EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MATHEMATISCH-
NATURWISSENSCHAFTLICHE FAKULTÄT

Medieninformatik / Human-Computer Interaction

Grundlagen der Multimediatechnik

Videoanalyse

20.01.2022, Prof. Dr. Enkelejda Kasneci

Termine und Themen

22.10.2021	Einführung
29.10.2021	Menschliche Wahrnehmung – visuell, akustisch, haptisch, …
05.11.2021	Informationstheorie, Textcodierung und -komprimierung
12.11.2021	Bildverbesserung
19.11.2021	Bildanalyse
26.11.2021	Grundlagen der Signalverarbeitung
03.12.2021	Bildkomprimierung
10.12.2021	Videokomprimierung
17.12.2022	Audiokomprimierung
21.01.2022	Videoanalyse & Dynamic Time Warping
28.01.2022	Gestenanalyse
04.02.2022	Tiefendatengenerierung
11.02.2022	FAQ mit den Tutoren
17.02.2022	Klausur, 14-16 Uhr, N10+N11

Schnitterkennung

Schnitt (Cut):

- Plötzlicher Wechsel des Bildinhaltes zwischen zwei kontinuierlichen Aufnahmen

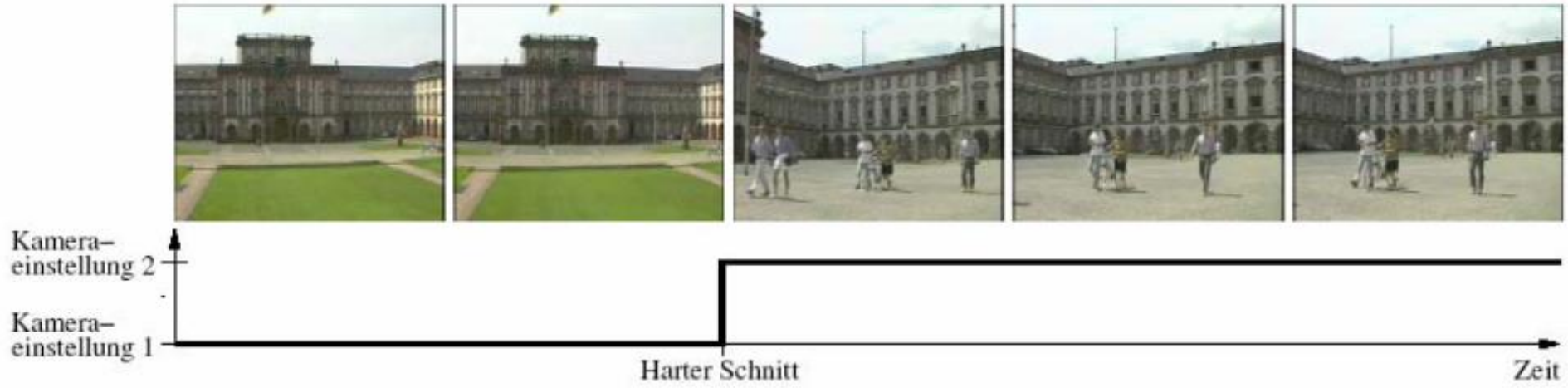


Bild: Stephan Kopf, Videoanalyse

Schnitterkennung:

- Erkennen von Schnitten innerhalb einer Videosequenz

Aufnahmeübergänge: Ein- und Ausblenden

Einblenden (Fade in)

- Wechsel eines Bildinhaltes von monochromer Farbe zu Bild

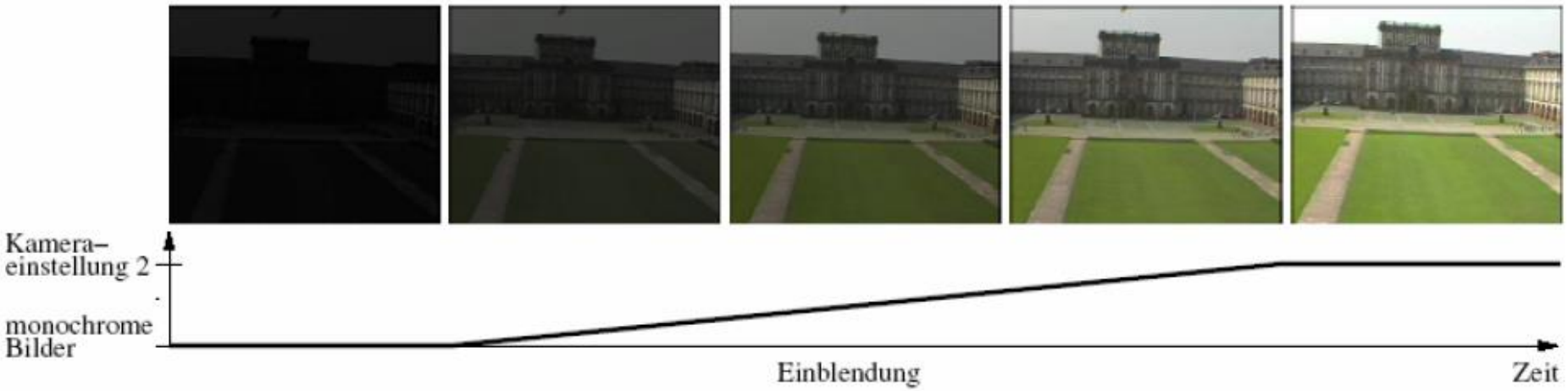


Bild: Stephan Kopf, Videoanalyse

Ausblenden (Fade out)

- Wechsel eines Bildinhaltes von Bild zu monochromer Farbe
  - Beispiel: Überblenden von weiß/schwarz

Aufnahmeübergänge: Überblendung

Überblendung

- Fließender Übergang zwischen zwei Bildinhalten mit Bildüberlagerung

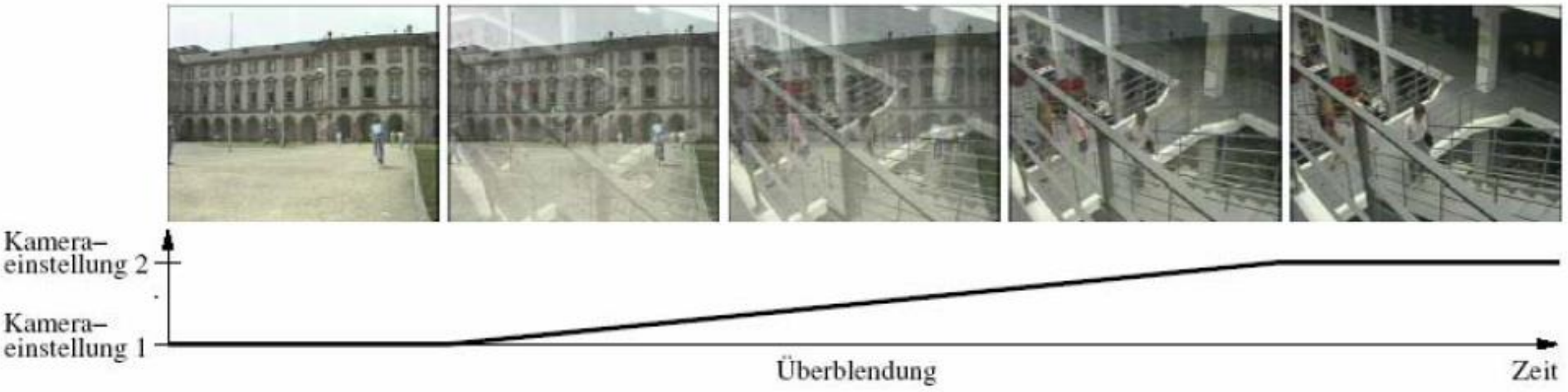


Bild: Stephan Kopf, Videoanalyse

Erkennung von Aufnahmeübergängen

Algorithmen

- Pixelbasierte Verfahren
- Unterschiede von Farbhistogrammen
- Kantenextraktion Edge Change Ratio (ECR)
- Kantenorientierter Kontrast

