

Projektdokumentation

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

von

Laura Anger	(Matrikelnr. 11086356)
Timo Breuer	(Matrikelnr. 11089822)
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	3
2.1	Bewegungsschätzung	4
2.2	Kostenfunktion	5
2.2.1	Datenterm	5
2.2.2	Örtliche Kohärenz	6
2.2.3	Zeitliche Kohärenz	7
2.3	Programmablauf	7
2.4	Visualisierung	8
3	Auswertung	8
3.1	Testmaterial	8
3.2	Auswertung der Testmaterialien	11
3.3	Einschwingverhalten	13
3.3.1	Bewegungsvektorfelder	13
3.3.2	Innerhalb eines Bildes	14
3.4	Parameter der Regularisierungsterme	16
4	Zusammenfassung	17
5	Arbeitsaufteilung	18

1 Einleitung

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 [2.2.1](#), [2.2.2](#) und [2.2.3](#).

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

In [\[1\]](#) wird das Verfahren, was dieser Ausarbeitung zu Grunde liegt beschrieben. Es handelt sich dabei um ein Block-Matching-Verfahren auf Grundlage des Satzes von

Bayes. Im Folgenden wird das Verfahren erläutert und auf die Änderungen im Algorithmus eingegangen, welche erfolgt sind, um das Verfahren für normale Bildsequenzen anzupassen.

2.1 Bewegungsschätzung

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

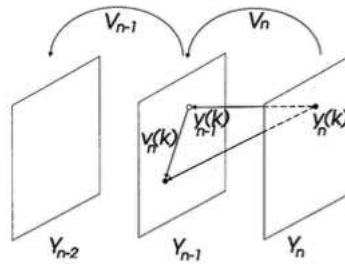


Abbildung 1: Beziehung zwischen Bewegungsvektorfeld und Einzelbildern [1].

In Abbildung 1 sieht man drei Bilder 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 Bildes 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 Bildes, 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, wie bereits erwähnt, 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, das Rauschen weitgehend zu vernachlässigen und durch einen reinen Datenterm zu ersetzen. Mehr Details dazu kann man in Kapitel 2.2.1 nachlesen. Die beiden anderen Faktoren können

zum einen zur örtlichen und zum anderen zur zeitlichen Kohärenz vereinfacht werden. Diese Faktoren werden in den Kapiteln 2.2.2 bzw. 2.2.3 näher beschrieben.

2.2 Kostenfunktion

Nach Aach und Kunz^[1] lässt sich die unter 2 erwähnte Wahrscheinlichkeitsdichte formulieren als:

$$ce^{-\sum_{\vec{k}} \left(\frac{[d(\vec{k}, \vec{v}(\vec{k}))]^2}{4 \cdot \sigma^2(\vec{k})} + \lambda \sum_{\vec{l} \in N(\vec{k})} g_l \|\vec{v}(\vec{k}) - \vec{v}(\vec{l})\|_2^\nu + \lambda_T \sum_{\vec{k}} \left\| \hat{\vec{V}}_{n-1}(\vec{k}) - \vec{v}(\vec{k}) \right\|_2^\nu \right)} \quad (2)$$

Der Ausdruck wird maximal, wenn der negative Exponent minimal wird. Ziel der Aufgabe ist daher die Minimierung einer Kostenfunktion, deren Komponenten im folgenden Kapitel erläutert werden sollen.

Die Kostenfunktion lässt auf die wahrscheinlichste Bewegungsrichtung schätzen. Zur gezielten Untersuchung werden 14 alternative Vektorkandidaten, die sich nach [1] ergeben, betrachtet.

Die Implementierung bestimmt auf Basis der aktuell untersuchten Bildposition die 14 alternativen Kandidaten für die Bewegungsschätzung. Die ermittelten Vektorkandidaten werden hierzu in einer Liste gehalten.

2.2.1 Datenterm

Der Datenterm als Teil der zu minimierenden Kostenfunktion hat die Aufgabe, die Veränderung der Grauwerte beim Vergleich zweier Bilder zu quantifizieren. Da angenommen wird, dass diese gemessenen Veränderungen nur durch Bewegung und Rauschen verursacht wird, lässt sich das Rauschen selbst modellieren, indem vor dem Vergleich der Bilder die tatsächliche Bewegung herausgerechnet, also eins der Bilder entsprechend des zuvor ermittelten Bewegungsvektors verschoben wird.

