

Folien “Video Processing and Communications” Two-Dimensional Motion Estimation

Ergänzende Kommentare

Die Folie “**Outline**” gibt einen Überblick über die behandelten Themen. Das “**2D Motion Model**” (2. Punkt unter “Outline”) wird auf der **Folie 3** näher betrachtet. Dazu gehören Überlegungen zum *Kameramodell* mit den zugehörigen Abbildungseigenschaften (Punkt 1 auf Folie 3), welche auf den Folien 4, 5 und 6 gezeigt sind. Damit im Zusammenhang stehen auch die typischen Kamerabewegungen, welche auf Folie 12 gezeigt sind. Die *3D Bewegung* (Punkt 2 auf Folie 3 bzw. 1. Punkt unter “Outline”) wird auf Folien 7 und 8 näher betrachtet. Die *Projektion der 3D Bewegung* (Punkt 3 auf Folie 3) auf die 2D Bildebene wird auf Folie 9 skizziert. Beispiele für die 2D Motion Vektoren (MV) von Folie 9 sind auf Folie 10 gegeben. Folie 11 handelt von Verdeckungen, ein Spezialproblem, welches bei der Projektion von 3D auf 2D auftritt. *2D Bewegungen*, die als Folge der 3D Bewegung eines *starrten Objekts* auftreten (Punkt 4 auf Folie 3), sowie die zugehörige *projektive Abbildung* werden auf den Folien 13, 14 und 15 behandelt. Die Approximation der projektiven Abbildung durch eine affine oder bilineare Abbildung (Punkt 5 auf Folie 3) wird auf den Folien 16 und 17 behandelt.

Das Thema “**2D Motion vs. Optical Flow**” (3. Punkt unter “Outline”) wird auf Folie 18 behandelt.

Das Thema “**Optical Flow Equation and Ambiguity in Motion Estimation**” (4. Punkt unter “Outline”) wird auf den Folien 19 und 20 behandelt.

Das Thema “**Motion Representation**” (5. Punkt unter “Outline”) wird auf Folie 21 behandelt.

Das Thema “**Motion Estimation Techniques**” (6. Punkt unter “Outline”) wird ab Folie 22 behandelt. Blockbasierte Verfahren werden von Folie 22 bis 31 behandelt. Die letzten drei Folien davon (29, 30 und 31) behandeln die hierarchische Suche. Folien 23-28 behandeln die “exhaustive search” (vollständige Suche). Während das Pseudo-Code Beispiel auf Folie 24 die Suche mit ganzzahliger Pixelgenauigkeit zeigt, wird auf Folien 25, 26, 27 und 28 die Suche mit Subpixel-Genauigkeit (fractional pixel accuracy) illustriert.

Folie 4 zeigt die Abbildung eines Punktes X im 3D Raum auf den Bildpunkt x in der 2D Bildebene. C ist der Kameramittelpunkt, F die Brennweite. Es wird angenommen, dass der Ursprung des 3D (Welt-)koordinatensystems im Kameramittelpunkt C liegt. Als Kameramodell wird das (einfache) Modell einer Lochkamera (*pinhole*) verwendet. Zusätzliche Effekte, wie z.B. Linsenverzerrung, sind bei diesem Modell nicht berücksichtigt.

Folie 5 illustriert die perspektive Projektion an Hand des Lochkamera-Modells. Zur übersichtlicheren Darstellung ist hier, im Gegensatz zu Folie 4, die Kamera-Bildebene

auf der selben Seite wie das 3D Objekt eingezeichnet, wodurch die auf Folie 4 entstehende "Invertierung" des Bildes vermieden wird. Die Formeln geben die Zusammenhänge zwischen 3D Objektkoordinaten und 2D Bildkoordinaten an, wobei die Brennweite F als Skalierungsfaktor eingeht. Die Z -Koordinate entspricht der Entfernung des Objekts von der Kamera.

