

Projektdokumentation

Bewegungsanalyse einer Videosequenz
mit dem Ansatz des Papers von Aach und Kunz

von

Laura Anger (Matrikelnr. 11086356)
Timo Breuer (Matrikelnr. XXXXXXXXX)
Lukas Kolhagen (Matrikelnr. 11084355)

Durchgeführt im
Master Medientechnologie
im
Sommersemester 2016

Betreuer:

Prof. Dr. Dietmar Kunz
Institut für Medien- und Phototechnik

Inhaltsverzeichnis

1	Einleitung	3
1.1	Ansatz im Paper von Aach und Kunz	3
1.2	Projektziel	3
2	Verfahren	4
2.1	Bewegungsschätzung	4
2.2	Programmablauf	5
2.3	Kostenfunktion	5
2.3.1	Datenterm	6
2.3.2	Örtliche Kohärenz	6
2.3.3	Zeitliche Kohärenz	6
2.4	Visualisierung	6
3	Auswertung	6
3.1	Testmaterial	6
3.2	Auswertung Testmaterial 1 und Testmaterial 2	8
3.3	Auswertung Testmaterial 3	8
3.4	Einschwingverhalten	9
3.4.1	Bewegungsvektorfelder	9
3.4.2	Innerhalb eines Bildes	9
3.5	Parameter der Regularisierungsterme	9
4	Zusammenfassung	10
5	Arbeitsaufteilung der Dokumentation	10

1 Einleitung

Lukas

Diese Ausarbeitung ist Teil der Abschlussprojekt-Dokumentation im Modul „Weiterführende Themen der Bildverarbeitung“ im Master Medientechnologie an der Technischen Hochschule Köln.

Das Projekt beschäftigte sich mit der Bewegungsanalyse einer Videosequenz mit dem Ansatz des Papers von Aach und Kunz^[1]. Es wurde bearbeitet von Laura Anger, Timo Breuer und Lukas Kolhagen.

1.1 Ansatz im Paper von Aach und Kunz

Die Grundlage für das vorliegende Projekt bildet das Paper „Bayesian motion estimation for temporally recursive noise reduction in X-ray fluoroscopy“. Dieses beschäftigt sich mit der Entwicklung einer robusten Methode zur Bewegungsschätzung für die speziellen Anforderungen der stark rauschenden Aufnahmen einer Röntgen-Fluoroskopie. Der Ansatz beruht auf der Modellierung drei essenzieller Faktoren:

Datenterm: Unterschied der Grauwerte zweier aufeinander folgender Bildern.

Örtliche Kohärenz: Außer an Randbereichen von Objekten, bewegen sich Nachbarschaften meist in die gleiche Richtung.

Zeitliche Kohärenz: Bewegungen verlaufen normalerweise kontinuierlich, sodass sich ein Bildblock zwischen zwei Bildern wahrscheinlich in dieselbe Richtung weiterbewegt oder die Richtung nur gering ändert.

Eine genaue Beschreibung dieser Faktoren erfolgt in [2.3](#).

1.2 Projektziel

Die Zielsetzung für das Projekt „Bewegungsanalyse einer Videosequenz“ war eine Übertragung des Ansatzes von Röntgenbildern auf normale Videosequenzen. Infolge dessen war eine Vernachlässigung der speziellen Anforderung des Bildrauschens möglich, da Röntgenbilder – insbesondere als Teil einer Fluoroskopie – zum Schutz des Patienten und des medizinischen Personals nur sehr geringe Röntgendosen enthalten dürfen und deshalb ein extrem schlechtes Signal-zu-Rauschverhältnis aufweisen. Diese Problematik besteht bei normalen Videosequenzen nicht, weshalb für die Untersuchung von vergleichsweise geringem und etwa gleich verteiltem Rauschen ausgegangen werden konnte.

2 Verfahren

Laura

In [1] wird das Verfahren, was dieser Ausarbeitung zu Grunde liegt beschrieben. Der dort aufgeführte Algorithmus wurde speziell für Röntgenaufnahmen entwickelt, um Arzt und Patienten im Rahmen von Behandlungen, welche Röntgenstrahlung verwenden vor gesundheitlichen Konsequenzen zu schützen. Es handelt sich bei dem Ansatz um ein Block-Matching Verfahren, welche dem Satz von Bayes folgen.

