4 Automatische Inhaltsanalyse von digitalen Videos

4.1 Basis-Parameter für die Video-Analyse

4.2 Ermittlung von semantischen Eigenschaften aus der Video-Spur

4.3 Basis-Parameter für die Audio-Analyse

4.4 Ermittlung von semantischen Eigenschaften aus der Audio-Spur

4.5 Anwendungsbeispiele

Wozu dient die automatische Inhaltsanalyse?

Die erste Generation der Multimedia-Rechner war lediglich in der Lage, Video- und Audioströme weiterzuleiten und auf den Ausgabegeräten darzustellen (Bildschirm, Lautsprecher).

Moderne Multimedia-Rechner erlauben durch ihre hohe Leistungsfähigkeit eine Verarbeitung der multimedialen Ströme.

Ein interessantes Forschungsgebiet ist die **automatische Inhaltsanalyse**. Man versucht, den Rechner so viel wie möglich über den Inhalt eines Videos herausfinden zu lassen. Anwendungsbeispiele sind

- die automatische Indexierung von Video-Archiven, beispielsweise bei Fernsehsendern
- das automatische Filtern von rohem Video-Material auf der Suche nach relevanten Informationen
- die automatische Erstellung von Video-Abstracts
- die Zerlegung und Neukomposition von Videomaterial

4.1 Basis-Parameter für die Video-Analyse

Man kann die Inhaltsanalyse in drei Schritte gliedern:

- die Ermittlung von Basis-Parametern (physikalischen Parametern) aus dem digitalen Datenstrom
- die Berechnung von semantischen Eigenschaften auf einer höheren Abstraktionsebene
- die Zusammensetzung der einzelnen Algorithmen zu Endbenutzer-Anwendungen.

Wir werden dies im Folgenden für Video (Einzelbilder und Bildfolgen) und für Audio skizzieren.

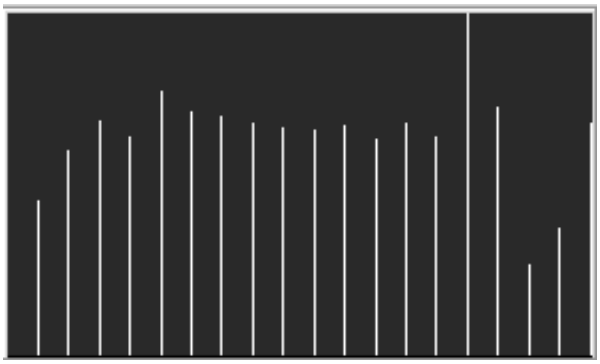
4.1.1 Einzelbild-Analyse

Farbhistogramme

Die einfachste und wichtigste Charakterisierung eines Einzelbildes ist das **Farbhistogramm**. Es stellt die Verteilung von Farbwerten (oder Graustufenwerten) im Bild dar. Für Farbbilder ist das Histogramm dreidimensional (RGB oder YUV), für Graustufenbilder ist es eindimensional.

Das Farbhistogramm wird heute schon vielfach als ein einfacher Filter in Bilddatenbanken eingesetzt.

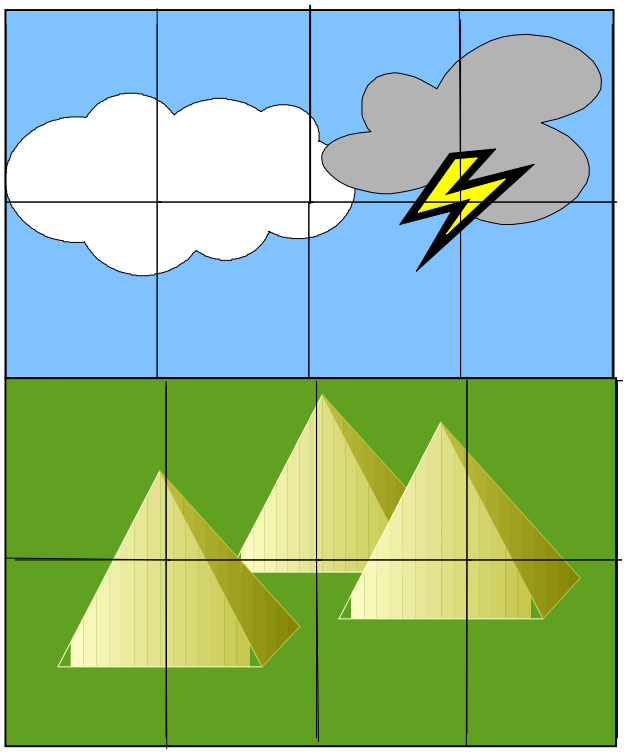
Beispiel: Zwei Graustufenhistogramme



Mosaik-Bildung

Farbhistogramme haben den gravierenden Nachteil, dass sie nicht unterscheiden, wo im Bild und wie gruppiert die Farbwerte in den Bildern vorkommen. Möglicherweise ist ein Bild mit viel Himmel nicht von einem Bild mit viel Meer zu unterscheiden. Oder ein Bild mit einem Sonnenuntergang nicht von einem Bild mit vielen kalifornischen Mohnblumen (orange).

