

3. Analyse der Kamerabewegung

Video - Inhaltsanalyse

Stephan Kopf

Bewegungen in Videos

- **Objektbewegungen** (*object motion*)
- Kameraoperationen bzw. **Kamerabewegungen** (*camera motion*)

Semantische Informationen

- Aussagen über Schnitte
- Segmentierung von Objekten
- Berechnung von Panoramabildern



MIT Media Lab, 1996

Inhalt

- Modellierung der Kamerabewegung
- Berechnung von Bewegungsvektoren
- Berechnung des Kameramodells
- Erkennung gültiger Kameraparameter
- Ableitung semantischer Informationen

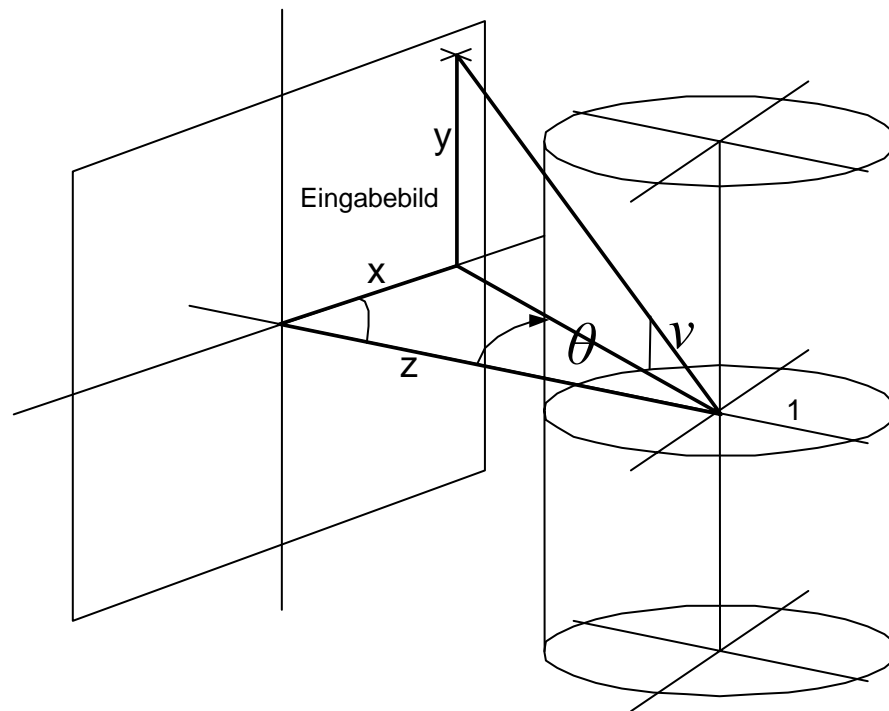
Modellierung der Kamerabewegung

Beschreibung der Kamerabewegung zwischen
2 Bildern:

- *zylindrisches Kameramodell*
- *sphärisches Kameramodell*
- *8-Parameter Modell*

Zylindrisches Kameramodell (I)

- Bildet horizontale Rotationen der Kamera ab.
- Ein Bild wird durch zylindrische Koordinaten dargestellt:



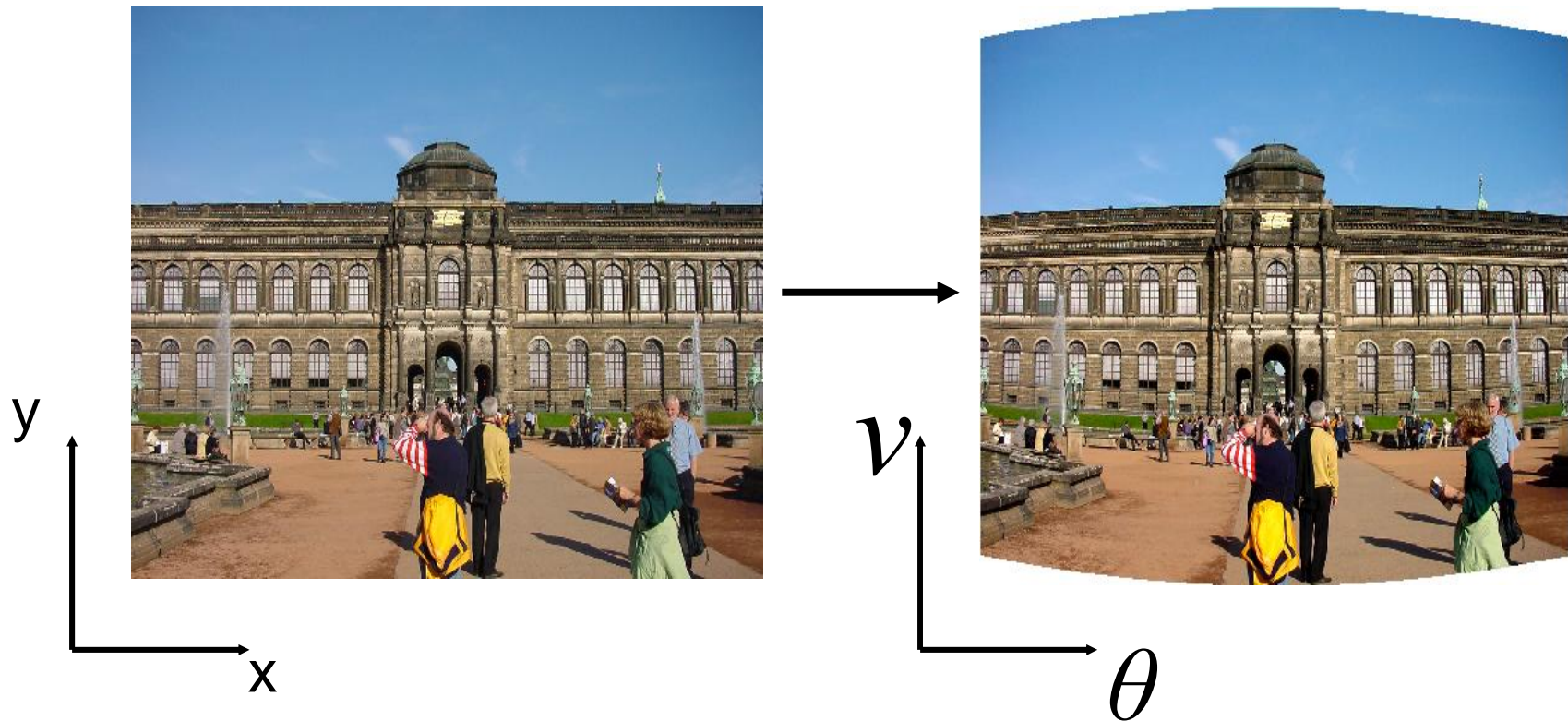
Winkel: $\theta = \arctan(x / z)$

Höhe:: $v = y / \sqrt{x^2 + z^2}$

Brennweite: z

Zylindrisches Kameramodell (II)

Beispiel



Zylindrisches Kameramodel (III)

Verschiebung von zwei transformierten Bildern

- Bilder werden übereinander gelegt.
- Zweites Bild wird so lange verschoben, bis Pixelunterschiede minimal sind.



Zylindrisches Kameramodell (IV)

Berechnung der Verschiebung

- Gesucht: Vektor $(t_x; t_y)$
- Fehler E zwischen Bild f und g wird minimiert:

$$E(t_x, t_y) = \sum_{x,y} |f(x, y) - g(x + t_x, y + t_y)|^2$$

- Komplexität bei vollständiger Suche über alle Vektoren:

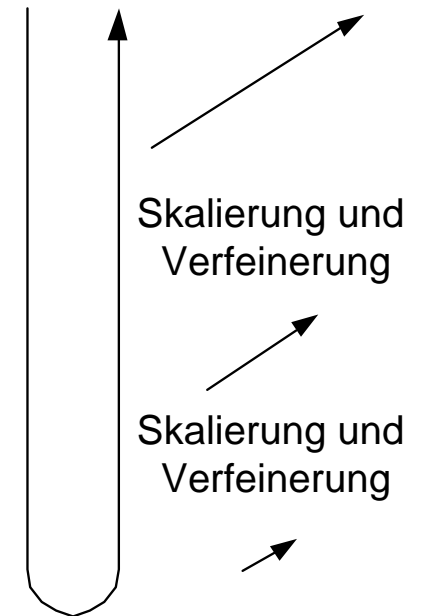
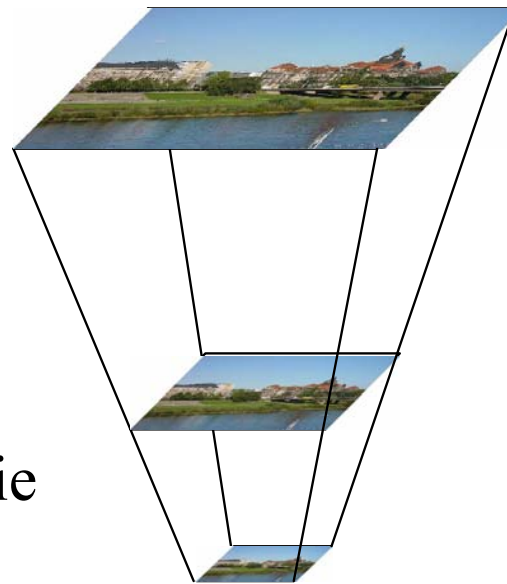
$$O(N^2 M^2) \approx O(N^4) \quad \rightarrow \text{sehr hoher Rechenaufwand}$$

Bildgröße: $N \times N$, Suchradius: $M \times M$

Zylindrisches Kameramodell (V)

Hierarchische Vorgehensweise

- Erzeuge Pyramide von Bildern in unterschiedlichen Auflösungen.
- Berechne Verschiebung für die geringste Auflösung.
- Skalieren Verschiebung für die nächste Ebene und berechne Verschiebung erneut.