Pixelbasierte Schnitterkennung

Summe der absoluten Pixeldifferenzen $\boldsymbol{D_{SAD}}$ zweier Bilder $\boldsymbol{I_i, I_{i-1}}$

$$\boldsymbol{D_{SAD}} = \frac{\mathbf{1}}{\boldsymbol{N_x \cdot N_y}} \cdot \sum_{\boldsymbol{x=1}}^{\boldsymbol{N_x}} \sum_{\boldsymbol{y=1}}^{\boldsymbol{N_y}} |\boldsymbol{I_i(x,y) - I_{i-1}(x,y)}|$$

mit $N_x = \text{Bildbreite}, \mathrm{N_y} = \text{Bildhöhe}$

Falls $D_{SAD} > Threshold\ T \Rightarrow Harter\ Schnitt$

Vorteil: geringe Komplexität, robuste Ergebnisse
Nachteil: hohe Fehlerraten bei starker Bewegung (Objekt oder Kamera)

Histogrammbasierte Schnitterkennung

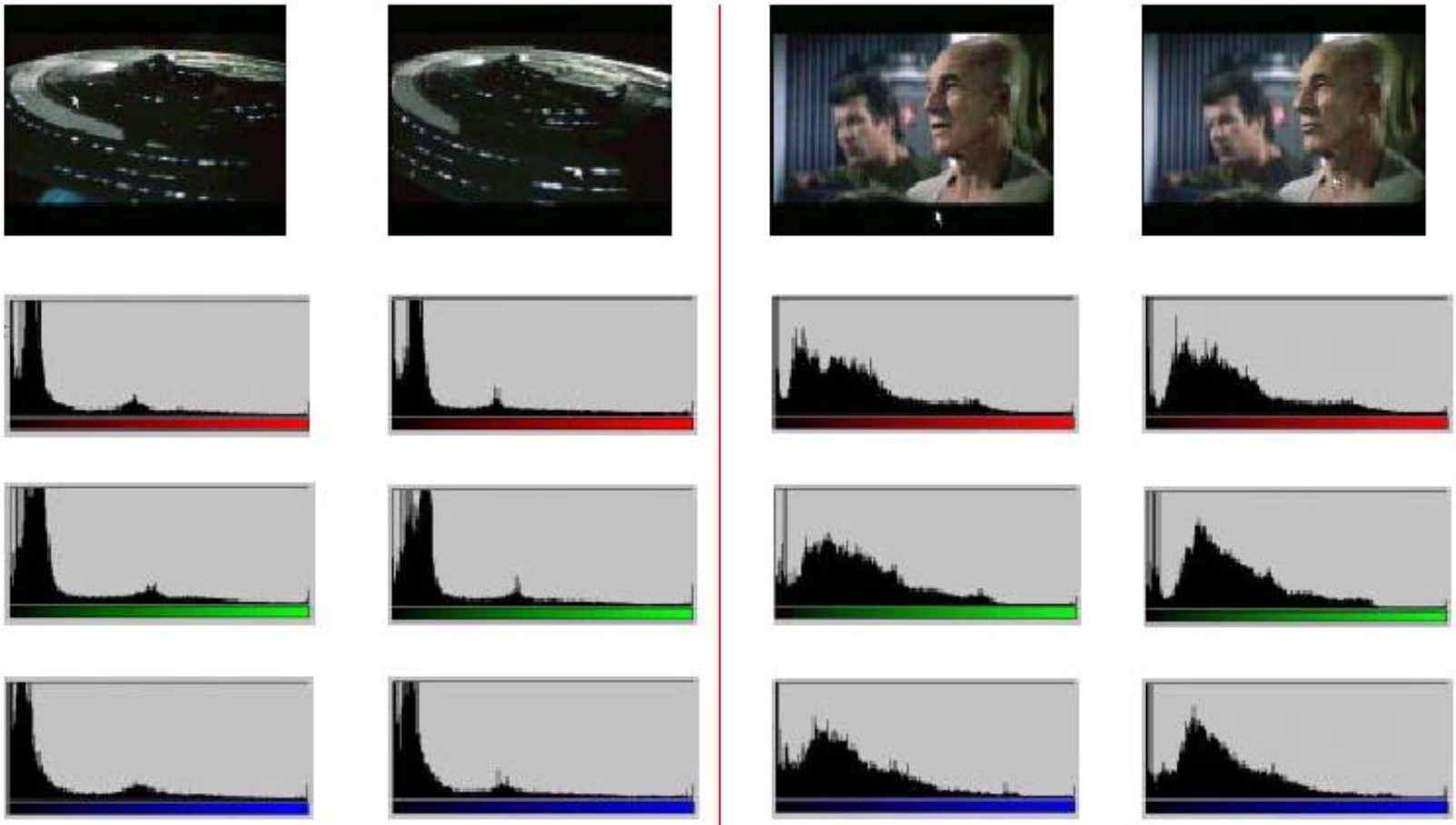
Schnitt

- Schnitt wird erkannt, wenn Farbhistogramme benachbarter Bilder $(i-1)$ und i sich mehr unterscheiden als Schwelle T
- Berechnung
  - Histogramm $H_i(r,g,b)$ eines Bildes i
  - RGB-Farbtripel (r,g,b)
  - Bilder $(i-1)$ und i

$$\sum_{r,g,b} \left(|H_i(r,g,b) - H_{i-1}(r,g,b)|\right) \geq T$$

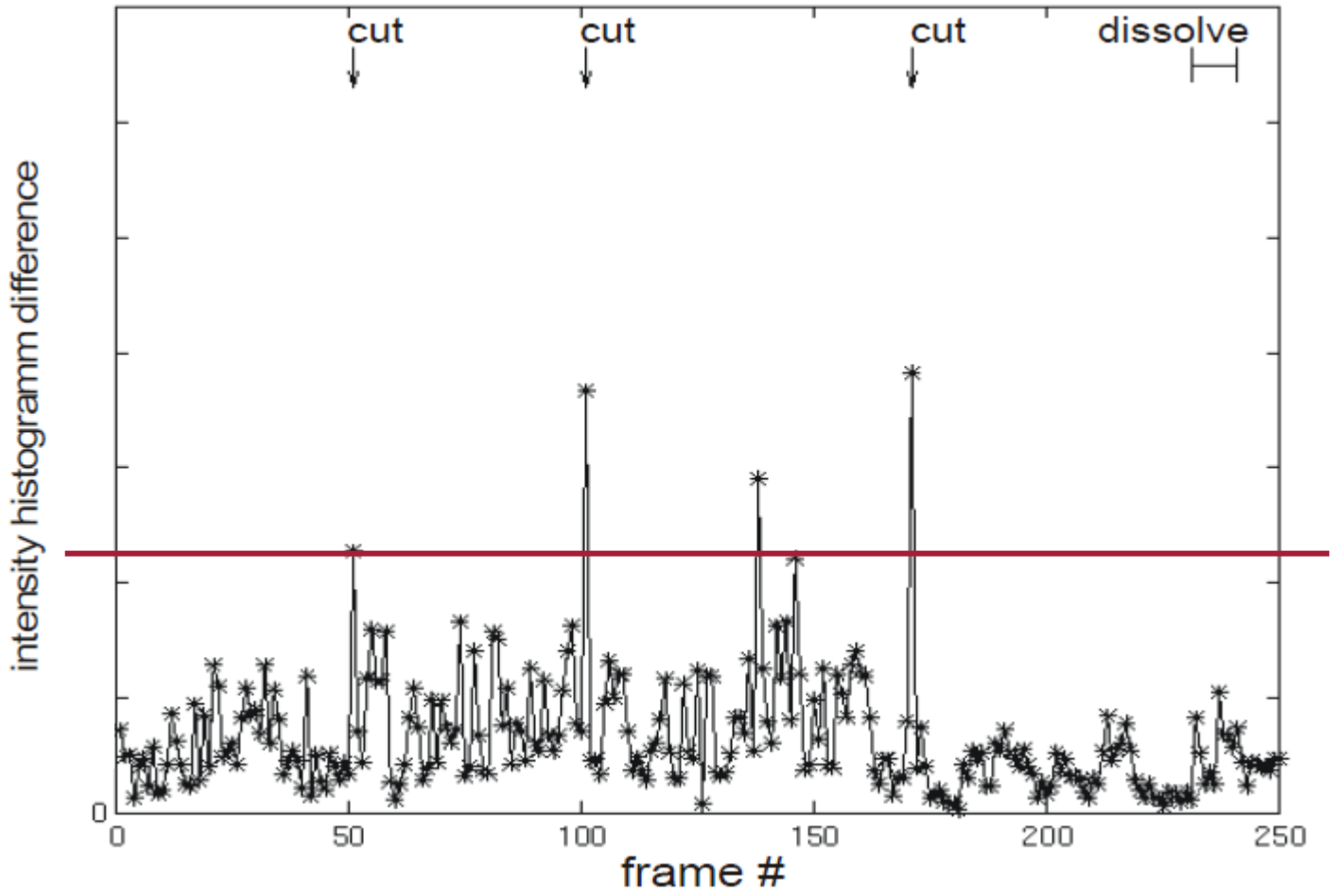
- auch die normierte Histogrammdifferenz verwendbar

