

Live-Wire Segmentierung für hochaufgelöste Farbbilder mit optimierter Graphensuche

Thorsten Zerfaß^{1,2}, Sebastian Mues-Hinterwäller¹,
Dietrich Paulus² und Thomas Wittenberg¹

¹Fraunhofer-Institut für Integrierte Schaltungen IIS, Erlangen

²Institut für Computervisualistik, Univ. Koblenz-Landau, AG Aktives Sehen
Email: zerfastn@iis.fraunhofer.de

Zusammenfassung. Das aus der Literatur bekannte Live-Wire Verfahren beschreibt einen konturbasierten, interaktiven Ansatz zur Segmentierung. Die Kontur eines Objekts wird dabei durch einen Pfad minimaler Kosten beschrieben, der durch eine Menge von Saatpunkten führt, die zuvor durch den Benutzer selektiert wurden. Der Live-Wire Ansatz liefert gute Ergebnisse hinsichtlich Segmentierungsqualität und Geschwindigkeit bei minimalem Interaktionsaufwand und dies mit relativ geringem Zeitaufwand. Bei den in der Literatur beschriebenen Verfahren werden für die Berechnung der Pfade oft nur Grauwertbilder verwendet, die Berechnung anhand von Farbbildern wird nur sehr selten behandelt. Diese Arbeit befasst sich intensiv mit deren Verwendung und stellt neue Ansätze vor, die Farbinformationen für die Segmentierung zu nutzen, um hierdurch die Qualität der Segmentierungsergebnisse zu steigern. Zudem wird für den Bereich der Graphensuche innerhalb des Live-Wire Verfahrens bisher der Algorithmus von Dijkstra verwendet. Im Kontext dieser Arbeit wird als alternativer Ansatz der F*-Algorithmus für eine wesentlich effizientere Graphensuche speziell für hochaufgelöste Bilder vorgestellt. Die vorgestellten Verbesserungen des Live-Wire Verfahrens werden abschließend an einer repräsentativen Stichprobe evaluiert.

1 Problemstellung

Durch eine Vielzahl neuer Techniken und Forschungsergebnisse hat sich die Anzahl der verfügbaren medizinischen Bildaufnahme-modalitäten in den vergangenen Jahren deutlich erhöht. Insbesondere farbige Modalitäten werden verstärkt in der Bildgebung eingesetzt. Zudem hat sich eine starke Verbesserung der Bildqualität und Bildauflösung bereits bestehender Verfahren ergeben. Um einen Arzt bei der objektiven Beurteilung der Bilddaten zu unterstützen, bieten moderne Mensch-Maschine-Schnittstellen in Kombination mit Bildverarbeitungsansätzen Möglichkeiten, um wichtige Bereiche in einem Bild interaktiv herauszuheben oder zu markieren. Das Live-Wire Verfahren [1] stellt einen solchen Ansatz dar, um eine Interaktion des Benutzers mit dem System zu ermöglichen, und gleichzeitig gute Segmentierungsergebnisse zu erhalten.

Im Gegensatz zu vollautomatischen Segmentierungsverfahren, die i.d.R. nur applikations-spezifisch gute Ergebnisse liefern, lassen sich Live-Wire Verfahren

universell einsetzen. Zudem liefern Live-Wire Verfahren eine hohe Qualität in Bezug auf Genauigkeit und Geschwindigkeit einer Segmentierung. Um jedoch den eingangs erwähnten Entwicklungen bzgl. neuen – farbigen – Bildmodalitäten sowie gesteigerten Bildauflösungen gerecht zu werden, bedarf es sinnvollen Erweiterungen des Live-Wire Verfahrens. In den letzten Jahren sind hierzu die verschiedensten Ansätze vorgestellt worden.

2 Stand der Forschung

Das Live-Wire Verfahren wurde 1992 von Mortensen und Barrett vorgestellt [1]. Es beschreibt eine interaktive Methode zur Segmentierung von Objektkonturen unter Verwendung des Graphensuchalgorithmus von Dijkstra [2] und einer Funktion zur Berechnung der Kosten für bestimmte Pfade innerhalb eines Bildes anhand von Grauwertintensitäten. Für die Verarbeitung von Farbbildern wird in der Literatur die Anwendung der in [1] vorgestellten Kostenfunktion auf den jeweils einzelnen Farbkanälen vorgeschlagen. Dies führt durchaus zu einer Verbesserung der Ergebnisse, jedoch erhöht sich auch die Berechnungszeit für die gesamte Kostenfunktion.