Im vorliegenden Fall soll das Verfahren allerdings auf gewöhnlichen Bildsequenzen angewandt werden. Im Folgenden wird das zu grundliegende Verfahren erläutert und auf eventuelle Änderungen im Algorithmus eingegangen.

2.1 Bewegungsschätzung

Laura

Um zu verstehen, wie die Bewegungsschätzung im einzelnen funktioniert, muss man sich zunächst klar machen, wie das geschätzte Bewegungsvektorfeld überhaupt definiert ist.

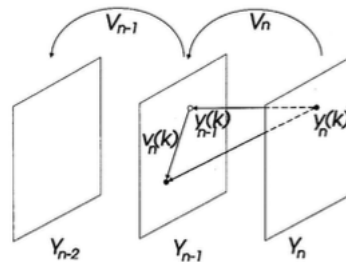


Abbildung 1: Beziehung zwischen Bewegungsvektorfeld und Einzelbildern [1]

In Abbildung 1 sieht man drei Frames einer Bildsequenz. Y_n steht hierbei für das n -te Bild dieser Sequenz und $y_n(k)$ ist der Grauwert des Pixels an der Stelle k . Das k steht für ein Koordinatenpärchen (k_x, k_y) und es gilt $k = 1, \dots, N$, wobei N die Anzahl der Pixel eines Frames ist. Der eingezeichnete Vektor $v_n(k)$ ist derjenige Vektor, der auf k addiert wird, sodass sich die Position des entsprechenden Pixels im vorherigen Bild ergibt. All diese Bewegungsvektoren zusammengenommen ergeben das Bewegungsvektorfeld eines Frames, welches in der Abbildung mit V_n bezeichnet wird.

Das geschätzte Bewegungsvektorfeld \hat{V}_n ergibt sich nach den Ausführungen von Aach und Kunz zu:

$$\hat{V}_n = \arg \max_V \left[p(Y_n, Y_{n-1}, |V) \cdot p(\hat{V}_{n-1} | V) \cdot p(V) \right] \quad (1)$$

Da es sich um einen bayesschen Ansatz handelt, gilt es sich mit bedingten Wahrscheinlichkeiten auseinander zu setzen. Der erste Faktor der Formel 1 ist im Originalansatz zur Modellierung des Rauschens gedacht. Es ist davon auszugehen, dass Röntgenbilder weit aus mehr verrauscht sind, als reale Bildsequenzen auf die das vorliegende Verfahren angewandt werden soll. Deshalb liegt es nahe, dass Rauschen weitgehend zu vernachlässigen und durch einen reinen Datenterm zu ersetzen. Mehr Details dazu kann man in Kapitel 2.3.1 nachlesen. Bei den beiden anderen Faktoren handelt es sich zum einen um die örtliche und zum anderen um die zeitliche Kohärenz. Beide werden in den Kapiteln 2.3.2 bzw. 2.3.3 näher beschrieben.

2.2 Programmablauf

Der Programmablauf des ImageJ-Plug-Ins in Form der `run()`-Methode, die aufgrund des implementierten `PlugInFilter`-Interfaces überschrieben werden muss, ist in Abbildung ?? dargestellt.

Grob betrachtet, werden in der `run()`-Methode drei `for`-Schleifen geschachtelt ausgeführt. Die innerste Schleife läuft über jeden Block des Bildes und bestimmt für diesen den Bewegungsvektor mit den geringsten Kosten aus den 14 berechneten Alternativen. Die Blockgröße ist variabel und kann z. B. 8×8 betragen.

Die nächst äußere Schleife ist für die Iterationen zuständig. In den Iterationen wird innerhalb eines Bildes des Input-Stacks das Bewegungsvektorfeld mehrfach berechnet und jeweils das zuletzt berechnete Ergebnis als Grundlage für die nächste Berechnung verwendet. Dies ist erforderlich, um die Abweichung der Bewegungsvektoren pro Frame so gering wie möglich zu halten. Gleichzeitig hängt die Wahl der Anzahl der Iterationen natürlich davon ab, welche Performanz die Berechnung erreichen soll, also wie schnell ein Ergebnis benötigt wird.