Der zu minimierende Ausdruck für die Modellierung des Rauschens für ganze Bilder lautet:

$$p(Y_n, Y_{n-1}|V) = c' \cdot e^{-\sum_{\vec{k}} \frac{[d(\vec{k}, \vec{v}(\vec{k}))]^2}{4 \cdot \sigma^2(\vec{k})}} \quad (3)$$

Da ebenfalls angenommen wird, dass die Bewegung über das ganze Bild nicht konstant ist, wird das Bild in Blöcke unterteilt. Des weiteren führt die Übertragung des eigentlichen Problems von Röntgenbildern auf normale Bildsequenzen zu der Erkenntnis, dass die Poisson-Verteilung des Rauschens bei Röntgenbildern aufgrund der niedrigen Anzahl von Röntgenquanten, bei normalen Bildern in eine Normalverteilung übergeht.

Aus diesem Grund kann der Term unter dem Bruchstrich im Exponenten von Gleichung 3 vernachlässigt werden. Da der verbleibende Ausdruck von Gleichung 3 minimal wird, wenn der Exponent der Exponentialfunktion minimal wird, genügt für das Minimierungsproblem

$$d\left(\vec{k}, \vec{v}\left(\vec{k}\right)\right) = y_n\left(\vec{k}\right) - y_{n-1}\left(\vec{k} + \vec{v}\left(\vec{k}\right)\right) \quad (4)$$

wobei $y_n\left(\vec{k}\right)$ der aktuelle Block und $y_{n-1}\left(\vec{k} + \vec{v}\left(\vec{k}\right)\right)$ der zu vergleichende Block des vorherigen Bildes, verschoben um $\vec{v}\left(\vec{k}\right)$ ist.

Um etwaige Probleme am Rand des Bildes zu verhindern, wurde eine Randbehandlung implementiert, welche verhindert, dass Pixeladressen im vorherigen (y_{n-1}) Bild außerhalb der Bildgrenzen liegen.

2.2.2 Örtliche Kohärenz

Die örtliche Kohärenz trägt als gewichteter Regularisierungsterm zur Kostenfunktion bei. Bei der Bestimmung der örtlichen Kohärenz wird von der Annahme ausgegangen, dass benachbarte Regionen in Bildfolgen ähnliche Bewegungsrichtungen aufweisen. Hierzu werden neben dem neuen möglichen Bewegungsvektor der untersuchten Bildposition die Bewegungsvektoren der jeweiligen Achter-Nachbarschaft betrachtet. Über die Differenzen des Bewegungsvektor und der benachbarten Bewegungsvektoren ergibt sich eine a-priori-Wahrscheinlichkeitsdichte. Ziel ist daher die Minimierung des Exponenten der Wahrscheinlichkeitsdichte:

$$p(V) = ce^{-\lambda E(V)} = c \prod_{\vec{k}} e^{-\lambda E_L(\vec{k}, V)} \quad (5)$$

mit

$$E_L\left(\vec{k}, V\right) = \sum_{\vec{l} \in N(\vec{k})} g_{\vec{l}} \left\| \vec{v}\left(\vec{k}\right) - \vec{v}\left(\vec{l}\right) \right\|_2^\nu \quad (6)$$

Der Faktor $g_{\vec{l}}$ gewichtet die normierte Differenz abhängig von der Position des betrachteten benachbarten Bewegungsvektor. Über den Parameter ν kann die verallgemeinerte Normalverteilung angepasst werden. Die Normalverteilung ($\nu = 2$) glättet das Bewegungsvektorfeld zu stark. Für $\nu = 1.3$ ergibt sich eine praktikable Parametrisierung.^[1]

Die Implementierung berechnet demnach die Summe bestehend aus den gewichteten, normierten und mit dem Parameter ν potenzierten Differenzen. Hierzu wird neben dem Index der aktuell untersuchten Bildposition/-region der Listenindex des neuen

möglichen Bewegungsvektors benötigt. Mit Hilfe des Index der untersuchten Bildposition können dann die benötigten Positionen der Nachbarregionen bestimmt werden. Zu beachten ist hierbei, dass für Randbereiche weniger Nachbarregionen untersucht werden bzw. nicht vorhanden sind. Nachdem die Differenzen gemäß dem euklidischen Abstand normiert sind, werden diese mit dem Parameter ν potenziert. Das Ergebnis wird für horizontale und vertikale Nachbarregionen mit dem Faktor $g_l = 1$ und für diagonale Nachbarregionen mit dem Faktor $g_l = 0.5$ gewichtet. Anschließend werden die gewichteten Ergebnisse aufsummiert und zurück gegeben.