Die Auswahl einer Kontur erfolgt durch die Selektion eines sogenannten Saatpunktes auf der Kante des zu segmentierenden Objekts. Die Auswahl eines zweiten Punktes auf der Objektkontur initiiert die Berechnung des optimalen Pfades zwischen dem Saatpunkte und dem aktuellen Punkt, wobei der aktuelle Punkt durch die Position des Mauszeigers gegeben ist. Bewegt der Benutzer die Maus, bekommt er jeweils den optimalen Pfad zwischen Mauszeiger und Saatpunkt angezeigt. Hat sich der Benutzer für eine bestimmte Position oder einen bestimmten Pfad entschieden, so beendet er die Markierung der Teilkontur, und startet wiederum die Suche nach einem optimalen Pfad zum nächsten Punkt. Der aktuell selektierte Punkt dient nun als neuer Saatpunkt für die weitere Berechnung. Der beschriebene Vorgang wiederholt sich so lange, bis die gesamte Kontur des Objekts segmentiert wurde. Die dabei entstehende Kontur besteht aus den einzeln selektierten optimalen Pfaden.

3 Fortschritt durch den Beitrag

Der vorliegende Beitrag stellt zwei Erweiterungen des bekannten Live-Wire Verfahrens vor. Zum einen wird das Verfahren auf einen neuen Ansatz zur Verarbeitung von Farbbildern erweitert, zum anderen wird der zur Graphensuche vorgeschlagene Algorithmus von Dijkstra [2] durch den F^* -Algorithmus [3, 4] ersetzt. Durch Verwendung der zusätzlichen Informationen aus den verschiedenen Farbkanälen erlaubt die Erweiterung auf Farbbilder eine Verbesserung der Kantenfindung, außerdem wird die Berechnungszeit durch die Verwendung einer alternativen Kostenfunktion deutlich gesenkt. Der F^* -Algorithmus verbessert im Hinblick auf Bilddaten mit hoher Auflösung die Geschwindigkeit bei der Graphensuche und erlaubt auf diesen somit die effiziente Anwendung des Live-Wire Verfahrens.

4 Methoden

4.1 Erweiterung auf Farbbilder

Bei der Segmentierung von Objekten auf Farbbildern kann die Verwendung der Grauwertintensitäten der einzelnen Bildpunkte bei der Berechnung der Kostenfunktion zu Problemen führen, da es viele Situationen gibt, bei denen wichtige Kanteninformationen zwischen farbigen Objekten bei der Umrechnung in ein Grauwertbild verloren gehen. Eine Erweiterung der Kostenfunktion auf Farbbilder und deren zusätzliche Information erscheint deshalb sinnvoll. Ein einfacher Ansatz hierzu wäre die Verwendung der in [1] vorgestellten Kostenfunktion auf den jeweils einzelnen Kanälen des Bildes. Die Berechnung dieser Kostenfunktion ist jedoch relativ zeitaufwändig, für die Berechnung auf mehreren Farbkanälen sollte deshalb ein anderer Ansatz gewählt werden.