Auf **Folie 6** ist, im Gegensatz zum Lochkamera-Model mit der zugehörigen perspektiven Projektion aus Folie 5, die orthografische Projektion ("Parallelprojektion") gezeigt, welche unter bestimmten Voraussetzungen als Näherung verwendet werden kann.

Folie 7 illustriert, dass eine beliebige 3D Bewegung eines starren Objekts durch eine Hintereinanderanwendung einer Rotation R und einer Translation T dargestellt werden kann. Das Objekt wird vor Durchführung der Rotation in den Mittelpunkt des Koordinatensystems verschoben (C) und danach wieder zurückverschoben. Die Rotation (bzw. Rotationsmatrix) ist durch die Angabe der Rotationswinkel um die 3 Koordinatenachsen ($\theta_x, \theta_y, \theta_z$) eindeutig bestimmt.

Folie 8 nennt 2 Möglichkeiten, die Bewegung von "nicht-starren" (d.h. verformbaren) Objekten zu beschreiben: (a) Man unterteilt das Gesamtobjekt in mehrere starre Teilobjekte, die miteinander verbunden sind (z.B. Unterteilung des menschlichen Körpers in Körpersegmente). (b) Man beschreibt die Gesamtbewegung durch eine globale Bewegung plus einer überlagerten lokalen Bewegung in Teilbereichen (z.B. Gesamtbewegung eines Schiffs mit überlagerter Bewegung des Segels).

Folie 9 illustriert den Zusammenhang zwischen dem 3D Motion Vektor (MV) \mathbf{D} und dem 2D Motion Vektor \mathbf{d} .

Folie 10 zeigt die Zusammenfassung der einzelnen Bewegungsvektoren in einem Bewegungsfeld (Motion Field). Jeder einzelne Bewegungsvektor besitzt einen Betrag und eine Richtung. Für die grafische Darstellung wird der Betrag zur besseren Sichtbarkeit häufig skaliert.

Folie 11 illustriert das Problem der Verdeckungen. Für verdeckte Gebiete können die Motion Vektoren nicht bestimmt werden.

Folie 12 zeigt verschiedene 3D Kamerabewegungen. Nicht gezeigt ist in dieser Abbildung das Zoom, d.h. die Veränderung der Brennweite.

Folie 13 zeigt die 2D Bewegungsfelder zu 2 typischen Kamerabewegungen.

Auf **Folie 14** wird die formelmäßige Darstellung einer 2D Bewegung, welche durch eine allgemeine 3D Bewegung eines starren Objekts entstanden ist (diese lässt sich ja durch Angabe einer Rotationsmatrix R und eines Translationsvektors T darstellen – siehe Folie 7), abgeleitet. Die Ausgangsformel gibt den Zusammenhang zwischen den 3D Koordinaten (X, Y, Z) vor der Objektbewegung und den 3D Koordinaten (X', Y', Z') nach der Objektbewegung an. In die Ausgangsformel wird die Formel von Folie 5 eingesetzt, welche den Zusammenhang zwischen Objekt- und Bildkoordinaten unter Berücksichtigung der Brennweite F angibt. Dadurch erhält man den formelmäßigen Zusammenhang zwischen den Bildkoordinaten (x, y) vor der Objektbewegung und den

Bildkoordinaten (x', y') nach der Objektbewegung. Für den Spezialfall, dass das betrachtete 3D Objekt eine planare Oberfläche aufweist, erhält man durch Einsetzen der Ebenengleichung die Formel für die *Projektive Abbildung*, welche in der Praxis eine wichtige Rolle spielt. In praktischen Anwendungen versucht man häufig, gekrümmte Oberflächen zunächst durch Ebenen anzunähern, um dann die projektive Abbildung anwenden zu können.