Eine erste Verbesserung besteht darin, das Bild in ein Mosaik von Rechtecken gleicher Größe zu zerlegen und dann zu verlangen, dass die gesuchten Farbwerte in bestimmten Mosaik-Bereichen vorkommen.



Farbkohärenz-Vektoren

Ein Farbkohärenzvektor (color coherence vector, CCV) enthält pro Farbwert zwei Einträge: einen Wert α mit dem Prozentsatz der Pixel in Regionen, die größer als der Durchschnitt sind, einen zweiten Wert β mit dem Prozentsatz der Pixel in Regionen kleiner als der Durchschnitt:

$$CCV = <(\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n)>$$

Mit Farbkohärenzvektoren lassen sich Bildähnlichkeiten präziser fassen. Farbkohärenz-Vektoren sind unabhängig von der Position der Objekte im Bildbereich.

Kantenerkennung

Ein weiterer interessanter Parameter für die Analyse von Bildern sind **Kantenverläufe**. Sie grenzen die einzelnen Objekte eines Bildes voneinander ab. Die Kantenerkennung (edge detection) ist ein wichtiges Gebiet der klassischen Bildverarbeitung.

Es gibt zwei grundsätzliche Möglichkeiten, Kanten im Bild zu finden:

- Linienverfolgung
- iteratives Abgrenzen von Regionen.

Die Erfahrung zeigt, dass das iterative Abgrenzen von Regionen in der Praxis meist besser funktioniert.

Algorithmus “Linienverfolgung“

1. Beginne mit einem Pixel, das auf der gesuchten Kante liegt
2. Für alle Endpunkte von bereits einbezogenen Linien:
   2.1 Untersuche die 1-Pixel-Nachbarschaft des Endpunkts
   2.2 Wenn ein Nachbarpixel eine Farbdifferenz von weniger als Δ_c hat, füge es zu der Linie hinzu,

   bis es keinen Fortschritt mehr gibt.

Wir sehen sofort zwei Probleme:

1. Was ist der richtige Wert für Δ_c?
2. Was passiert, wenn die wahre Linie durch Rausch-Pixel unterbrochen ist? Sollen wir den Suchbereich auf mehrere Pixel erhöhen und interpolieren? Auf wie viele Pixel?

Die Linienverfolgung erweist sich in der Praxis als **unzuverlässig**.

Algorithmus “Regionenausweitung“ (region growing)

1. Die initiale Menge der Regionen ist leer.
2. Finde ein beliebiges Pixel, das noch nicht in einer Region enthalten ist. Dieses Pixel definiert die aktuelle Region.
3. Wiederhole für alle Pixel in der aktuellen Region
   3.1 Untersuche die 1-Pixel-Nachbarschaft des Pixels
   3.2 Wenn ein Nachbarpixel eine Farbdifferenz von weniger als Δ_c hat, füge es zu der Region hinzu

   bis die aktuelle Region nicht mehr wächst.
4. Wenn es noch Pixel gibt, die zu keiner Region gehören, mache weiter mit Schritt 2.

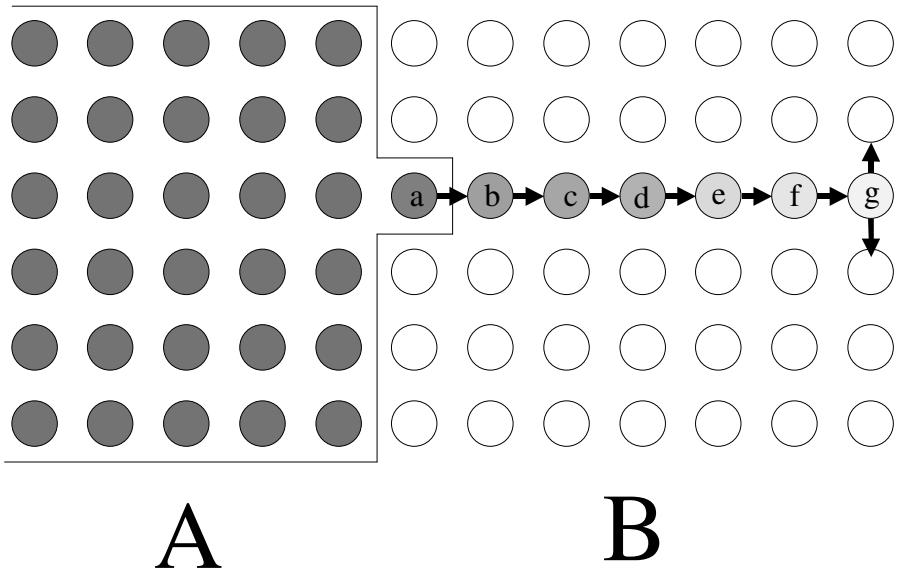
Der Parameter Δ_c ist die **Homogenitätsschwelle** für die Regionen.

Problem mit “Regionenausweitung“

An der Grenze der aktuellen Region kann eine Kette von Pixels mit

$$|c_{i+1} - c_i| \prec \Delta_c$$

die Kante stark verfälschen.