Histogramme: Beispiele



Schnitt

Histogramme: Differenzgraph

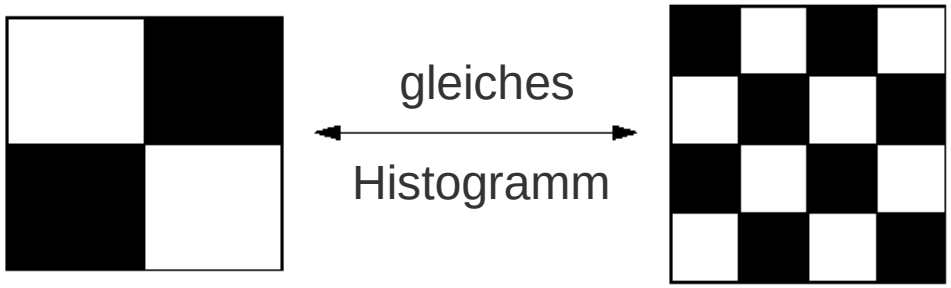


Schwelle T

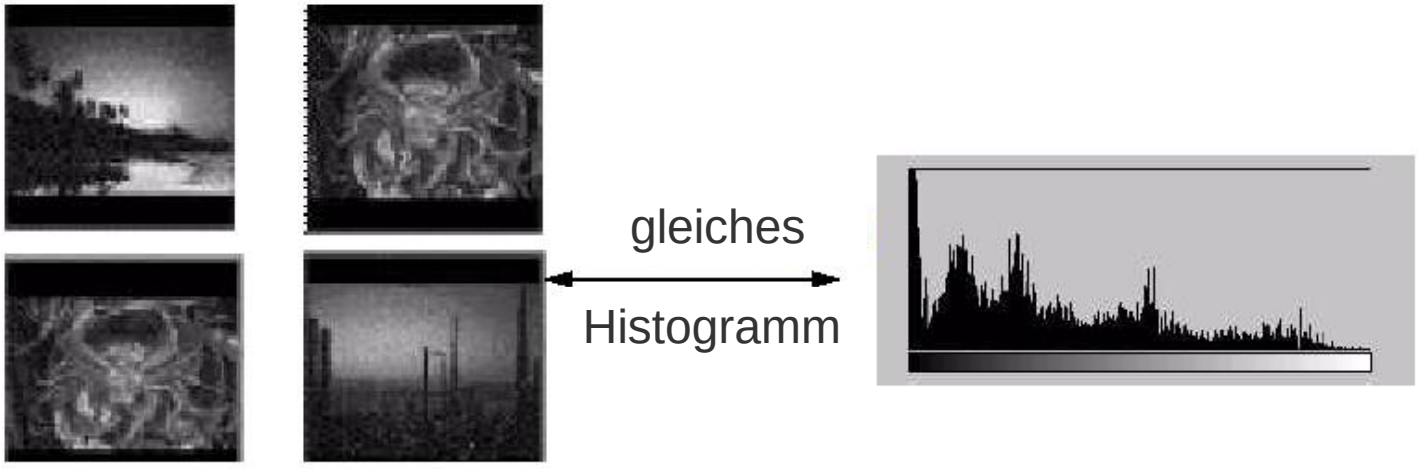
Histogrammbasierte Schnitterkennung: Probleme

Unterschiedliche Bilder können dieselben Histogramme haben

- **Einfaches Beispiel**



- **Kompliziertes Beispiel**



Histogrammbasierte Schnitterkennung: Probleme (2)

Probleme

- Verschiedene Bilder können ähnliche Histogramme haben
- Farbwerte von aufeinanderfolgenden Bildern können sich stark ändern, ohne dass es zu einem Schnitt kommt
  - Explosionen
  - Wechsel der Szenenbeleuchtung
- Schnelle Bewegung großer Objekte, deformierbare Objekte

Erkennungs-Performanz

- Abhängig vom gewählten Histogramm-Threshold
- Muss auf den Videoinhalt abgestimmt werden für optimale Ergebnisse
- Action-Szenen führen häufig zu falschen Schnitterkennungen

Verbesserungsmöglichkeit:

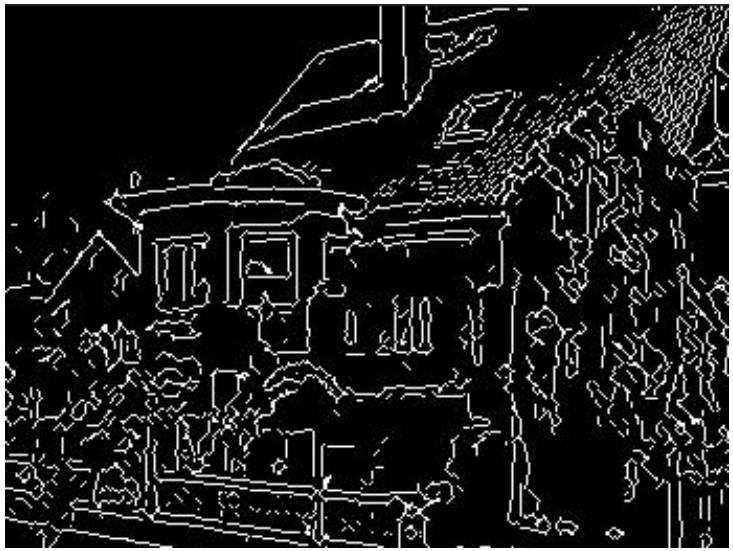
- Einteilung des Bildes in Regionen (z.B. 16)
- Durchführung der Schnitterkennung nur auf den ähnlichsten Regionen (z.B. 8)

Kantenbasierte Schnitterkennung

Vorgehensweise

- Berechnung der Kantenbilder durch Canny-Algorithmus
  - Bildglättung
  - Anwendung Sobel Filter
  - Berechnung der Kantenstärke
  - Non-maximum-supression
  - Hysterese

- Berechnung der ECR:
  Edge Change Ratio



https://upload.wikimedia.org/wikipedia/commons/c/ca/Canny_Final.JPG

Edge Change Ratio (ECR)

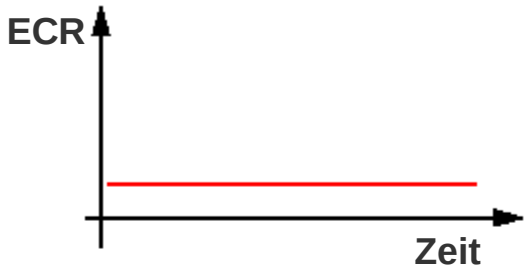
Eigenschaften

- Kantenpixel in Bild i und $(i-1)$: s_i und s_{i-1}
- E_{out}: Pixel in Bild $(i-1)$ ist Kantenpixel, Pixel in Bild i ist kein Kantenpixel
- E_{in}: Pixel in Bild $(i-1)$ ist kein Kantenpixel, Pixel in Bild i ist Kantenpixel
- Kantenunterschiede zwischen Bildern i und $(i-1)$

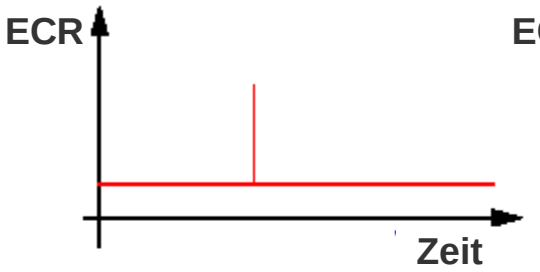
$$ECR_{i-1} = \max\left(\frac{E_{in}}{s_{i-1}}, \frac{E_{out}}{s_i}\right)$$