Die äußere Schleife läuft über den Stack und sorgt somit dafür, dass die Berechnungen pro Block und Iteration jeweils für jedes Bild des Input-Stacks erfolgen.

Sobald ein Bild fertig bearbeitet ist, die äußere Schleife also einen Durchlauf absolviert hat, wird der Fortschrittsbalken im ImageJ-Fenster aktualisiert, damit der Benutzer eine Rückmeldung bekommt, wie weit die Berechnung fortgeschritten ist.

Zuletzt wird in der `run()`-Methode noch das Ausgabefenster, in welchem der Input-Stack inklusive der überlagerten Bewegungsvektorfelder dargestellt wird, generiert und dessen Methode zur Anzeige aufgerufen.

2.3 Kostenfunktion

Alle

2.3.1 Datenterm

Lukas

2.3.2 Örtliche Kohärenz

Timo

2.3.3 Zeitliche Kohärenz

Lukas

2.4 Visualisierung

Laura

Um die Bewegungsvektorfelder darzustellen, wird sich mit Symbolen beholfen. Diese werden blockweise über die Videosequenz gelegt. Aus Darstellungsgründen wird das Video dazu um Faktor 4 in beide Bildrichtungen hoch skaliert. Da die einzelnen Bewegungsvektoren immer zwischen zwei Frames berechnet werden, wird genau ein Bewegungsvektorfeld weniger erzeugt, als das Eingangsmaterial Frames hat.

Bei den bereits erwähnten Symbolen handelt es sich um Kreise und Pfeile. Ein Kreis wird immer dann genutzt, wenn für den entsprechenden Block keine oder nur eine kleine Bewegung durch das Verfahren ermittelt wurde. Die Pfeile zeigen immer in Richtung der ermittelten Verschiebung. Beide Symbole haben ihren Startpunkt im Mittelpunkt des jeweiligen Blocks.

Die Randbereiche der einzelnen Bewegungsvektorfelder werden bei der Visualisierung ausgespart, da hier durch den Algorithmus begründete Fehlberechnungen auftreten.

3 Auswertung

Laura & Timo

3.1 Testmaterial

Laura

Das Verfahren wird anhand von drei unterschiedlichen Bildsequenzen getestet, die im Folgenden Testmaterial 1-3 genannt werden und alle eine Auflösung von 256x2656 Pixeln haben. Bei Testmaterial 1 und Testmaterial 2 wurde ein Bild um bekannte Werte in x- und y-Richtung verschoben. Anzumerken ist auch, dass beide Testmaterialien identisch verschoben wurden. Bei Testmaterial 3 kann die Bewegung der zu sehenden Objekte lediglich geschätzt werden. Alle Testmaterialien wurden für diese Dokumentation aufgenommen bzw. erstellt.



Abbildung 2: Standbild Testmaterial 1.

Testmaterial 1

Diese Bildsequenz besteht aus 150 Frames und wurde aus einem Bild der Stadt Köln erzeugt.

Bei den ersten und letzten 50 Frames wird ein Ausschnitt des Bildes um jeweils 10 Pixel nach links bzw. oben verschoben. In den mittleren 50 Frames wird der Bildinhalt um 10 Pixel nach links und 10 Pixel nach unten verschoben.



Abbildung 3: Standbild Testmaterial 2.

Testmaterial 2

Auf den 150 Frames, die diese Bildsequenz umfasst ist ein Blumenmotiv zu sehen, welches jeweils über 50 Frames in verschiedene Richtungen verschoben wird. Auf den ersten Frames erfolgt eine Verschiebung um 10 Pixel nach links. Während die mittleren Frames um sowohl 10 Pixel nach links, als auch 10 Pixel nach oben verschoben wurden, wurde der Bildinhalt der verbleibenden Frames nur um 10 Pixel nach oben verschoben.



Abbildung 4: Standbild Testmaterial 3.