2.2.3 Zeitliche Kohärenz

Da Bewegungen normalerweise gleichmäßig ablaufen – wenn sie nicht gerade beginnen oder enden – basiert die zeitliche Kohärenz darauf, dass die Bewegung (z. B. im aktuellen Block) der Bewegung im gleichen Block des vorherigen Bildes ähnlich ist. Für die Berechnung der Kosten folgt als Minimierungsterm aus

$$p(\hat{V}_{n-1}|V) = c \cdot e^{-\lambda_T \sum_{\vec{k}} \left\| \hat{V}_{n-1}(\vec{k}) - \vec{v}(\vec{k}) \right\|_2^\nu} \quad (7)$$

dass wieder nur der Exponent der Exponentialfunktion ausgewertet werden muss, da der gesamte Ausdruck minimal wird, wenn der Exponent minimal wird.

In (7) ist $\hat{V}_{n-1}(\vec{k})$ das geschätzte Bewegungsvektorfeld des vorherigen Bildes und $\vec{v}(\vec{k})$ der Bewegungsvektor des aktuellen Blocks, welcher zur Position des Blocks im vorherigen Bildes führt. Auf λ_T und ν wurde bereits im vorigen Abschnitt 2.2.2 eingegangen.

2.3 Programmablauf

Der Programmablauf des ImageJ-Plug-Ins in Form der `run()`-Methode, die aufgrund des implementierten `PlugInFilter`-Interfaces überschrieben werden muss, ist in Abbildung 2 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 Bild 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.

2.4 Visualisierung

Um die Bewegungsvektorfelder V_n 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 Bildern berechnet werden, wird genau ein Bewegungsvektorfeld weniger erzeugt, als das Eingangsmaterial Bilder 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 können.

3 Auswertung

3.1 Testmaterial

Das Verfahren wird anhand von drei unterschiedlichen Bildsequenzen getestet, die im Folgenden Testmaterial 1–3 genannt werden und alle eine Auflösung von 256×256 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.

