchronisation die Gesamtzeit zur Berechnung und Aktualisierung des Panoramabilds. Abb. 2(b) zeigt die durchschnittliche Prozesszeit der drei Threads über der Featurepunktanzahl P , gemessen auf einem Standard PC-System mit 2.2 GHz Intel Xeon Dual Prozessoren. Die Prozesszeit des Describers ist hierbei linear abhängig von P , da der Rechenaufwand eines Featuredeskriptors d konstant ist. Der Offsetwert ist durch die vom Bildinhalt unabhängigen Vorverarbeitungsschritte wie Multiskalenrepräsentation, Filteroperationen, usw. bestimmt. Die Prozesszeit des Matchers dagegen erhöht sich quadratisch mit $\frac{P(P-1)}{2}$. Der Stitchingprozess ist zeitlich konstant und unabhängig von P , da ausschließlich die Bildtransformation H und das Blending ausgeführt wird. Der Verlauf der Gesamtzeit zwischen Bildeingang und Panoramabildanzeige ist in Abb. 2(b) als Summation der Einzelprozesszeiten und durch eine sequentielle

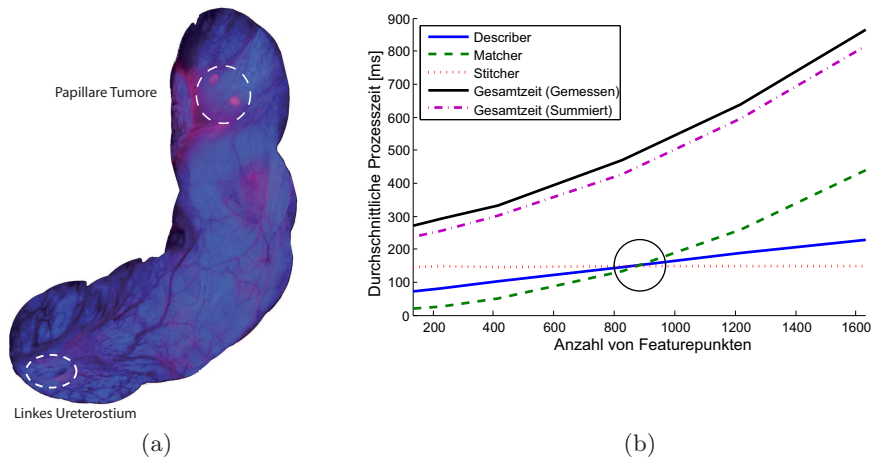


Abb. 2. Blasenpanorama bestehend aus 32 Einzelbildern (a) und durchschnittliche Prozesszeit der Threads über der Anzahl von Featurepunkten (b).

Messung dargestellt. Die konstante Abweichung beschreibt die Rechenzeit des Speichermanagements und allgemeinen Systemoperationen.

4 Diskussion

Der optimale Kompromiss zwischen Rechenzeit und großer Featureanzahl P ist bestimmt durch die Threadsynchrisation. Ein Optimum wird bei gleicher Prozessdauer der parallelisierten Threads erreicht. Dieser Arbeitspunkt ist in Abb. 2(b) durch den Schnittpunkt der Zeitverläufe bei $P_A = 855$ bestimmt. Eine Reduzierung der Gesamtrechenzeit ist somit direkt abhängig von den einzelnen Prozesszeiten der drei Threads. Nach Optimierung der Bildschrittweite ΔF in dem Arbeitspunkt P_A , beträgt die Gesamtverarbeitungszeit für ein Bild mit voller Auflösung (768×576) und Panoramadarstellung ohne jegliche Abwärtsabtastung [1, 2] im Durchschnitt 257.9 ms. Die Bildwiederholrate ergibt sich damit zu 3.8 Bilder/Sekunde. Der Multi-Thread basierte Mosaickingalgorithmus ermöglicht somit eine schnelle sukzessive Erstellung von PDD-Bildkompositionen. Bei einer typischen Zystoskopbewegung, liefert der Algorithmus ein robustes Bildstitching, da der Überlappungsbereich zweier Bildpaare ausreichend groß ist. Aufgrund der erreichten Bildwiederholrate kann das Panoramabild dem Chirurg nun während der Blasenspiegelung auf einem zweiten Monitor als zusätzliche Orientierungshilfe dargestellt und erste klinische Studien erhoben werden. Weitere Prozessanalysen der parallelisierten Frameworks können unabhängig voneinander erfolgen und neue Featuredetektoren, Matching- oder Blendingmethoden implementiert und optimiert werden. Darüber hinaus können zusätzliche Prozesse wie iterative Registrierungsmethoden (z.B. Bundleadjustment [3]) in Form von neuen Threads dem System schnell und einfach hinzugefügt werden.

Literaturverzeichnis

1. Behrens A, Stehle T, Gross S, et al. Local and global panoramic imaging for fluorescence bladder endoscopy. In: Proceedings of EMBC; 2009. p. 6990–3.
2. Behrens A. Creating panoramic images for bladder fluorescence endoscopy. Acta Polytechnica Journal of Advanced Eng. 2008;48(3):50–4.
3. Miranda-Luna R, Daul C, Blondel W, et al. Mosaicing of bladder endoscopic image sequences: distortion calibration and registration algorithm. IEEE Trans Biomed Eng. 2008;55(2):541–553.
4. Gross S, Stehle T. RealTimeFrame - A real time processing framework for medical video sequences. Acta Polytechnica Journal of Advanced Eng. 2008;48(3):15–9.
5. Gross S, Behrens A, Stehle T. Rapid development of video processing algorithms with RealTimeFrame. In: Biomedica. Liege; 2009. p. 217–20.
6. Bay H, Ess A, Tuytelaars T, et al. SURF: Speeded Up Robust Features. Comput Vis Image Underst. 2008;110(3):346–59.
7. Fischler M, Bolles R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM. 1981;24(6):381–95.