- ECR kann als einfache Eigenschaft zu Verfolgung von Bewegungsintensität verwendet werden

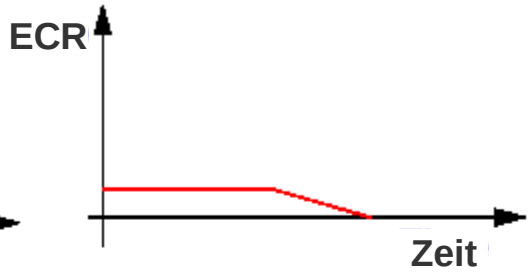
ECR-Schnitterkennung: Prinzipielle Idee



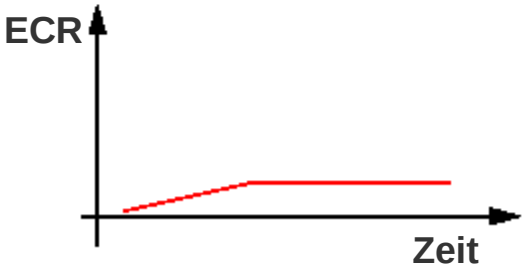
Innerhalb einer Szene



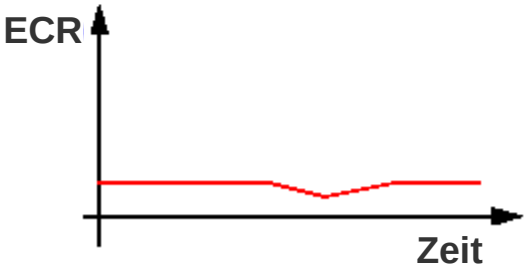
Harter Schnitt (Cut)



Ausblenden (Fade out)



Einblenden (Fade in)



Überblenden (Dissolve)

ECR-Schnitterkennung (2)

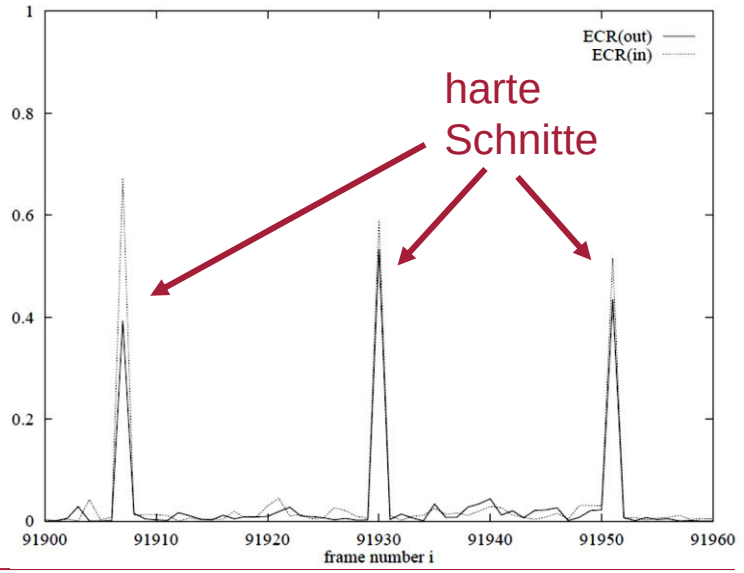
Schnitt

- Wenn ECR_i Kantenwechselverhältnis zwischen Bildern i und $(i-1)$ ist, wird ein Schnitt erkannt, falls

$$ECR_i \geq T$$

gilt, wobei T eine vorgegebene Schwelle ist.

- Schnelle Objekt- und Kamerabewegungen führen zu höheren ECR-Werten ohne Schnitte
  - Bewegungskompensation zwischen Bildern notwendig



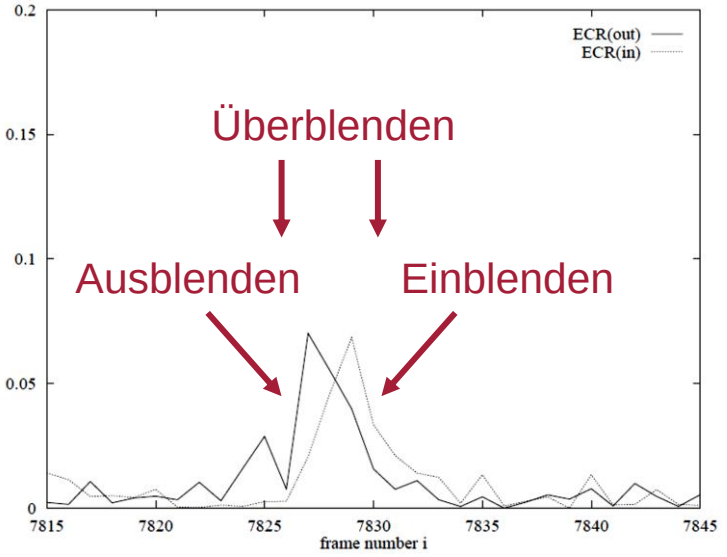
ECR-Schnitterkennung (3)

Überblendungen

- sind schwer zu erkennen
- typische Kurve eines ECR-Graphen

Beispiel: Überblenden

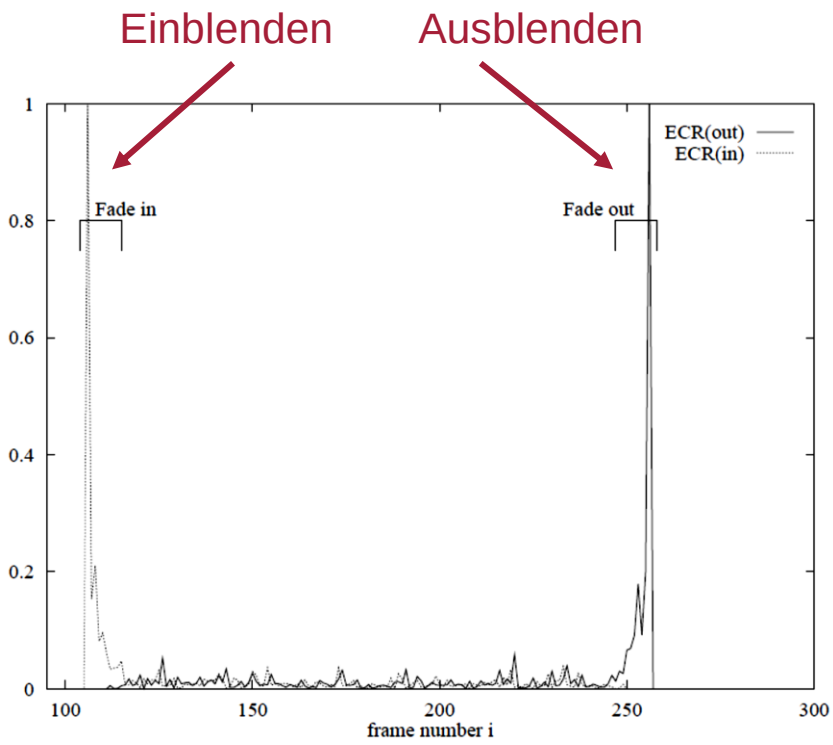
- Ränder der ersten Aufnahme verschwinden linear: $ECR_{i-1}^{out} = E_{out}/s_i$
- Ränder der neuen Aufnahmen erscheinen linear: $ECR_{i-1}^{in} = E_{in}/s_{i-1}$



ECR-Schnitterkennung (4)

Einblenden, Ausblenden

- **Ausblenden:** Zahl der Kantenpixel ist Null nach der ersten Bildsequenz
- **Einblenden:** Zahl der Kantenpixel ist Null vor der ersten Bildsequenz



- Erkennung von Überblendung durch Histogramme
  - Unzuverlässig, da Farbübergänge in Aufnahmen häufig vorkommen und daher nicht typisch für Erkennung von Überblendungen sind

ECR-Schnitterkennung (5)