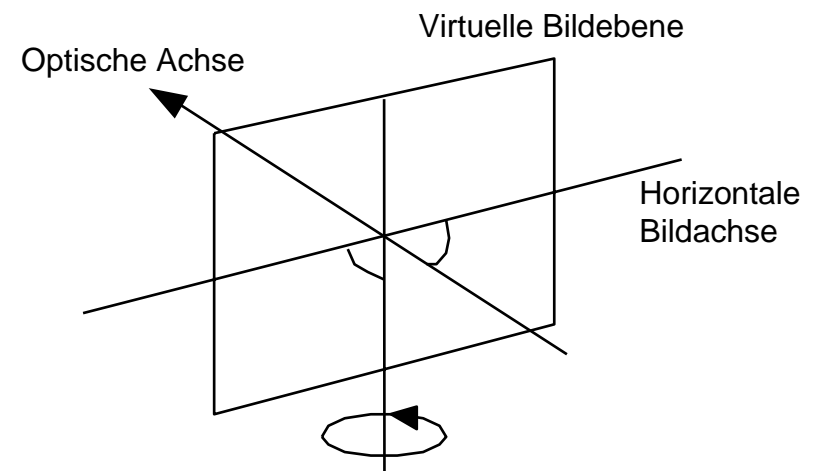


Zylindrisches Kameramodell (VI)