Auf **Folie 15** sind die Auswirkungen der projektiven Abbildung sowie einiger gebräuchlicher Näherungen illustriert. Zwei charakteristische Eigenschaften der projektiven Abbildung sind (a) der “chirping effect” und (b) der “converging effect” (auch “Keystone effect” genannt). Aus den Abbildungen kann man erkennen, dass die affine und bilineare Abbildung den chirping effect nicht nachbilden: Die beobachtete räumliche Frequenz – siehe Abstand der Fenster - bleibt gleich, unabhängig von der Entfernung vom Betrachter. Der Keystone-Effekt (d.h. parallele Geraden treffen sich im Endlichen) wird von der bilinearen Abbildung erfasst, nicht jedoch von der affinen Abbildung (siehe Linien der oberen und unteren Fensterreihe). Weiters lässt sich beobachten, dass bei der bilinearen Näherung (ebenso wie bei den drei Näherungen, die ganz rechts dargestellt sind) nicht-achsenparallele Geraden im Allgemeinen nicht mehr auf Geraden abgebildet werden, sondern gekrümmt erscheinen.

Folie 16 zeigt, dass die affine Abbildung durch 6 und die bilineare Abbildung durch 8 Parameter darstellbar sind. Der entscheidende Vorteil dieser Näherungen gegenüber der projektiven Abbildung (Folie 14, unten), die auch 8 Parameter beinhaltet, ist, dass die affine und bilineare Abbildung durch Polynome darstellbar sind, während die (exakte) projektive Abbildung durch einen Bruch dargestellt wird. Zur Bemerkung bzgl. Abbildung von Dreiecken bzw. Vierecken -> siehe Folie 17.

Folie 17 illustriert die Bewegungsvektoren zu 4 verschiedenen Bewegungsmodellen. Das einfachste Modell ist die Translation (links oben). Die affine Abbildung (rechts oben) wird durch Angabe der Verschiebungsvektoren an 3 Eckpunkten eindeutig bestimmt. Der Verschiebungsvektor für den 4. Eckpunkt (sowie die Vektoren im Innern der dargestellten Fläche) ist dadurch automatisch festgelegt. Anders bei der bilinearen Abbildung (links unten): Diese wird durch Angabe der Bewegungsvektoren an allen 4 Eckpunkten festgelegt. Die projektive Abbildung (rechts unten) wird ebenfalls durch Angabe der Bewegungsvektoren an allen 4 Eckpunkten festgelegt. Zu beachten ist, dass durch die projektive Abbildung auch *beliebige* Vierecke aufeinander abgebildet werden können (durch Vorgabe der Bewegungsvektoren an den Eckpunkten), während bei der bilinearen Abbildung nicht-achsenparallele Geraden im Allgemeinen nach der Abbildung nicht mehr als Geraden erscheinen. Die verschiedenen Abbildungsmodelle spielen eine wichtige Rolle bei der Entwicklung von Verfahren zur Beschreibung von Bewegungen in Videosequenzen.

Folie 18 illustriert den Unterschied zwischen *tatsächlicher* 2D Bewegung (als Folge einer zu Grunde liegenden 3D Bewegung) und der *wahrgenommenen* 2D Bewegung, welche auch als *Optical Flow* bezeichnet wird.

Auf **Folie 19** wird die Optical Flow Gleichung abgeleitet, welche die Grundlage für viele Algorithmen der Videoverarbeitung bildet (siehe LU Videoverarbeitung). In der Literatur findet man verschiedene Möglichkeiten, die Gleichung abzuleiten, hier ist nur eine Variante angegeben. Die Funktion ψ bezieht sich auf die Helligkeit (bzw.