Probleme

- Schnelle Objekte oder Kamerabewegung
- Explosionen
- Überblendungen und Ausblendungen
  - Sanfte Übergängen sind schwer zu erkennen, insbesondere Anfang/Ende eines Übergangs

- **Erkennungs-Performanz**
  - **Harte Schnitte:** sehr gut geeignet
  - **Ein-/Ausblendungen:** sehr hohe Rate falsch erkannter Schnitte → bedingt geeignet
  - **Überblendungen:** extrem hohe Rate falsch erkannter Schnitte → ungeeignet

Erkennung von Überblendungen mit kantenorientiertem Kontrast

Vorgehensweise

- Kantenstärke ist niedrig zwischen Überblendungen und Auflösungen
- Berechne Kanten und Kantenstärke
- $\boldsymbol{K_i(x,y)}$ bezeichne die **Kantenkarte** eines Bildes i und t_w und t_s der Schwellwert für ein **schwaches (weak)** bzw. ein **starkes (strong) Kantenpixel**
- **Vergleiche Beziehung $\boldsymbol{EC(i)}$ von starken zu schwachen Kanten**

$w(i) = \sum_{x,y} K_i(x,y)$ falls $t_w \leq K_i(x,y) < t_s$

$s(i) = \sum_{x,y} K_i(x,y)$ falls $K_i(x,y) \geq t_s$

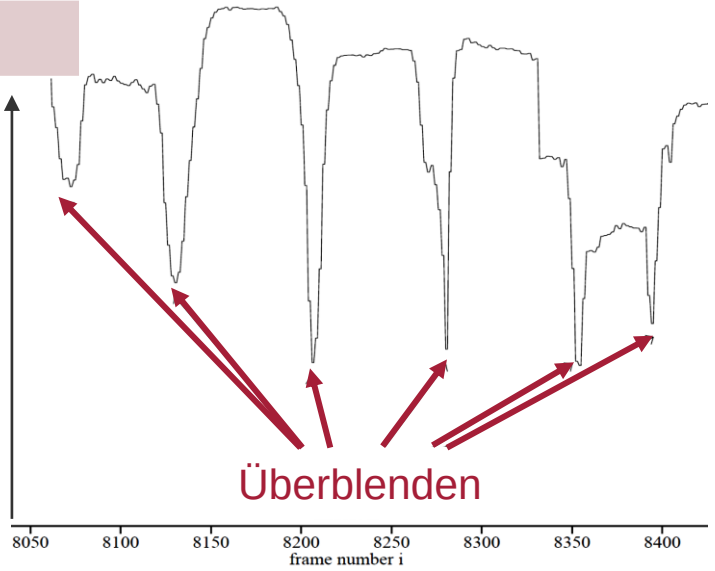
$EC(i) = 1 + \frac{s(i)-w(i)-1}{s(i)+w(i)+1}$; $EC(i) \in [0,2]$

Erkennung von Überblendungen mit kantenorientiertem Kontrast

Vorgehensweise

- Niedrige Kantenstärke → Überbelendung

EC(i)	Kanten
≈ 0	Keine ausgeprägten Kanten
0 < EC(i) < 1	Schwache Kanten dominieren
≈ 1	Schwache und starke Kanten
1 < EC(i) < 2	Starke Kanten dominieren
≈ 2	Nur starke Kanten



Schnitterkennungstechniken: Bewertung

Trefferrate (Recall) : Verhältnis richtig erkannter zu tatsächlichen Schnitten

$$recall = \frac{TP}{TP + FN}$$

Präzision: Anteil korrekt erkannter Schnitte an erkannten Schnitten

$$precision = \frac{TP}{TP + FP}$$

Schnitterkennungstechniken: Vergleich (Trefferrate)

Trefferrate: Verhältnis richtig erkannter zu tatsächlichen Schnitten

Schnitt-Erkennung	Schnitt-Typ	Generierte Dissolves	Ground-hog Day	Nach-richten	Baywatch
Histogramme*	**Schnitt**	59-90 %	18-98 %	40-99 %	50-82 %
	Ein-/Ausbl.	*nicht anwendbar*			
	Überbl.	*nicht anwendbar*			
ECR	**Schnitt**	90 %	97 %	91 %	69 %
	Ein-/Ausbl.	0 %	100%	0%	47 %
	Überbl.	72 %	67 %	0%	66 %
Kantenkontrast*	**Schnitt**	*nicht anwendbar*			
	Ein-/Ausbl.	*nicht anwendbar*			
	Überbl.	56-82 %	17 %	100 %	55-73 %

* Ergebnisse sind abhängig vom Threshold

Schnitterkennungstechniken: Vergleich (Fehlerrate)

Fehlerrate: Verhältnis falsch erkannter zu tatsächlichen Schnitten

Schnitt-Erkennung	Schnitt-Typ	Generierte Dissolves	Ground-hog Day	Nach-richten	Baywatch
Histogramme*	**Schnitt**	4-44 %	4-82 %	4-61 %	50-82 %
	Ein-/Ausbl.	*nicht anwendbar*			
	Überbl.	*nicht anwendbar*			
ECR	**Schnitt**	18 %	14 %	13 %	9 %
	Ein-/Ausbl.	27 %	657 %	100 %	526 %
	Überbl.	49 %	37100 %	5500 %	708 %
Kantenkontrast*	**Schnitt**	*nicht anwendbar*			
	Ein-/Ausbl.	*nicht anwendbar*			
	Überbl.	10-35 %	400-8500%	150-1150 %	182-314%

* Ergebnisse sind abhängig vom Threshold

Schnitterkennungstechniken: Vergleich (2)

Ergebnis

- ECR oder Histogramm-basierte Techniken, um Schnitte zu erkennen
- Kantenorientierter Kontrast, um Überblendungen zu erkennen
- Ausblendung problematisch, evtl. Kombination
  - Aufwändige Analyse, Definitionsfragen (Aus-/Überblendung)

Probleme

- Experimentelle Daten müssen manuell analysiert werden
- Definition von Schwellwerten
- Definition von Überblendung/Ausblendung

Aktionsintensität

- **Aktionsintensität:** interessantes Merkmal zur Unterscheidung verschiedener Genre

- **Methoden zur Bestimmung:**
  - **Bewegungsvektoren:** Verwendung des durchschnittlichen Betrags aller Bewegungsvektoren einer Szene
  - **Edge Change Ratio (ECR):** hohe Aktivitäten sind durch hohe ECR-Werte charakterisiert

Analyse von Bildsequenzen

Ziele

- Erkennen von Objekten
- Erkennen der Kamerabewegung (Schwenken, Kippen, Zooming ...)

Merkmal *Objektbewegung*

- weist auf Semantik hin
  - Beispiel: Bewegung vs. Sequenzen ohne Bewegung in Nachrichten
- Erkennung von Bewegung in Verbindung mit Segmentierung
  - menschliche Auge verwendet Bewegungs- und Objektinformation, um Objekte zu erkennen
  - Verfolgen von Objektgrenzen in aufeinanderfolgenden Bildern ergibt höhere Segmentierungsperformanz als Nutzung unbewegter Bilder

Merkmal *Kamerabewegung*

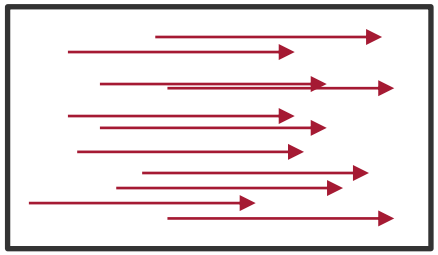
- Unterschied zur Objektbewegung: alle Bildpixel nehmen an der Bewegung teil

Einfluss des Kamerabetriebs

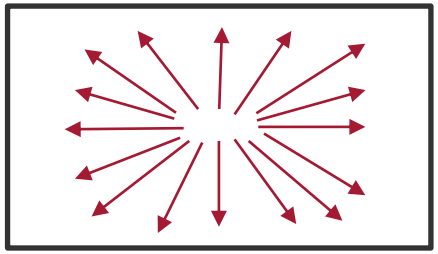
- Erkennen von Schwenken, Kippen, Zooming
- **Bewegung**
  - gilt für alle Pixel in einem Bild
  - in einer einheitlichen berechenbaren Weise
- **Beispiele**
  - **Schwenk:** alle Pixel werden von einer beliebigen Seiten von Bild i zu Bild $(i+1)$ bewegt
  - **Zooming:** alle Pixel – außer denen im Zoom-Zentrum – werden in Kreisen in Richtung auf den Rand des Bildes bewegt $(i+1)$.
  - **Kippen:** Pixel werden halbkreisförmig um Kipppunkt bewegt