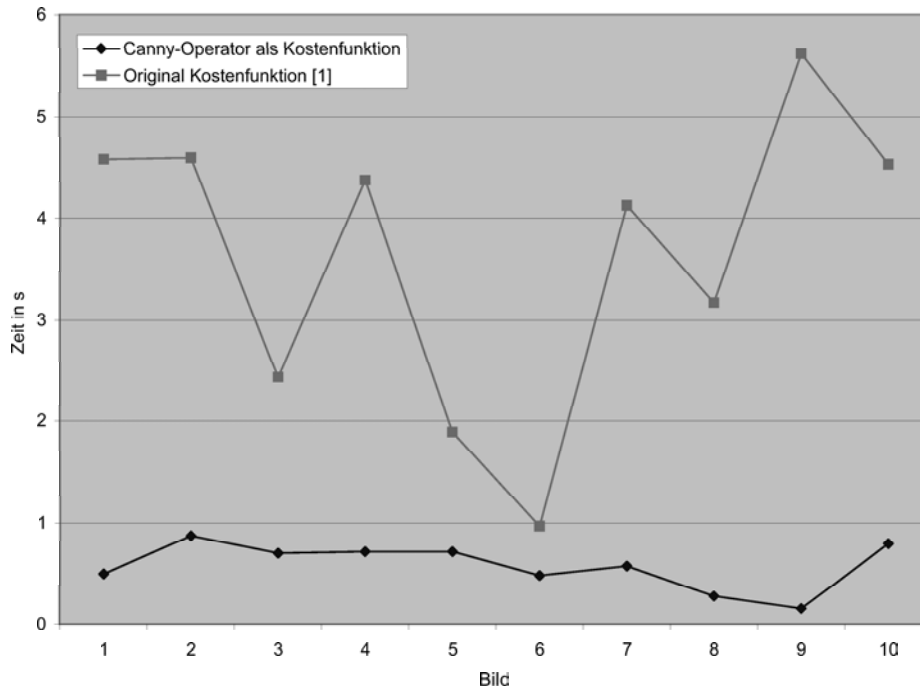
Ein alternativer Ansatz ist die Verwendung des Canny-Operators [5] auf allen drei Kanälen eines Bildes. Die Ergebnisse des Canny-Operators auf den jeweiligen Kanälen werden in einem anschließenden Schritt zusammengefasst, wobei die Resultate dabei mit empirisch ermittelten Werten unterschiedlich gewichtet werden. Die Vorteile dieses Ansatzes sind in der kurzen Berechnungszeit und in den guten Ergebnissen bei der Kantedetektion zu sehen. Weiterhin ergibt sich durch die Resultate, die der Canny-Operator liefert, eine Beschleunigung bei der Graphensuche, da es sich bei dem Ergebnis der Kostenfunktion nur noch um ein Binärbild handelt. Die optimalen Pfade sind dadurch auf einfache Weise und mit geringem Zeitaufwand zu berechnen.

4.2 F*-Graphensuche

Im Bereich der Graphensuche liefert der Algorithmus von Dijkstra gute Ergebnisse, jedoch treten Rechenzeit- und Speicherplatzprobleme auf, wenn sich die Auflösung der Bilddaten deutlich erhöht. Um dieses Problem zu beseitigen wurde der F*-Algorithmus [3, 4] als Alternative zum Dijkstra-Algorithmus [2] untersucht. Algorithmisch unterscheidet sich dieser deutlich von Dijkstra's Ansatz.

Wie der Algorithmus von Dijkstra benötigt der F*-Algorithmus die Angabe einer Kostenfunktion, um die Kosten für die Kanten des Graphen zu bestimmen. Auch die Initialisierung des Pfad-Feldes verläuft analog zum Dijkstra-Algorithmus. Die anschließenden Berechnungen auf dem Pfad-Feld unterscheiden sich jedoch vollständig. Den F*-Ansatz kann man sich als eine Art Distanztransformation auf dem Pfad-Feld vorstellen, bei dem die Werte innerhalb des Pfad-Feldes sukzessive aktualisiert werden. Hierbei sind die verwendeten Masken, sowie die Reihenfolge bei der Aktualisierung entscheidend. Das Pfad-Feld wird in mehreren Richtungen und mit verschiedenen Masken durchlaufen, die Werte innerhalb des Feldes werden dabei stets neu berechnet. Der Algorithmus terminiert, sobald sich in einer Iteration keine Änderung mehr innerhalb des Pfad-Feldes ergibt. Als Resultat erhält man ein Pfad-Feld, das die optimalen Pfade zu einem Saatpunkt enthält. Um den optimalen Pfad von einer bestimmten Position zum Saatpunkt zu erhalten, wird wie beim Dijkstra-Algorithmus ein Backtracking-Verfahren verwendet.

Abb. 1. Berechnungszeiten für die Kostenfunktionen.