Testmaterial 3

Bei Testmaterial 3 handelt es sich um eine reale Videosequenz, die in Köln Ehrenfeld aufgenommen wurde. Zu sehen sind ein Auto, dass sich von links nach rechts durch das Bild bewegt und eine Fahrradfahrerin, die das Bild genau entgegengesetzt durchfährt. Beide Objekte werden zeitweise durch verschiedene Objekte verdeckt. Die Kamera ist starr, weshalb für den restlichen Bildinhalt keine Bewegung zu erkennen ist. Insgesamt besteht die Testsequenz aus 118 Frames.

3.2 Auswertung Testmaterial 1 und Testmaterial 2

Laura

3.3 Auswertung Testmaterial 3

Laura

Die Bewegung im vorliegenden Testmaterial kann lediglich optisch ausgewertet werden, da die tatsächlichen Bewegungsvektoren nicht bekannt sind. Für die Bewegungsdetektion wurden die folgenden Einstellungen getroffen.

Parameter:

- ▷ 5 Iterationen
- ▷ $\nu = 1.5$
- ▷ $\lambda = 0.3$ und $\lambda_T = 0.3$

Abbildung 5 und Abbildung 6 zeigen jeweils ein Standbild aus dem Bewegungsvektorfeld, welches sich unter dem Namen *testmaterial3-bewegungsvektorfeld.tiff* auf der beiliegenden Daten-CD befindet.

Es ist deutlich zu erkennen, dass die Bereiche in denen keine Bewegung stattfindet weitgehend richtig detektiert wurden. Und auch die Bewegung des Autos und der Radfahrerin werden trotz der Verdeckung durch die Säule weitgehend erkannt. Auffällig ist, dass besonders die Blöcke, die hinter dem Auto liegen und deren Bildinhalt sich nicht mehr bewegt falsch berechnet sind. Das ist besonders gut in Abbildung 6 zu sehen und

lässt sich auf das Einschwingverhalten zurückführen. Dieses wird in Kapitel 3.4 erklärt und ausgewertet.

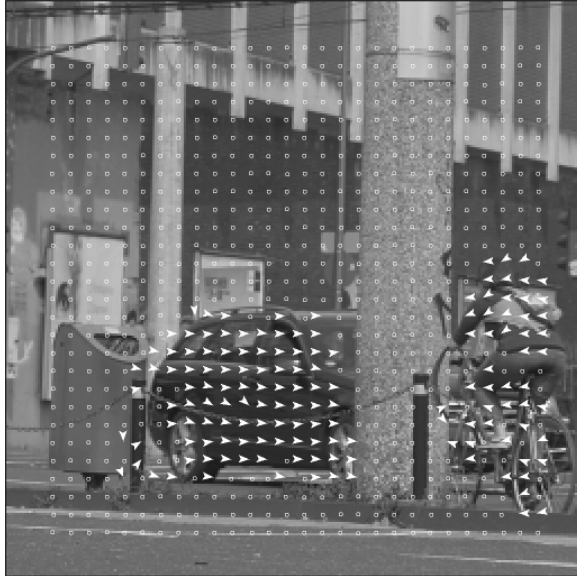


Abbildung 5: Bewegungsvektorfeld Testmaterial 3. Frame 43.



Abbildung 6: Bewegungsvektorfeld Testmaterial 3. Frame 47.

3.4 Einschwingverhalten

Als Ausgangspunkt werden alle Bewegungsvektorfelder zu Beginn des Algorithmus auf den Nullvektor gesetzt. Das Verfahren muss sich also zunächst einschwingen. Dabei wird im Folgenden zwischen dem interframe und dem intraframe Einschwingen unterschieden.

3.4.1 Bewegungsvektorfelder

Timo

3.4.2 Innerhalb eines Bildes

Laura

3.5 Parameter der Regularisierungsterme

Timo

4 Zusammenfassung

Lukas

5 Arbeitsaufteilung der Dokumentation

Literatur

- [1] T. Aach and D. Kunz. Bayesian motion estimation for temporally recursive noise reduction in x-ray fluoroscopy. *Philips Journal of Research*, 51(2):231–251, 1998.