Einfluss des Kamerabetriebs

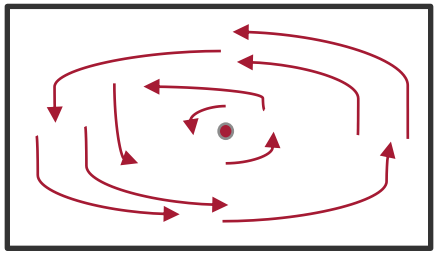
- **Beispiele der Pixelbewegung von Bild i zu i+1**
  - **Schwenk (nach links):**



  - **Zooming (in):**



  - **Kippen (nach rechts 90°):**



Erkennung des Kamerabetriebs

Algorithmus

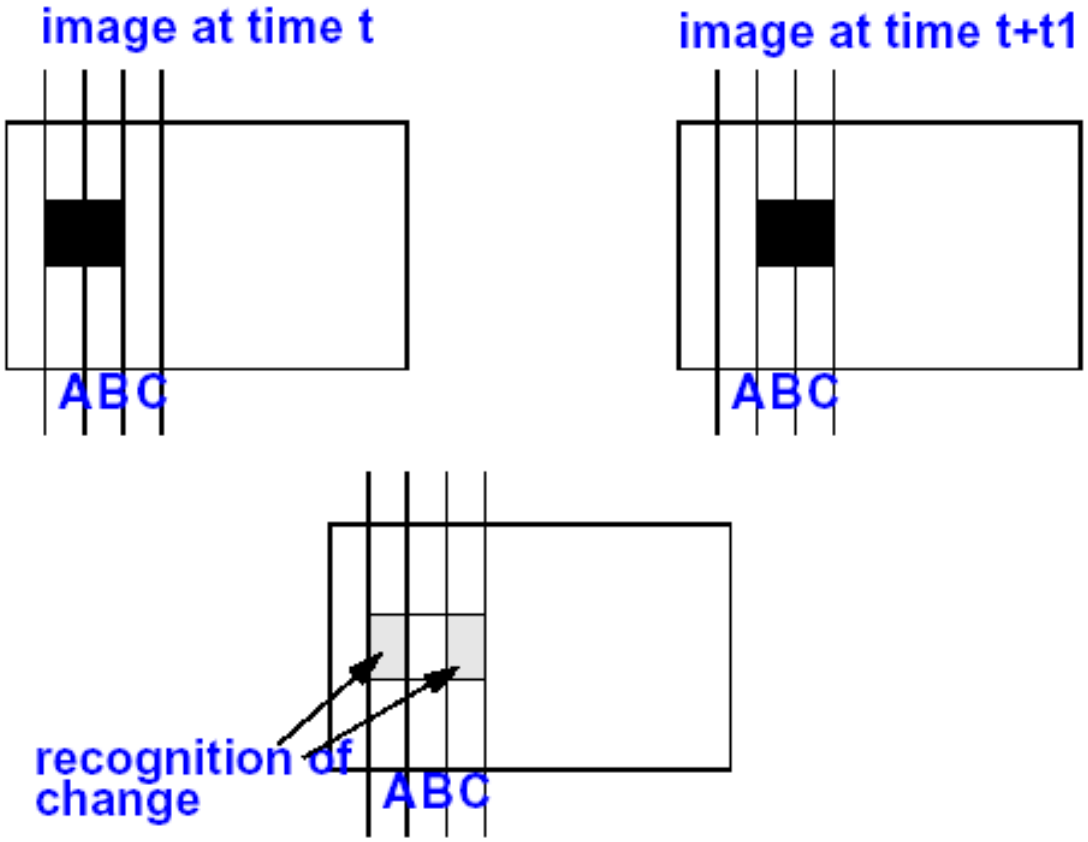
- Verwende Bewegungsvektoren zur Kompression von Algorithmen (MPEG oder H.26x) oder berechne optischen Fluss, um Bewegungsvektoren eines Videos zu erhalten
- Teste, ob Bewegungsvektoren vordefinierten Kamerabetriebsmustern im Hinblick auf absolute Länge und Orientierung entsprechen
  - die meisten Vektoren sind mit derselben Ausrichtung parallelgeschaltet → Schwenk
  - konzentrische Vektoren → Zoom
    - in Richtung auf das Zoom-Zentrum: Zoom-in
    - in Richtung auf den Rand eines Bildes: Zoom-out

Beschränkungen

- Algorithmus funktioniert nicht besonders gut, wenn
  - zu analysierende Szene einen signifikanten Anteil von Objektbewegung enthält
  - Objektbewegung stört die Kamerabewegung und verzerrt automatische Erkennung
  - leider ist Störung ziemlich häufig

Zur Erinnerung: Blockbasierte Bewegungsvektoren

- Blockbasierte Bewegungsvektoren mit beschränkter Anwendbarkeit für semantische Analyse



Optischer Fluss

- Vektorfeld zur Beschreibung von Objekt- oder Kamerabewegung zwischen zwei Bildern

- Effiziente Berechnung der Bewegungsvektoren: Verwendung grauwertige Bilder

Optischer Fluss:

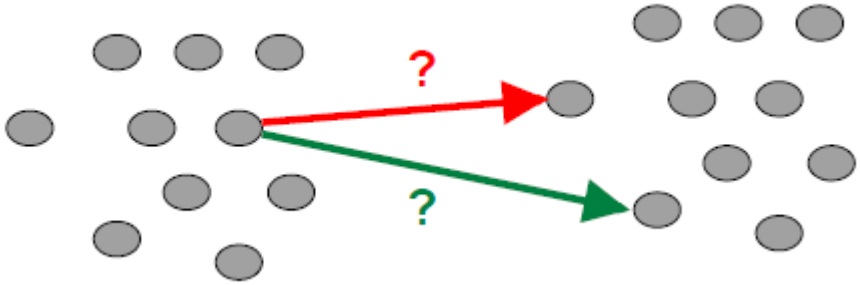
- Bewegung von grauwertigen Mustern über Bildfläche
- 1. Schritt: berechnet Bewegungsvektor jedes grauwertige Pixel
- 2. Schritt: berechnet kontinuierliches Vektorfeld (Interpolation)

Vorgehensweise (Beispiele):

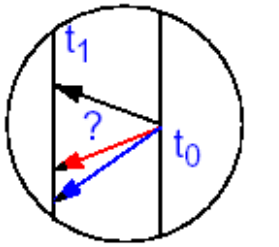
- Differentielle Techniken (Ableitungen von Grauwerten)
- Korrelationsbasierte Techniken (Korrelation von Regionen)