5 Ergebnisse

Die beschriebenen Erweiterungen des Live-Wire Verfahrens wurden auf der Basis von 36 Referenzbildern aus unterschiedlichen Bildmodalitäten (CT, Endoskopie, Mikroskopie, ...) und Bildinhalten durch eine Gruppe von zehn Experten evaluiert. Jedes Bild wurde von jedem Experten mit dem original Live-Wire Verfahren sowie mit elf Varianten der modifizierten Version interaktiv segmentiert. Zudem wurde eine Evaluation an synthetisch erzeugten Bildern durchgeführt. Dabei wurde das Verhalten des modifizierten Live-Wire Verfahrens im Hinblick auf Rauschen untersucht. Hierzu wurden die synthetisch erzeugten Bilder mit unterschiedlich starkem „Salt ‘n-Pepper“ Rauschen versehen, der Verwendung dieses Rauschens kommen hierbei jedoch keine speziellen Bedeutungen zu.

Beim Vergleich der Verfahren lagen die Schwerpunkte bei der Evaluation auf der Segmentierungsgeschwindigkeit, sowie auf der Segmentierungsgenauigkeit. Die Evaluation der Segmentierungsgenauigkeit wurde zum einen in Bezug auf Freihandsegmentierungen, zum anderen in Bezug zu Referenzsegmentierungen durchgeführt. Es wurden dabei mehrere Metriken herangezogen um die Segmentierungsergebnisse zu vergleichen. Hierzu zählten Dice-, Tanimoto-, Kappa- und Randpixel-Koeffizient, sowie Fehlerwahrscheinlichkeit, Konsistenzwerte, Hausdorff-Distanz, CCH-Wert, CCC-Wert und Vektor-Distanz.

Durch die Verwendung der beschriebenen Erweiterungen wurden sowohl bei der Segmentierungsgeschwindigkeit als auch bei der Segmentierungsgenauigkeit positive Ergebnisse erzielt. In Abb.1 ist eine Gegenüberstellung der Berechnungszeiten auf mehreren Bildern für die in [1] vorgestellte Kostenfunktion und die Berechnung über den Canny-Operator [5] dargestellt, hier sind die zeitlichen Vorteile durch die Verwendung des Canny-Operators deutlich zu sehen. Gegenüber der Freihandsegmentierung liefert das vorgestellte Verfahren wesentlich bessere Resultate, sowohl bei der Geschwindigkeit als auch bei der Qualität. Ebenso konnten bessere Ergebnisse in beiden Bereichen im Bezug zum ursprünglichen Verfahren [1] erzielt werden. Beispielsweise verringert sich die Anzahl der benötigten Saatpunkte, was in Konsequenz zu einer weiteren Beschleunigung der Segmentierungszeit führt.

6 Diskussion

Durch die Erweiterung des Live-Wire Verfahrens auf Farbbilder, sowie der zusätzlichen Verwendung des F*-Verfahrens [3, 4], kann der in [1] beschriebene interaktive Segmentierungsansatz im Hinblick auf Geschwindigkeit und Qualität der Segmentierung deutlich verbessert werden. Während die Verwendung einer mehrkanaligen Kostenfunktion zur Güte der Segmentierung auf Farbbildern beiträgt, äußert sich die Verwendung des F*-Verfahrens bei der Segmentierung hochauflöster Bilder. Speziell bei Bildern mit räumlichen Auflösungen größer als 512x512 Bildpunkten zeigt sich die Effizienz des F*-Ansatzes gegenüber des ursprünglichen Verfahrens von Dijkstra [2], da mit einer Erhöhung der Bildauflösung auch die Länge und Anzahl aller zu verwaltenden optimalen Pfade und somit die notwendige Rechen- und Speicherkapazität ansteigt.

Literaturverzeichnis

1. Mortensen EN, Morse BS, Barrett WA, et al. Adaptive Boundary Detection Using 'Live-Wire' Two-Dimensional Dynamic Programming. In: IEEE Proceedings of Computers in Cardiology (CIC '92). Durham, NC; 1992. p. 635–638.
2. Dijkstra EW. A note on two problems in connexion with graphs. In: Numerische Mathematik. vol. 1. Mathematisch Centrum, Amsterdam, The Netherlands; 1959. p. 269–271.
3. Ford Jr LR. Network Flow Theory. The RAND Corporation, Santa Monica, California; 1956.
4. Fischler MA, Tenenbaum JM, Wolf HC. Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique. Computer Graphics and Image Processing 1981;15:201–223.
5. Canny J. A Computational Approach to Edge Detection. In: Fischler MA, Firschein O, editors. Readings in Computer Vision: Issues, Problems, Principles, and Paradigms. Los Altos, CA.: Kaufmann; 1987. p. 184–203.