Grauwerte) der Bildpunkte, x und y sind die Bildkoordinaten (Pixel) und t die Zeit. Ausgangspunkt der Ableitung ist die Annahme, dass ein- und derselbe Bildpunkt in verschiedenen Frames (d.h. zu verschiedenen Zeitpunkten) denselben Grauwert beibehält. (Dies ist in der Praxis jedoch nicht immer erfüllt, z.B. durch Änderungen in der Beleuchtung.) Diese “constant intensity assumption” ist in der 1. Gleichung wiedergegeben: Der Grauwert nach der Bewegung - nach der zeitlichen Änderung dt hat sich der Punkt (x, y) um den Verschiebungsvektor (dx, dy) weiterbewegt – ist derselbe wie vor der Bewegung (d.h. zum Zeitpunkt t am Ort (x, y)). In der zweiten Gleichung wird die Grauwertfunktion nach der Bewegung in eine Taylorreihe entwickelt. Die Annäherung durch die ersten Terme einer Taylorreihe entspricht einer Linearisierung der Funktion, welche aber nur in einer kleinen Umgebung des Ausgangswertes gültig ist (vgl. Näherung eines Kurvenverlaufs durch die Tangente an einen Kurvenpunkt). Die *Optical Flow Gleichung* ist daher nur für kleine Änderungen (d.h., kleine Bewegungsänderungen zwischen den betrachteten Frames) gültig. Durch Gleichsetzung der beiden rechten Seiten erhält man die Optical Flow Gleichung in der letzten Zeile auf Folie 19. Sie ist zunächst als Funktion der differentiellen Änderungen dx, dy, dt dargestellt und dann umformuliert (durch Division durch dt) zu einer Darstellung durch die Geschwindigkeitsvektoren $(v_x, v_y) = (dx/dt, dy/dt)$. Ganz rechts ist eine alternative Schreibweise mit Hilfe des Gradientenoperators ∇ gegeben.

Folie 20 illustriert, dass durch die Optical Flow Gleichung der Bewegungsvektor nur in der Richtung des Gradienten (d.h. in Richtung der stärksten Änderung im Bild(grauwert)inhalt) festgelegt ist. Die Bewegungskomponente normal dazu ist unbestimmt. Der Gradientenvektor ist in der Abbildung mit $\nabla \psi$ bezeichnet. Der Bewegungsvektor v wird in eine Gradienten-parallele Komponente vn und in eine Komponente normal dazu (vt) aufgespalten. Durch die Optical Flow Gleichung ist vn festgelegt, vt kann beliebig sein. Weiters kann in jenen Gebieten, wo der Gradient Null ist (d.h. homogener Bildinhalt, keine Textur bzw. Grauwertänderung), die Optical Flow Gleichung nicht gelöst werden.

Folie 21 zeigt verschiedene Möglichkeiten zur Modellierung von Bewegung. Am einfachsten ist ein globales Bewegungsmodell (links oben), jedoch wird dieses Bildinhalte mit mehreren sich voneinander unabhängig bewegenden Objekten nur unzureichend beschreiben. (Es kann aber in Einzelfällen ausreichend sein, z.B. statische Szene mit Kamerazoom – vgl. Folie 13.) Das detaillierteste und flexibelste Bewegungsmodell ist das Pixel-basierte (rechts oben): Für jedes Pixel wird ein eigener Bewegungsvektor bestimmt. Neben dem hohen Rechenaufwand hat dies jedoch auch den Nachteil, dass etwaige Zusammenhänge zwischen benachbarten Bildpunkten verloren gehen. Am idealsten wäre ein Regionen-basiertes Modell (rechts unten). Die Segmentierung von zusammengehörigen Regionen vor der Bewegungsbestimmung ist jedoch ein eigenes – nicht leicht zu lösendes - Problem. Als Kompromiss wird in der Praxis häufig ein Block-basiertes Modell verwendet (links unten), welches im Folgenden auch als Grundlage für die Suchalgorithmen auf Folie 22 bis 31 verwendet wird. Nicht angeführt ist in diesem Überblick die Netz-basierte Darstellung, welche dann auf Folie 32-35 illustriert wird.