Optischer Fluss: Probleme

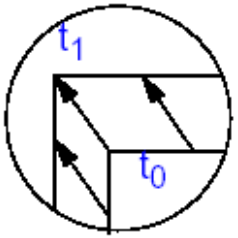
Korrespondenzproblem



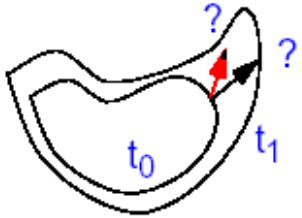
Andere Probleme



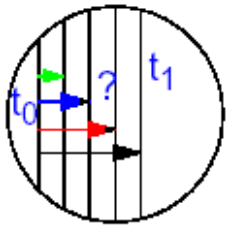
Blendenproblem



Lösung des Blendenproblems



Deformierbare Körper

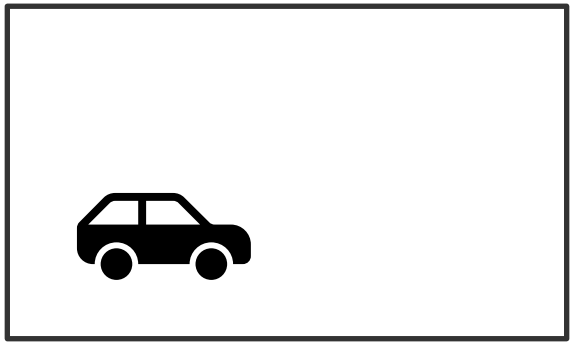


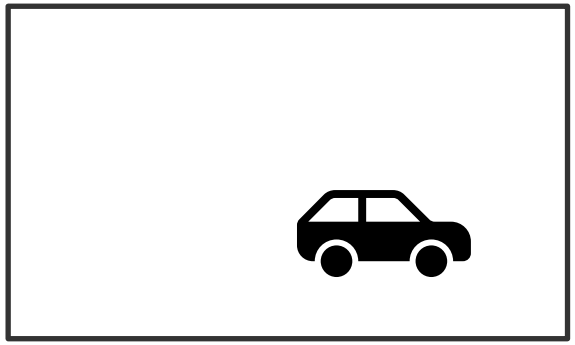
Periodische Strukturen

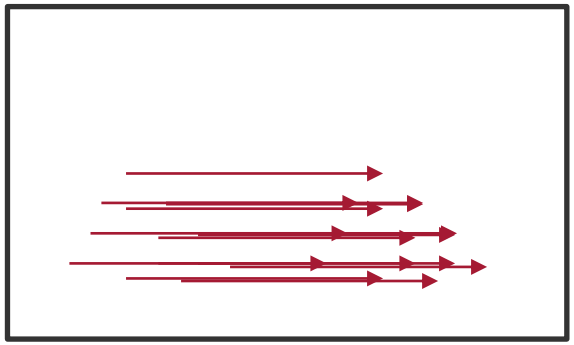
Optischer Fluss ist kein verlässliches Merkmal für Inhaltsanalyse!

Optischer Fluss

- **Vereinfachendes Beispiel:**

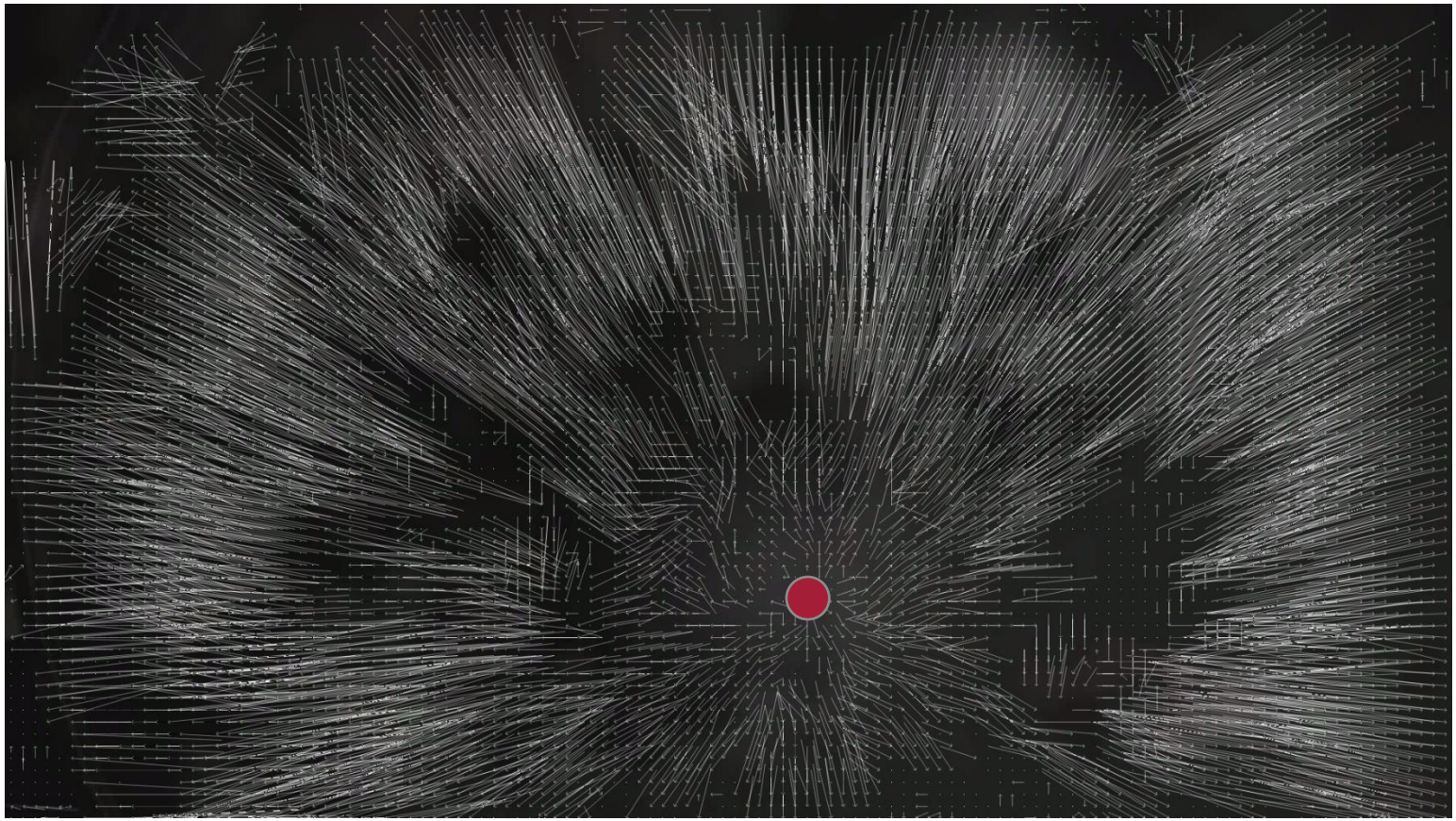






Optischer Fluss

- **Reales Beispiel: Schnelle Fahrt auf markierten Punkt (rot)**



https://de.wikipedia.org/wiki/Optischer_Fluss

Optischer Fluss

- **Verschiedene Algorithmen zur Berechnung**
  - Horn-Schunck-Methode
  - Lucas-Kanade-Methode

    → verschiedene Algorithmen bereits in OpenCV enthalten

- **Weitere Anwendungsgebiete:**
  - Objekttracking im Bereich Autonomes Fahren/Robotik
  - Bildsegmentierung
  - Bewegungserfassung für Mimik-/Gestenerkennung

Szenenanalyse

Schlüsselbilder (Key-Frames)

- repräsentieren Inhalt einer Szene in komprimierter Form
- informieren auf klare Weise den Benutzer über die wichtigsten Aspekte einer Szene
- enthalten Information über ein Objekt, das Teil einer Szene ist
- reduzieren die Bandbreite, die nötig ist, um den Inhalt einer Szene wiederzugewinnen
- benötigen semantisches Verständnis einer Szene, um ausgewählt werden zu können
  - semantisches Verständnis: Gesichtserkennung
- kann in der Videokamera berechnet werden, indem Eigenschaften einer niedrigen Ebene, wie z.B. Farbe und Textur, verwendet werden

Videoähnlichkeit

Prämissen

- Existierendes Material kann in anderen Kontexten wiederverwendet werden
  - **Beispiel:** verwende Fassungen von Reportagen in Nachrichtensendungen (zusammen mit einer neuen Aufnahmesequentialisierung)

- **Vergleichskriterien**
  - **Korrespondenz:**
    - Entsprechung zweier Videosequenzen
    - Evtl. zeitliche Anpassung (temporal alignment) erforderlich
  - **Resequentialisierung:**
    - Wechsel der Szenenreihenfolge in einem Video

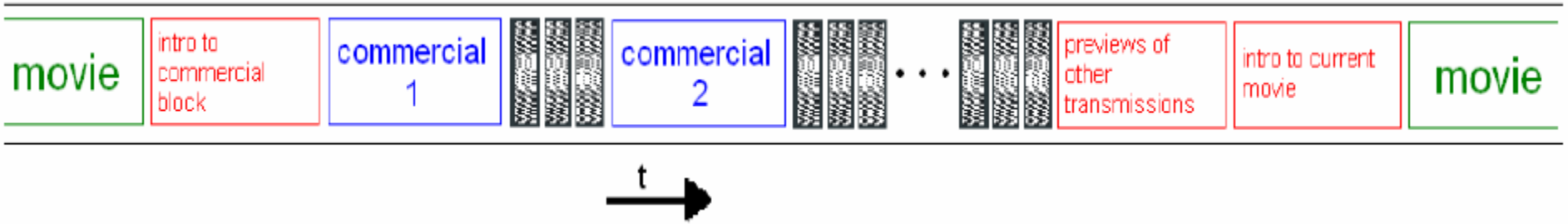
Genre-Erkennung

- **Ziel**
  - Zuweisung eines Genres (Musikvideo, Nachrichten, Sportübertragung, Spielfilm, Werbeclip, …)