Nachteil

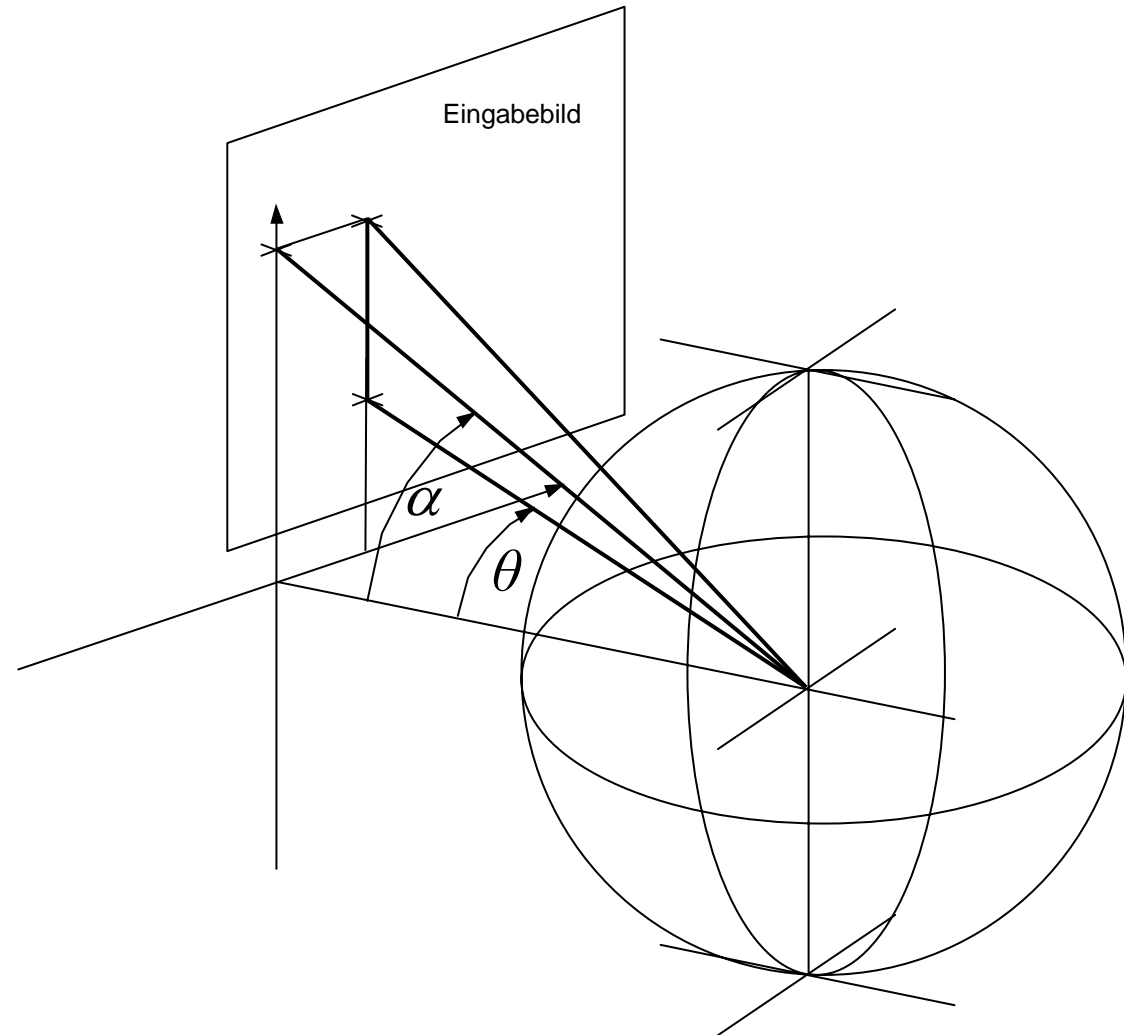


- Gerade Linien werden gebogen.
- Die Brennweite (Zoom) muss bekannt sein.
- Lediglich horizontale Rotationen werden abgebildet.



Sphärisches Kameramodell

- Abbildung des Bildes auf eine Kugel
- Horizontale und vertikale Rotationen der Kamera können beschrieben werden.



8-Parameter Modell (1)

- Annahme: Umgebung ist auf eine flache Glasplatte gemalt
- Welche Bewegungen der Glasplatte sind möglich?

2D

- Affine Bewegung (Translation, Rotation, Skalierung)
- kann durch ein **6-Parameter Modell** spezifiziert werden:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

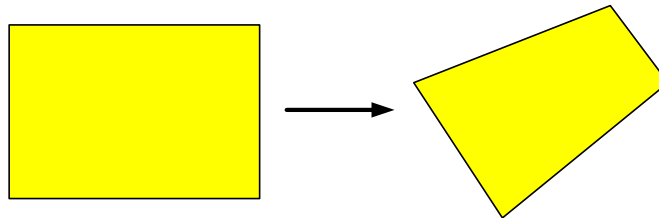
Rotation
Skalierung

Translation

8-Parameter Modell (2)

3D

- Affine Bewegung wird durch die Parameter $a_{i,j}$ und t_x, t_y beschrieben
- Zusätzlich: Perspektivische Verzerrungen



- 2 weitere Parameter b_1 und b_2 beschreiben die perspektivische Verzerrung

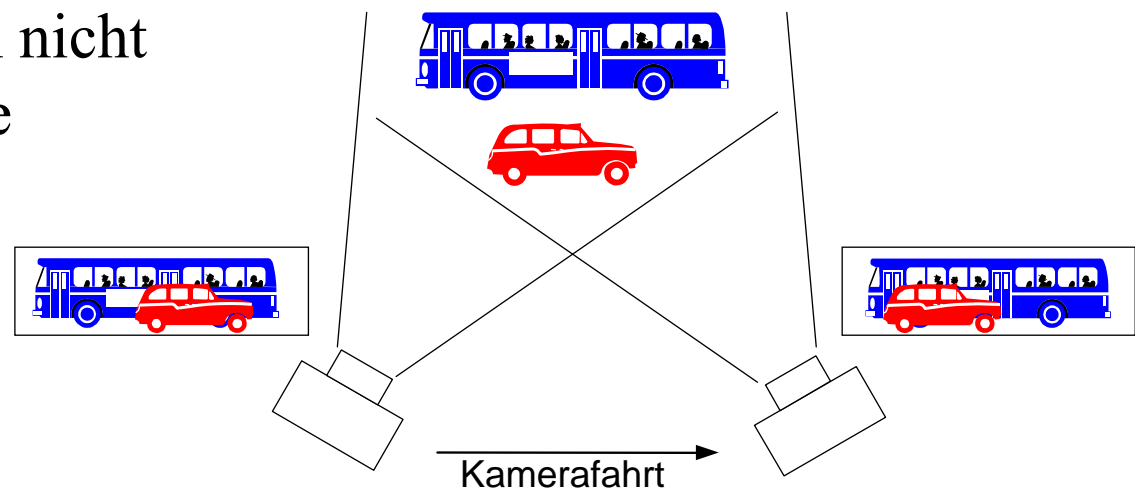
$$x' = \frac{a_{11}x + a_{12}y + t_x}{b_1x + b_2y + 1} \quad y' = \frac{a_{21}x + a_{22}y + t_y}{b_1x + b_2y + 1}$$

8-Parameter Modell (3)