Folie 22 beschreibt das Prinzip eines Block-basierten Matchingverfahrens zur Bewegungsbestimmung. Verwendete Bezeichnungen/Abkürzungen: MV (Motion Vector), pel (= pixel), DFD (Displaced Frame Difference: Unterschied zwischen 2 betrachteten Frames), B_m (betrachteter Block B [mit Index m]), dm (Bewegungsvektor

zugehörig zum Blocks B_m - in diesem Modell wird angenommen, dass alle Pixel eines Blocks ein- und dieselbe Translation, beschrieben durch dm , ausführen). ψ_1 und ψ_2 sind die Grauwertfunktionen der Frames vor und nach der Bewegung, E_{DFD} ist die Fehlerfunktion (Error), welche die Grauwertabweichungen zwischen dem ursprünglichen Block im ersten Frame und jenem Block im 2. Frame, der vom ursprünglichen Frame die Positionsabweichung dm besitzt, angibt. Die Suchaufgabe ist, für jeden Block B_m jenen Bewegungsvektor dm zu finden, welcher die Fehlerfunktion E (d.h. die Abweichung im Grauwertbild-Blockinhalt) minimiert. Setzt man den Parameter p auf 1, so wird die Fehlerfunktion durch einfaches Aufsummieren der Absolutwerte der Grauwertdifferenzen berechnet. Dies wird auf der Folie als *MAD* (Maximum Absolute Difference) Verfahren bezeichnet. Aufsummiert wird über alle Pixel x , die zum betrachteten Block B_m gehören.

Folie 23 illustriert die Block-basierte Suche. Es wird ein Suchbereich definiert (*search region*), innerhalb dieses Suchbereichs werden alle möglichen Lösungen berechnet (*exhaustive search* = *vollständige* Suche). Jene Lösung, die den minimalen Fehlerwert (= über Block aufsummierte Grauwertabweichungen) ergibt, bestimmt den für diesen Block berechneten optimalen Verschiebungsvektor dm .

Folie 24 zeigt Pseudo-Code (Matlab Beispiel) für die vollständige Block-basierte Suche (Exhaustive Block-based Matching Algorithm – *EBMA*), wobei sich die Suche im ganzzahligen Pixelraster bewegt.

Folien 25-28 illustrieren die Suche im Halbpixel-Raster, was eine höhere Genauigkeit bei der Bestimmung der Bewegungsvektoren ermöglicht. Im Suchframe wird dabei das Raster verfeinert (d.h., die Auflösung künstlich erhöht), indem die Grauwerte für die (fiktiven) Zwischenpunkte auf den Halbpixel-Positionen aus den ursprünglichen Grauwerten auf den ganzzahligen Pixelpositionen interpoliert werden.

Folien 29-31 illustrieren das Prinzip einer hierarchischen Suche unter Verwendung eines Block-basierten Verfahrens. Es wird zunächst eine Bildpyramide konstruiert, die ein und dasselbe Bild in sukzessive geringeren Auflösungen beinhaltet. Die Suche beginnt auf der Ebene mit der geringsten Auflösung. Das Ergebnis dieser Suche wird dann als Ausgangswert für die Suche auf der nächsthöheren Auflösungsstufe verwendet.

Folien 32-35 illustrieren das Prinzip der Netz-basierten (mesh-based) Suche. Es kann sowohl ein Dreiecks- als auch ein Vierecksnetz verwendet werden. Während als Ergebnis der Block-basierten Suche allen Pixeln eines Blocks ein und derselbe Verschiebungsvektor zugeordnet wurde, wird bei der Netz-basierten Suche jedem *Netzknoten* ein Verschiebungsvektor zugeordnet. Die Verschiebungsvektoren *innerhalb* der Netzelemente (Dreiecke oder Vierecke) werden danach durch geeignete Interpolation der Verschiebungsvektoren an den Knoten ermittelt. Dadurch erreicht man stetige Übergänge an den Rändern der Netzelemente, die Artefakte an den Blockgrenzen, welche bei den Block-basierten Verfahren auftreten, sind dadurch eliminiert (siehe Abbildungen auf den letzten zwei Folien).