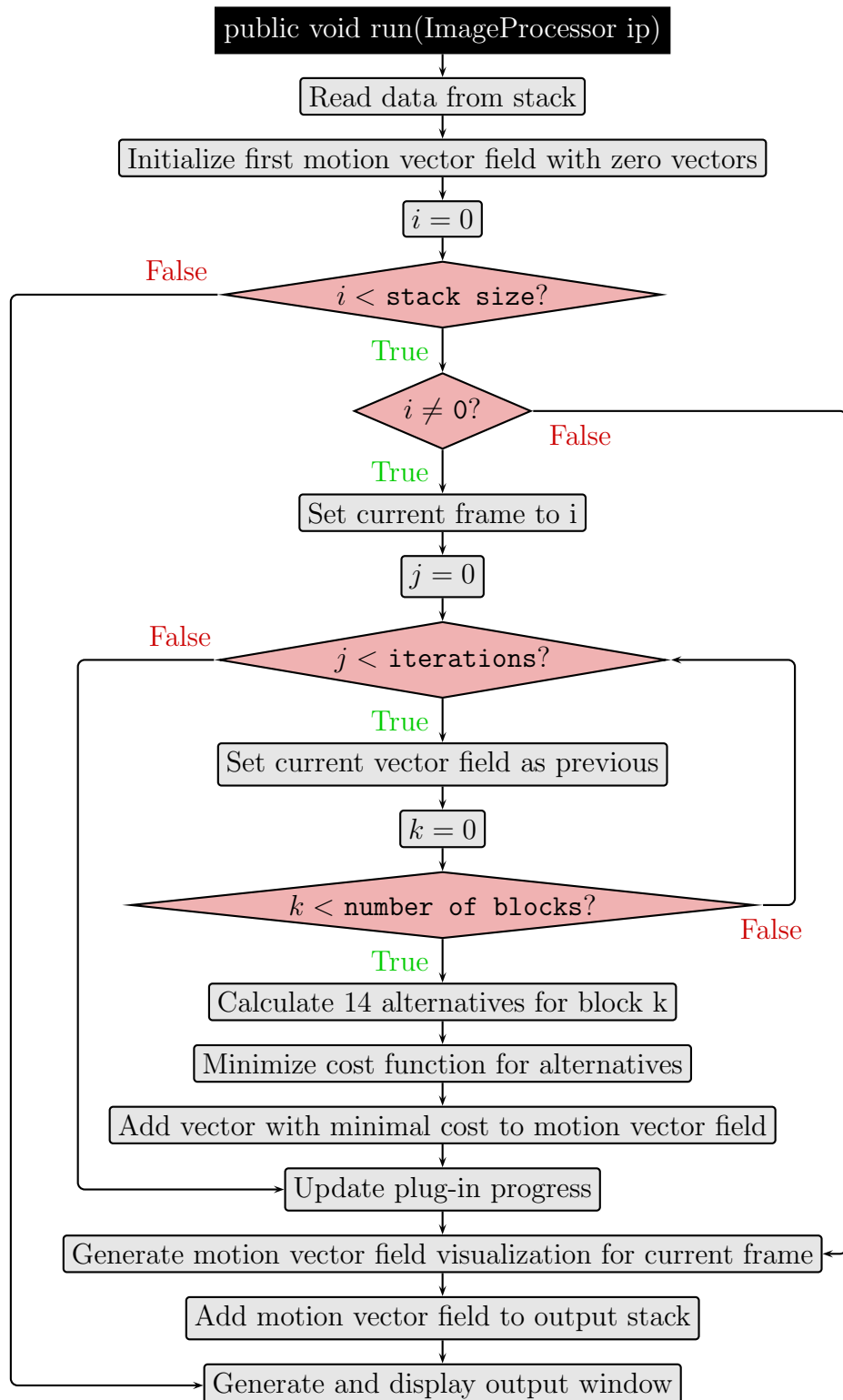


Abbildung 2: Programmablauf



Abbildung 3: Standbild Testmaterial 1.

Testmaterial 1

Diese Bildsequenz besteht aus 150 Bildern und wurde aus einem Standbild der Stadt Köln erzeugt.

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



Abbildung 4: Standbild Testmaterial 2.

Testmaterial 2

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



Abbildung 5: 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 Bildern.

3.2 Auswertung der Testmaterialien

Im Folgenden wird das Verfahren anhand der in Kapitel 3.1 aufgeführten Testmaterialien ausgewertet. Dabei wurden für alle Materialien die gleichen Einstellungsparameter gewählt:

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

In Abbildung 6 sind jeweils drei Bewegungsvektorfelder für Testmaterial 1 und Testmaterial 2 zu sehen. Es ist deutlich erkennbar, dass die Bewegungsdetektion sowohl für hochfrequente, als auch für tieffrequente Bildinhalte funktioniert hat.

Die Abweichungen der berechneten von den tatsächlichen Bewegungsvektoren können bestimmt werden, da die Bewegung bekannt ist. Da sich alle Blöcke eines Bildes mit der selben Geschwindigkeit in die selbe Richtung bewegen, kann die absolute Längenabweichung der einzelnen Vektoren über ein Bild aufsummiert und durch die Anzahl der Blöcke geteilt werden um ein Maß für die absolute Abweichung zu erhalten. Schaut man sich die so ermittelten Abweichungen für ein Bild an, wo in den vorherigen Bildern keine Richtungsänderung stattgefunden hat, so beträgt diese im Durchschnitt für beide Testmaterialien ungefähr ein halbes Pixel. Da es wie in Kapitel 3.3 näher beschrieben dauert, bis die Bewegungsvektoren eingeschwungen sind, kommt es in den Bildern 50 und 100 in beiden Testmaterialien zu einer Abweichung von bis zu 8 Pixeln. Das sind eben die Bilder, wo sich die Bewegung plötzlich ändert.