- eine horizontale oder vertikale Verschiebung (waagrecht oder senkrechter Schwenk): t_x, t_y
- Zoomeffekt und Rotation der Kamera entlang der Blickrichtung: $a_{i,j}$
- Perspektivische Verzerrungen: b_1, b_2
- Kamerafahrten werden nicht abgebildet, da für diese 3D Informationen benötigt werden.

$$x' = \frac{a_{11}x + a_{12}y + t_x}{b_1x + b_2y + 1}$$

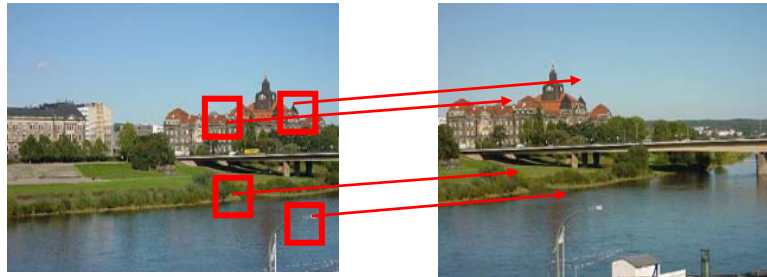
$$y' = \frac{a_{21}x + a_{22}y + t_y}{b_1x + b_2y + 1}$$



8-Parameter Modell (4)

Ermittlung der Parameter

- Suche 4 ähnliche Blöcke in beiden Bildern



- Die Blöcke definieren 4 Bewegungsvektoren, welche die Verschiebung der mittleren Pixel jedes Blockes beschreiben.
- Ein Bewegungsvektor beschreibt die horizontale und vertikale Verschiebung eines Blockes. Zur Berechnung der acht Parameter des Kameramodells reicht es aus, die genaue Verschiebung von vier Pixeln des Bildhintergrundes zwischen beiden Bildern zu kennen.

Bewegungsvektoren (1)

- Ein **Bewegungsvektor** (*motion vector*) beschreibt die Verschiebung eines Pixels zwischen zwei Bildern.
- **Optischer Fluss:** Beschreibt die Verschiebung aller Pixel eines Bildes.

Berechnung eines Bewegungsvektors mittels **Blockmatching**

- Das erste Bild wird in rechteckige Regionen eingeteilt.
- Für jede Region wird eine möglichst ähnliche Region im zweiten Bild gesucht.
- Für eine Position (x, y) in Bild i werden alle Positionen (x', y') in Bild j betrachtet, die in horizontaler oder vertikaler Richtung nicht mehr als r Pixel vom Punkt (x, y) entfernt liegen:

$$x - r \leq x' \leq x + r$$

$$y - r \leq y' \leq y + r.$$

Bewegungsvektoren (2)

Ähnlichkeit zwischen Blöcken

- Summe der absoluten Differenzen für Blöcke der Größe 16 x 16 Pixel

$$SAD = \sum_{\Delta x=-8}^7 \sum_{\Delta y=-8}^7 |I_i(x + \Delta x, y + \Delta y) - I_j(x' + \Delta x, y' + \Delta y)|.$$

- Die Position (x', y') wird so festgelegt, dass die Summe der absoluten Differenzen minimal wird. Der Bewegungsvektor von (x, y) nach (x', y') beschreibt die optimale Verschiebung des Blockes.

Nachteil

- Hoher Rechenaufwand
- In Regionen ohne eindeutige Strukturen sind viele Bewegungsvektoren fehlerhaft.

Bewegungsvektoren (3)

Berechnung eines Bewegungsvektors durch Erkennung spezieller Merkmale

- Identifiziere spezielle Merkmale (z.B. Ecken) im ersten Bild und suche diese im zweiten Bild.
- Ein Bewegungsvektor beschreibt dann die Veränderung der Position des ausgewählten Merkmals.

Erkennung von Ecken

- Schnittpunkt zweier Kanten, d.h. es müssen deutliche Bildänderungen in zwei unterschiedlichen Richtungen vorhanden sein.

Einschub: Erkennung von Ecken

Verfahren nach Moravec

- Für jedes Pixel wird die Region um dieses Pixel analysiert (4 benachbarte Pixel).
- Die Region wird verschoben und Unterschiede werden mittels quadrierter Differenzen ermittelt.
- Geringer Unterschied bei allen Verschiebungen
→ einheitliche Region, d.h. keine Ecken oder Kanten
- Deutlicher Unterschied bei einem Teil der Verschiebungen
→ Kante
- Deutlicher Unterschied bei allen Verschiebungen
→ Ecke

Bewegungsvektoren (4)

Zuordnung der Ecken zweier Bilder

1. Analysiere alle Kombinationen der Ecken der beiden Bilder.
2. Jede Kombination wird als ein möglicher Bewegungsvektor interpretiert, für den als Qualitätsmaß die Summe der absoluten Differenzen der umgebenden Blöcke berechnet wird.
3. Die beiden ähnlichsten Ecken werden einander zugeordnet, als ausgewählt markiert und definieren einen Bewegungsvektor.
4. Iterativ werden weitere Bewegungsvektoren erzeugt, indem unter den verbleibenden Ecken die jeweils ähnlichsten kombiniert und markiert werden.
5. Das Verfahren terminiert, wenn die Differenzen einen Schwellwert übersteigen und angenommen werden kann, dass keine korrekte Zuordnung von Ecken mehr möglich ist.