- **Technologie**
  - Kombination vieler physikalischer und Bild-Parameter um charakteristische Signatur zu berechnen

Erkennung von Werbeblöcken (ohne Datenbasis)

- **Struktur eines Werbeblocks**



- **Lokalisierung von Werbeblöcken in einem Videostrom**
  - Vorselektion
    - Lokalisiere dunkle monochrome Frames
    - Finde viele harte Schnitte
  - Finde Pegel mit hoher Aktion
    (hoher ECR, große Bewegungsvektoren)

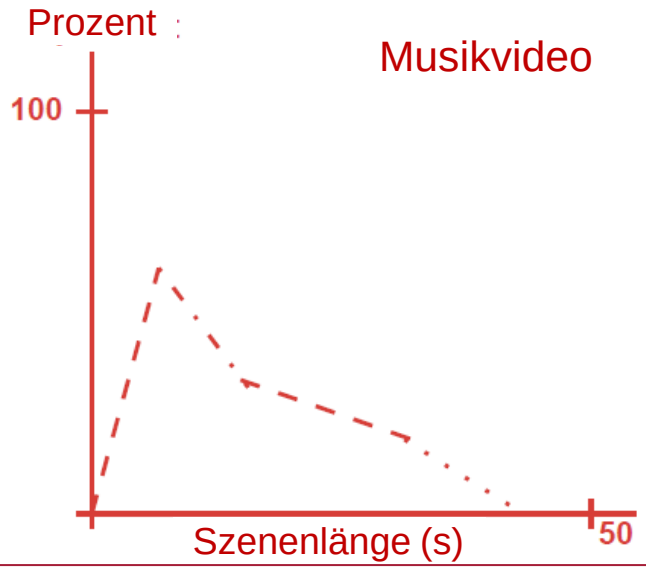
Erkennen des Videogenres

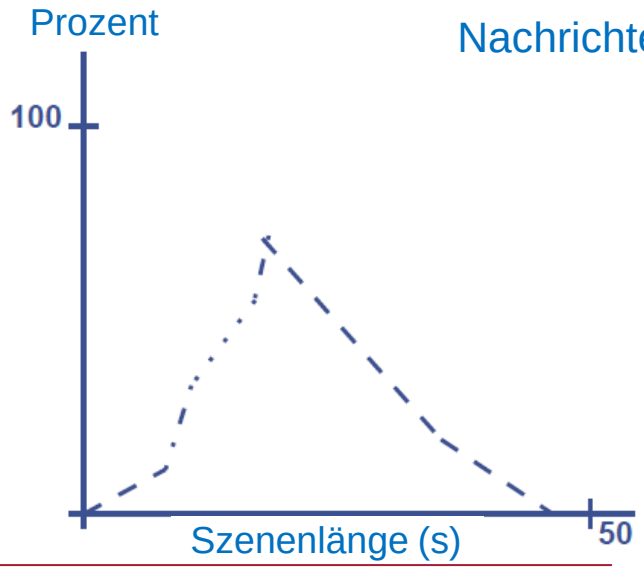
Ziel

- Erkennen eines Genres (Nachrichtensendung, Werbung, Musikclip)

Technik

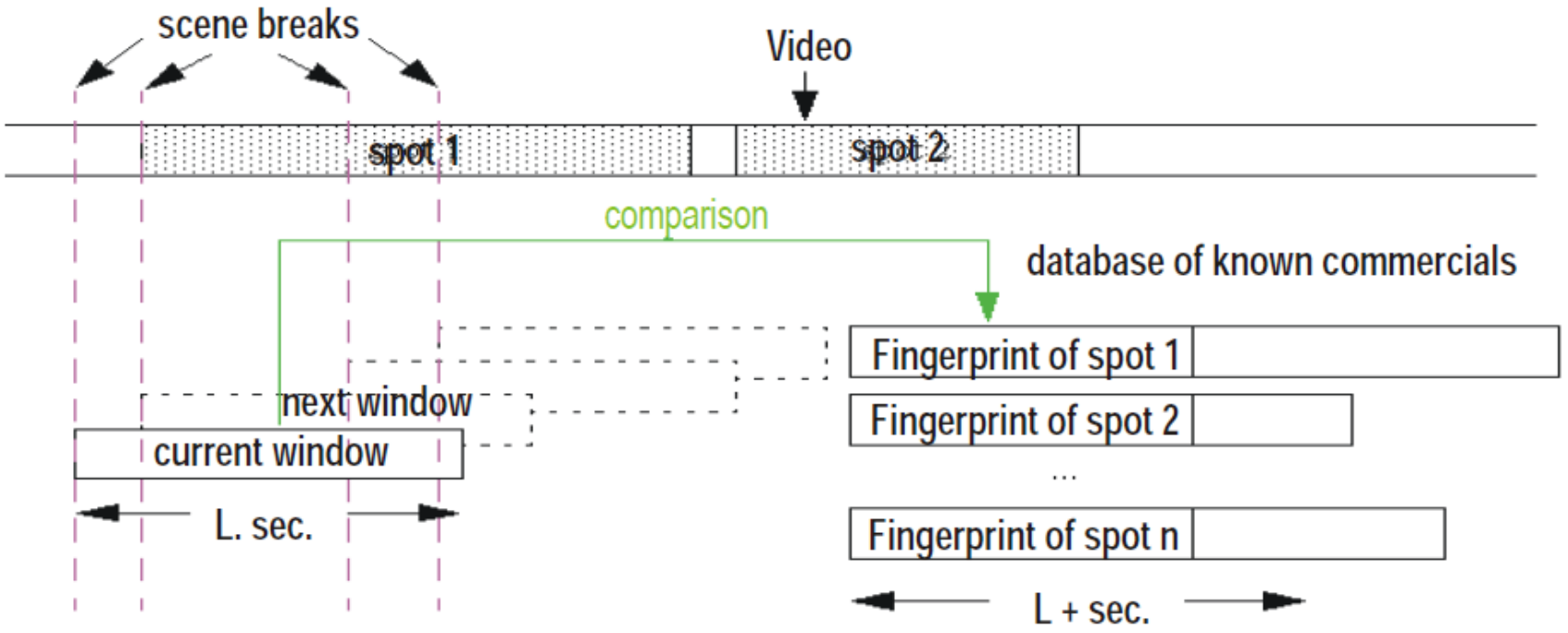
- Verbindung syntaktischer und semantischer Merkmale mit charakteristischem Fingerabdruck
- Erstelle Datenbasis mit Fingerabdrücken
- **Beispiel**





Erkennen einer Nachrichtensendung/Werbeblöcken

- Springe zum nächsten harten Schnitt
- Berechne den Fingerabdruck der nächsten L Sekunden
- Vergleiche Fingerabdruck mit Datenbasis



Erkennen einer Nachrichtensendung/Werbeblöcken

Experimente

- 140 Clips von 7 Genres: Nachrichtensendung, Fußball, Situationskomödie, Musikclip, Zeichentrickfilm, Werbung
- Klassifikation zwischen 87% (Werbung) und 99% (Nachrichtensendung) korrekt

Literatur

K. D. Tönnies:

Grundlagen der Bildverarbeitung,

Pearson Studium, 2005.

B. Jähne:

Digitale Bildverarbeitung,

Springer-Verlag,
6. Auflage 2005.

R. Steinmetz:

Multimedia-Technologie,

Springer-Verlag,
3. Auflage, 2000.

Quellenangabe: Bilder und Folienmaterial sind auszugsweise aus den Lehrbüchern und Materialien von Tönnies, Burger, Burge, Steinmetz und Jähne entnommen.