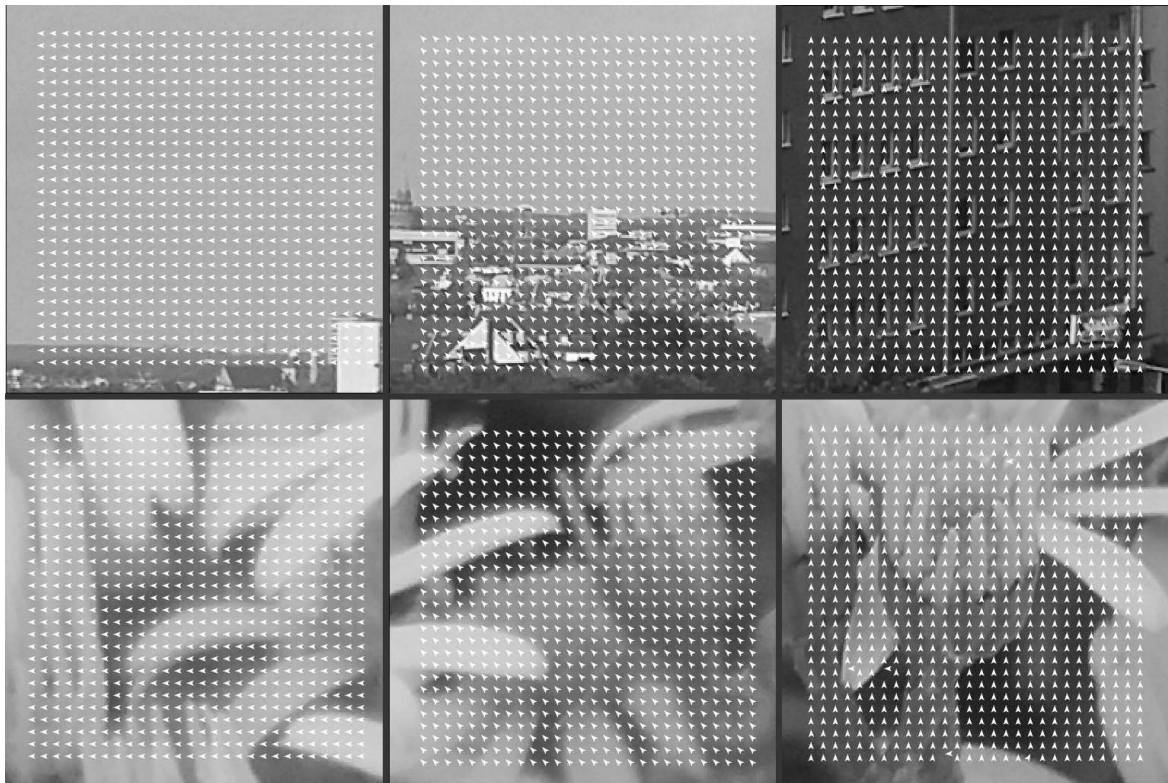


Abbildung 6: Bewegungsvektorfelder (v.l.n.r.): Bild 10, Bild 60 und Bild 110.
Oben Testmaterial 1 und unten Testmaterial 2.

Da es sich bei Testmaterial 3 um eine reale Videosequenz handelt, deren Bewegungsvektoren unbekannt sind, kann die Auswertung hier nur rein visuell vorgenommen werden.

Abbildung 7 und Abbildung 8 zeigen jeweils ein Standbild aus dem Bewegungsvektorfeld.

Es ist deutlich zu erkennen, dass die Bereiche in denen keine Bewegung stattfindet weitgehend richtig detektiert wurden. Und auch die Bewegungen 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 8 zu sehen und lässt sich auf das Einschwingverhalten zurückführen. Dieses wird in Kapitel 3.3 erklärt und ausgewertet.

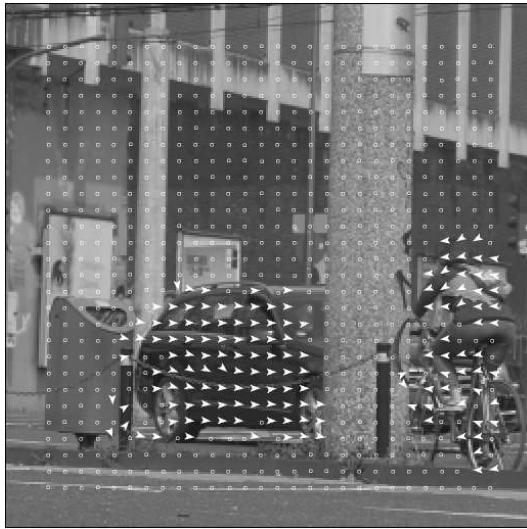


Abbildung 7: Bewegungsvektorfeld
Testmaterial 3. Bild 43.



Abbildung 8: Bewegungsvektorfeld
Testmaterial 3. Bild 47.

Alle erzeugten Bewegungsvektorfelder können auf der beiliegenden Daten-CD eingesehen werden. Die Dateien heißen *bvf-1.tif*, *bvf-2.tif* und *bvf-3.tif*.

3.3 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 Einschwingen (über mehrere Bilder hinweg) und dem intraframe Einschwingen (innerhalb eines Bildes) unterschieden.

3.3.1 Bewegungsvektorfelder

Im Folgenden wird das Konvergenzverhalten des Verfahrens über die Bilderfolgen untersucht. Zur Auswertung wird das *Testmaterial 1* und das *Testmaterial 2* benutzt. Die Parametereinstellungen werden wie folgt festgelegt:

- 5 Iterationen
- $\nu = 1.3$
- $\lambda = 0.001$ und $\lambda_T = 0.001$
- Bewegungsvektor $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 10 \\ 0 \end{pmatrix}$

Um das Konvergenzverhalten des Verfahrens zu untersuchen wird die Abweichung des geschätzten Bewegungsvektorfeldes zum realen (und bekannten) Bewegungsvektorfeld

über die Bildfolge bestimmt. Zum Bemessen der Abweichung wird der relative Fehler der einzelnen Vektoren ermittelt, aufsummiert und anschließend durch die Gesamtanzahl der Bewegungsvektoren gemittelt. Abbildung 9 veranschaulicht das Konvergenzverhalten der zwei Bildfolgen. Spätestens nach der siebten Bildfolge konvergiert das Verfahren. Neben den Einstellungen für ν , λ und λ_T wird das Konvergenzverhalten durch den Bildinhalt des Testmaterials beeinflusst.