Bewegungsvektoren (5)

Beispiel: Blockmatching

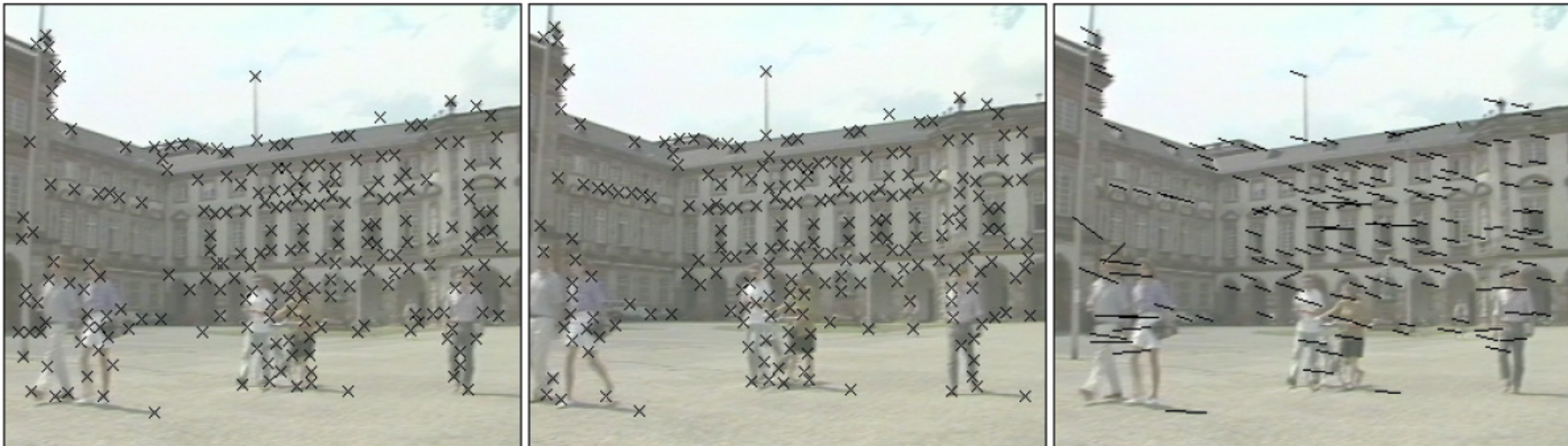


2 Bilder einer Videosequenz

Bewegungsvektoren
beim Blockmatching

Bewegungsvektoren (6)

Beispiel: Zuordnung von Ecken



erkannte Ecken

zugeordnete
Bewegungsvektoren

Problem:

- Fehlerhafte Zuordnung mehrerer Ecken, so dass falsche Bewegungsvektoren entstehen.

Berechnung des Kameramodells (I)

Ziel

- Unter allen Vektoren sollen die Vektoren ausgewählt werden, die das Kameramodell und nicht die Objekte im Vordergrund korrekt beschreiben

Vorgehensweise

- Annahme: mindestens die Hälfte der Vektoren beschreibt die Bewegung des Hintergrunds
- Berechnung der Kamerabewegung mit einer robusten Regressionsschätzung (*kleinste getrimmte Quadrate*)

Verfahren der kleinsten getrimmten Quadrate

1. Zufällig werden vier Bewegungsvektoren aus der Menge aller Vektoren ausgewählt.
2. Mit diesen vier Vektoren ist es möglich, durch Lösen eines linearen Gleichungssystems mit acht Gleichungen die acht Parameter des Kameramodells eindeutig zu berechnen.