Abweichung [%]

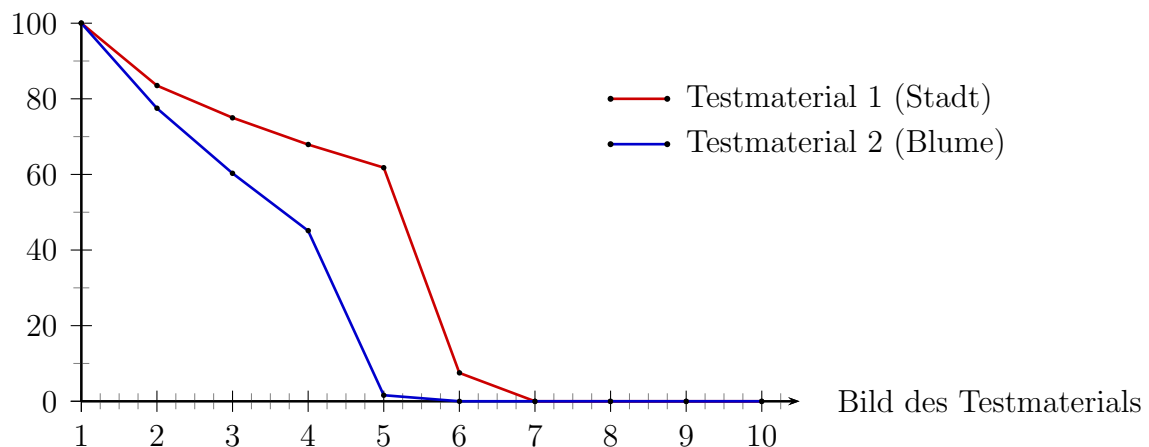


Abbildung 9: Einschwingverhalten der Bewegungsvektorfelder über die ersten 10 Bilder von Testmaterial 1 und 2.

3.3.2 Innerhalb eines Bildes

Wie in Kapitel 2.3 beschrieben wird die Bewegungsdetektion in jedem Bild über mehrere Iterationen hinweg ermittelt. Mit anderen Worten: V_n wird in jeder Iteration neu gesetzt. Im Folgenden wird die Veränderung der Vektoren nach jeder Iteration betrachtet. Dabei werden die Parameter wie in Kapitel 3.3.1 gewählt.

In Abbildung 10 werden die prozentualen Veränderungen der ersten sechs Bewegungsvektorfelder über 5 Iterationen hinweg aufgetragen. Es ist deutlich zu erkennen, dass bei den Bildern 1-4 nur eine kleine Veränderung der Bewegungsvektoren vorliegt. In Bild 5 geschieht die signifikante Änderung des Bewegungsvektorfeldes und somit die eigentliche Einschwingung innerhalb des Bildes. In Bild 6 findet kaum noch eine Änderung statt. Dies entspricht Abbildung 9, die das Einschwingverhalten über die einzelnen Bilder für 5 Iterationen zeigt.

Abweichung [%]

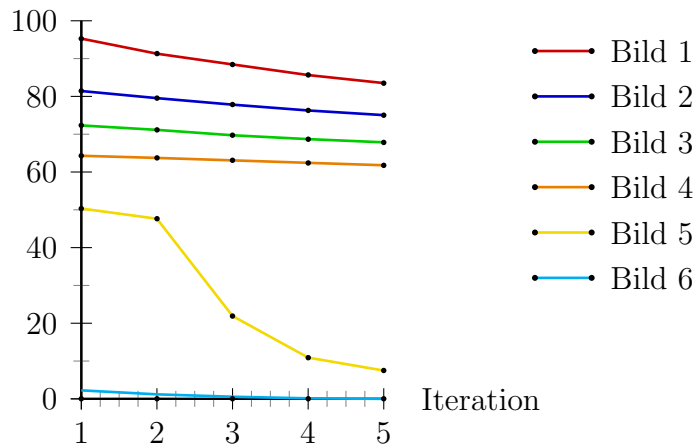


Abbildung 10: Einschwingverhalten der Bewegungsvektorfelder für Testmaterial 1 in Abhängigkeit der Iterationen innerhalb eines Bildes bei fünf Iterationen.