Berechnung des Kameramodells (II)

3. Eine Fehlerfunktion e_i klassifiziert den Fehler für jeden Bewegungsvektor i zwischen der tatsächlichen Position eines Pixels (x', y') im zweiten Bild und der durch das Kameramodell geschätzten Position (\hat{x}, \hat{y}) .

$$e_i = (x'_i - \hat{x}_i)^2 + (y'_i - \hat{y}_i)^2.$$

4. Vektoren aus Bereichen des Bildvordergrundes sollen die Fehlerfunktion nicht beeinflussen, d.h. es sollen nur Vektoren ausgewählt werden, die gut zum Modell passen. Dazu werden die Fehler der einzelnen Vektoren aufsteigend nach ihrer Größe sortiert, so dass in der zweiten Hälfte der Liste die Vektoren enthalten sind, die stärker vom Kameramodell abweichen.
5. Nach Berechnung des Fehlers E für die Schätzung des Kameramodells werden erneut vier Bewegungsvektoren aus der Menge aller Bewegungsvektoren zufällig ausgewählt (Schritt 1).

$$E = \sum_{i=1}^{N/2} e_i \quad \text{mit } e_1 \leq \dots \leq e_N.$$

Berechnung des Kameramodells (III)

Beispiel



helle Vektoren: starke Abweichung vom Modell

berechnetes Kameramodell

dunkle Vektoren: geringe Abweichung vom Modell

Ursachen für Fehler

Probleme

- schwach strukturierte Regionen
→ Fehler beim Blockmatching
- Kameraeinstellungen mit niedrigem Kontrast
→ geringe Anzahl an Ecken und fehlerhaftes Kameramodell
- Regelmäßige Strukturen im Bild
→ fehlerhafte Bewegungsvektoren
- Große Objekte im Bildvordergrund
→ mehr als die Hälfte der Bewegungsvektoren beschreiben
Vordergrundobjekte
→ fehlerhaftes Kameramodell

Erkennung gültiger Kameraparameter (I)

Kameraoperation	t_x, t_y	a_{11}, a_{22}	a_{12}, a_{21}	p_x, p_y
Statische Kamera	0	1	0	0
Translation	$\neq 0$	1	0	0
Skalierung				
- Zoom-in	0	$0 < a_{00} = a_{11} < 1$	0	0
- Zoom-out	0	$a_{00} = a_{11} > 1$	0	0
Rotation um Winkel θ	0	$a_{00} = a_{11} = \cos \theta$	$a_{01} = -a_{10} = \sin \theta$	0
Scherung				
- horizontal	0	1	$a_{01} \neq 0$	0
- vertikal	0	1	$a_{10} \neq 0$	0
Spiegelung				
- horizontal	0	$a_{00} = -1$	0	0
- vertikal	0	$a_{11} = -1$	0	0
Persp. Verzerrung	0	1	0	$\neq 0$

Erkennung gültiger Kameraparameter (II)



(a)



(b)



(c)



(d)



(e)



(f)



(g)

- (a) Keine Transformation: statische Kamera
- (b) Translation: horizontaler Schwenk ($t_x = -0,1 \cdot \text{Bildbreite}$)
- (c) Skalierung: Zoom-In ($a_{00} = a_{11} = 0,33$)
- (d) Rotation um 10 Grad ($a_{00} = a_{11} = 0,985$, $a_{01} = -a_{10} = 0,174$)
- (e) Scherung: in Videos nicht möglich ($a_{01} = 0,15$)
- (f) Spiegelung: in Videos nicht möglich ($a_{11} = -1$)
- (g) Perspektivische Verzerrung ($p_x = -4 \cdot 10^{-5}$, $p_y = 0$)

Erkennung gültiger Kameraparameter (III)

- Die Kameraparameter müssen innerhalb fester Intervalle liegen.

Zulässige Werte

Kameraoperation	t_x, t_y	a_{11}, a_{22}	a_{12}, a_{21}	p_x, p_y
statische Kamera	$0 \pm 0,8$	$1 \pm 0,01$	$0 \pm 0,01$	$0 \pm 1 \cdot 10^{-6}$
horizontaler Kameranachwenk	$0 \pm \frac{1}{5}W$	$1 \pm 0,02$	$0 \pm 0,02$	$0 \pm 2 \cdot 10^{-4}$
Zoomeffekt	$0 \pm 0,8$	$1 \pm 0,08$	$0 \pm 0,08$	$0 \pm 1 \cdot 10^{-5}$
Rotation (max. $\theta = 5^\circ$)	$0 \pm 0,8$	$1 \pm 0,01$	$0 \pm 0,09$	$0 \pm 1 \cdot 10^{-5}$

W : Bildbreite

Erkennung gültiger Kameraparameter (IV)

Erkennungsraten

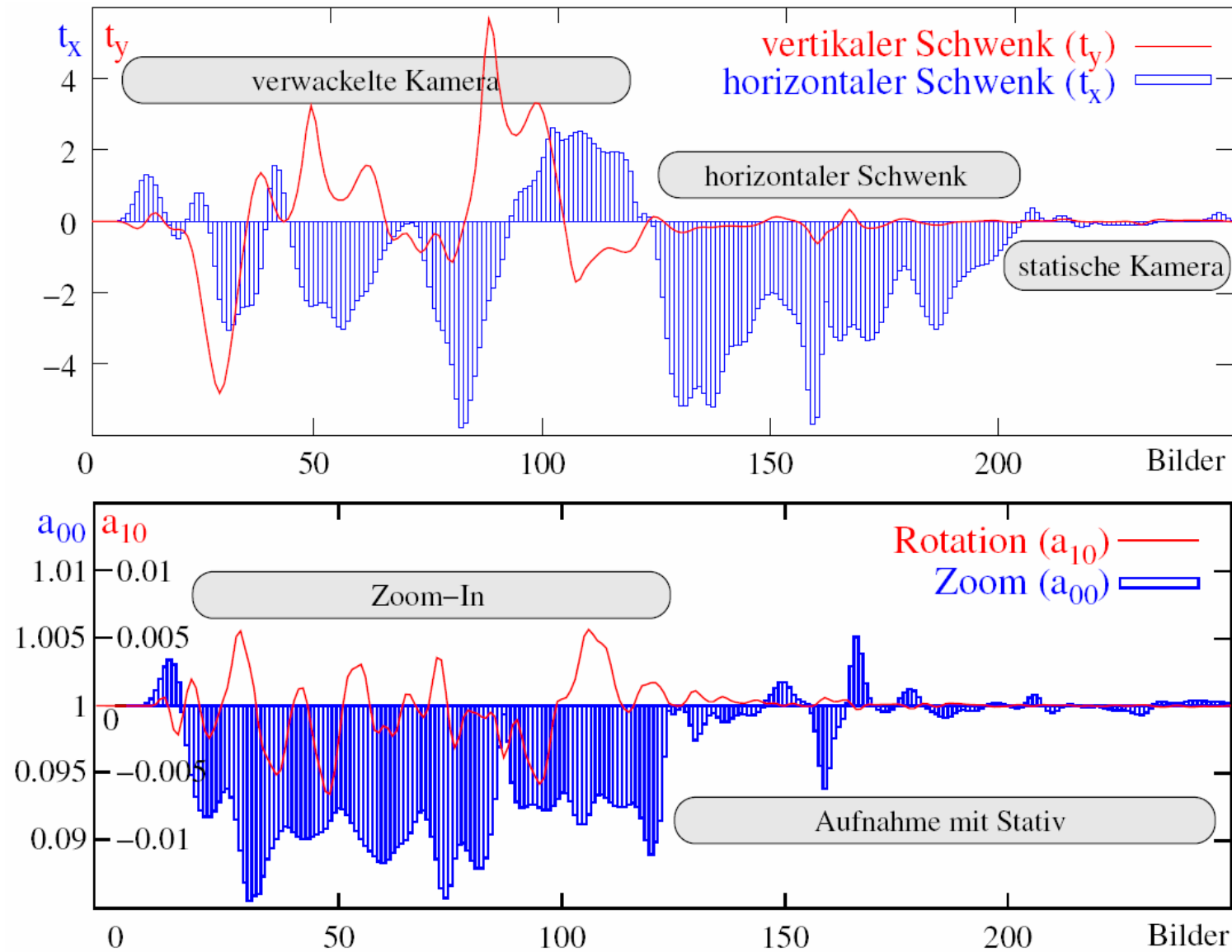
Kameramodell	Anteil	Ursache
korrekt erkannt	94,8 %	
Ecken wurden nicht erkannt	0,3 %	geringer Kontrast
Zuordnung der Ecken zu den Bewegungsvektoren nicht möglich	0,1 %	harte Schnitte, plötzliche Bildänderungen
fehlerhaftes Modell	4,8 %	harte und weiche Schnitte, Objektbewegungen

Ableitung semantischer Informationen (I)

Beschreibung der Kameraoperation

- Anhand der Kameraparameter können Start, Länge und Stärke eines Kameraschwenks oder Zoomeffektes automatisch charakterisiert werden.
- Anhand der Rotation können Rückschlüsse über die Art der Aufnahme gezogen werden (z.B. Aufnahme mit Stativ).

Ableitung semantischer Informationen (II)



Ableitung semantischer Informationen (III)

Analyse am Beispiel von 9 Testsequenzen

	horizontaler Schwenk	vertikaler Schwenk	eingehender Zoom	ausgehender Zoom
Dokumentation	31	12	12	21
Nachrichtensendung	40	18	14	30
Spielfilm	32	4	15	33
Talkshow	41	9	28	48
Serie	18	11	19	24
Zeichentrickfilm	3	1	2	16
Sportsendung	81	7	13	28
Musikclip	27	10	10	24
Werbung	18	19	18	20
Summe	301	88	123	254

Fragen ?