Während Abbildung 10 das bildweise Einschwingen für 5 Iterationen zeigt, wird in Abbildung 11 die prozentuale Abweichung der Bewegungsvektoren innerhalb eines Bildes für 10 Iterationen dargestellt.

Die Verdopplung der Iterationsschritte scheint dazu zu führen, dass bereits im dritten Bewegungsvektorfeld die Änderung gegen 0 geht und die Einschwingung bei einer gleichförmigen Bewegung wie in Testmaterial 1 zu sehen, beendet ist.

Abweichung [%]

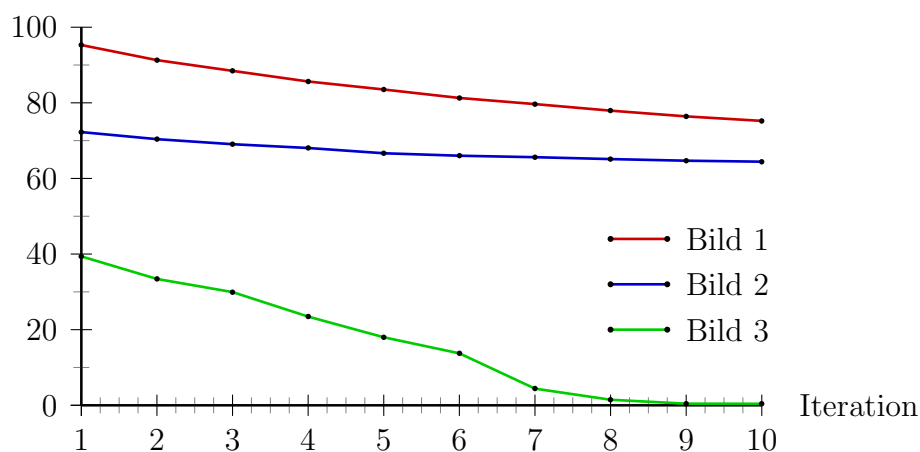


Abbildung 11: Einschwingverhalten der Bewegungsvektorfelder für Testmaterial 1 in Abhängigkeit der Iterationen innerhalb eines Bildes bei zehn Iterationen.

3.4 Parameter der Regularisierungsterme

Der folgende Abschnitt untersucht die geeignete Parametrisierung der Gewichtungen λ und λ_T der zwei Regularisierungsterme. Wie in [1] beschrieben, ist es möglich über die zwei Faktoren die Varianz der verallgemeinerten Normalverteilung zu modifizieren. Eine geeignete Parametrisierung kann demnach das Konvergenzverhalten des Verfahrens positiv beeinflussen.

Die Untersuchungen werden auf Basis von *Testmaterial 1* durchgeführt. Es gilt $\nu = 1.3$. Der Bewegungsvektor ist $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 10 \\ 0 \end{pmatrix}$. In Tabelle 1 wird das Einschwingen der ersten 10 Bewegungsvektorfelder unter Variation von λ und λ_T betrachtet. Zur Bestimmung der Abweichung des geschätzten Bewegungsvektorfeldes wird wie in 3.3.1 verfahren.

Abweichung [%]

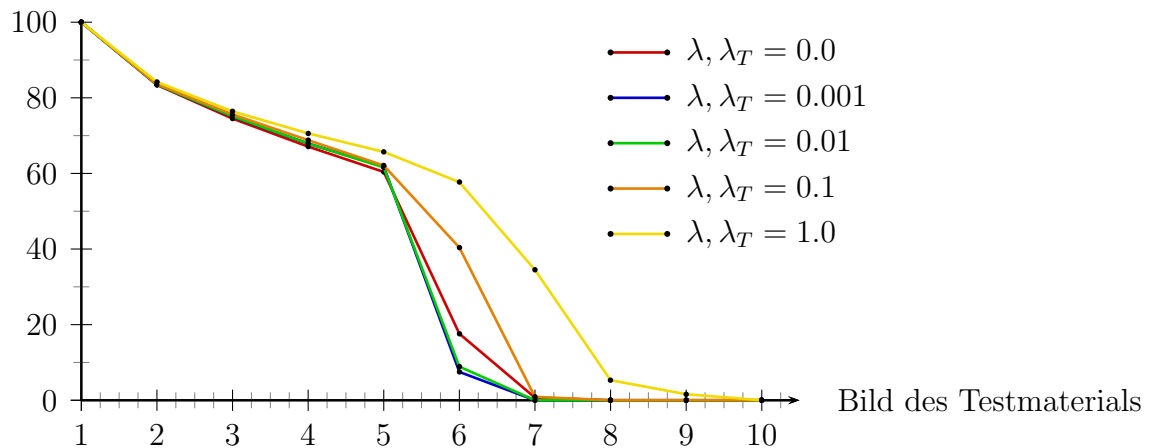


Abbildung 12: Einschwingverhalten der Bewegungsvektorfelder in Abhängigkeit der Regularisierungsparameter λ und λ_T .

Bild	Abweichung für $\lambda, \lambda_T = \dots$				
	0.0	0.001	0.01	0.1	1.0
1	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %
2	83.4 %	83.5 %	83.5 %	83.6 %	84.2 %
3	74.5 %	75.0 %	75.1 %	75.6 %	76.4 %
4	67.1 %	67.9 %	67.9 %	68.8 %	70.6 %
5	60.4 %	61.8 %	61.6 %	62.1 %	65.7 %
6	17.6 %	7.5 %	8.9 %	40.4 %	57.7 %
7	0.7 %	0.0 %	0.0 %	0.9 %	34.5 %
8	0.0 %	0.0 %	0.0 %	0.0 %	5.3 %
9	0.0 %	0.0 %	0.0 %	0.0 %	1.6 %
10	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %

Tabelle 1: Werte zu Abbildung 12.

Sowohl für λ als auch für λ_T werden dieselben Größen eingestellt. Nach spätestens 10 Bildfolgen konvergiert das Verfahren für jede der betrachteten Parametrisierungen. Für die Werte 0.001 und 0.01 konvergiert das Verfahren bereits nach sieben Bildfolgen vollständig. Zu beachten ist hierbei, dass für $\lambda = 0.001$ und $\lambda_T = 0.001$ die Abweichung des geschätzten Bewegungsvektorfeldes bereits nach der sechsten Bildfolge am geringsten ist (ca. 7.5 %). Wie Tabelle 1 zu entnehmen ist, nimmt für größere Werte das Konvergenzverhalten ab. Die Wahl von $\lambda = 0.001$ und $\lambda_T = 0.001$ liefert das Optimum für die betrachteten Einstellungen.

4 Zusammenfassung

Das Projekt, den Ansatz von Aach und Kunz^[1] auf normale Bildsequenzen zu übertragen und so eine robuste Methode zur Bewegungsschätzung von Realvideo-Sequenzen zu erhalten, konnte erfolgreich abgeschlossen werden.

Es wurde die Algorithmik an die geänderten Daten angepasst und in diesem Zuge das spezielle Rauschmodell für Röntgenaufnahmen vernachlässigt.

Für die Auswertung wurden drei verschiedene Testmaterialien erstellt, von denen zwei lediglich Standbilder enthalten, welche über den Bildausschnitt über die Zeit verschoben werden, sodass die tatsächliche Bewegung im Bild bekannt ist und als Grundwahrheit angenommen werden kann. Die verbleibende Testsequenz wurde ausgewählt, da diese Bewegungen von Realobjekten in verschiedenen Bewegungen, eine statische Kameraposition und zeitweise Verdeckungen der beiden Bewegungen bot. Die Auswertung anhand des Testmaterials hat gezeigt, dass die Bewegungsschätzung hinreichend gut für die verwendeten Testmaterialien funktioniert.

Des Weiteren belegt die Auswertung der Einschwingverhalten über die Iterationen pro Bild, dass – ähnlich wie im Paper von Aach und Kunz^[1] – eine signifikante Verbesserung der Ergebnisse über fünf bis zehn Iterationen erreicht werden kann.

Es stellt sich auch heraus, dass es zu Beginn einer Bildsequenz einige Bilder dauert,

bis sich der Fehler der Bewegungsschätzung auf ein akzeptables Niveau gesunken ist.

Insgesamt bewegen sich die Ergebnisse der Untersuchung in einem erwartbaren Bereich, da bei einer Anpassung des Verfahrens von Röntgen- auf normale Bildsequenzen keine starken Abweichungen eintreten sollten.

5 Arbeitsaufteilung

Die Implementierung wurde gemeinschaftlich und gleichverteilt bearbeitet. Die Aufteilung der Dokumentation gliedert sich wie folgt:

Anger: Abschnitte [2.1](#), [2.4](#), [3.1](#), [3.2](#) und [3.3.2](#).

Breuer: Abschnitte [2.2](#), [2.2.2](#), [3.3.1](#) und [3.4](#)

Kolhagen: Abschnitte [1](#), [2.2.1](#), [2.2.3](#), [2.3](#) und [4](#).

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.