Anmerkung: Es gibt auch den dualen Algorithmus **“region splitting“** sowie eine Kombination der beiden zu **“split-and-merge“**. Letzterer funktioniert in der Praxis oft am besten.

Der Homogenitätsparameter ist kritisch!

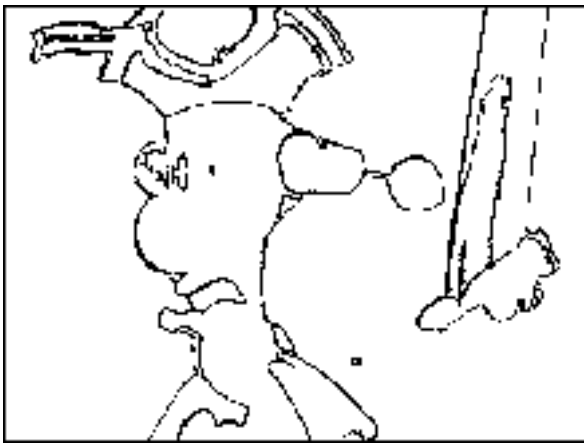
Die richtige Wahl des Homogenitätsparameters Δ_c ist sehr schwierig. Eine falsche Wahl führt zur Über- oder Untersegmentierung:

Originalbild

richtig segmentiert

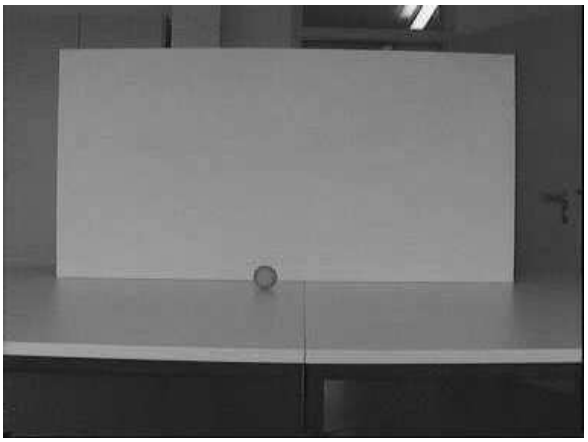
übersegmentiert

untersegmentiert

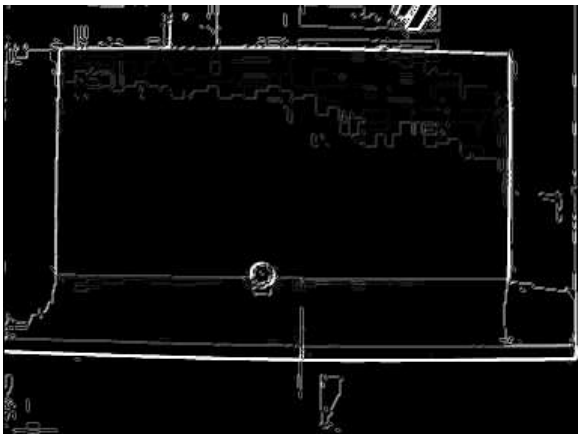


Ein Segmentierungsbeispiel

Eine farbige Kugel rollt vor einem grauen Hintergrund. Sie kann mit Hilfe der Kantenerkennung einigermaßen gut segmentiert werden. Das unten stehende Bild wurde mit dem Algorithmus “Regionenausweitung“ (region growing) berechnet.



Originalbild



segmentiertes Bild

Objektsegmentierung

Die Kantenerkennung bildet die Basis für die Objektsegmentierung. Hierbei versucht man, ein Bild in einzelne semantische Objekte zu zerlegen.

Leider sind nur in sehr einfachen Fällen die durch Kantenzüge eingegrenzten Objekte auch wirklich semantische Objekte des Bildes! Probleme bereiten insbesondere:

- Verdeckungen
- Objekte an den Bildrändern, die nur teilweise sichtbar sind
- Objekte, die sich verformen können (z.B. Personen)
- Objekte, die im 3D-Raum aus verschiedenen Winkeln aufgenommen wurden

und vieles mehr.

Fazit: Die Zerlegung eines Bildes in semantische Objekte ist außerordentlich schwierig.

4.1.2 Bildfolgen-Analyse

Durch die Analyse von Bildfolgen versucht man, Bildinhalte besser zu verstehen.

Die **Bewegung eines Objekts** kann Hinweise auf die Semantik geben, zum Beispiel könnte eine Zickzack-Bewegung charakteristisch für einen Skiläufer beim Abfahrtslauf sein.

Die **Bewegung der Kamera** (Schwenk, Kamerafahrt, Zoom usw.) unterscheidet sich von der Bewegung von Objekten dadurch, dass alle Bildpunkte in berechenbarer Weise davon betroffen sind (zum Beispiel durch Translation bei einem Schwenk). So kann es gelingen, die Kamerabewegung automatisch zu erkennen.

Weiterhin kann die Bewegungserkennung die Segmentierung von Objekten sehr erleichtern. Das menschliche Auge nutzt Bewegungserkennung in hohem Maße zur Objekterkennung; zum Beispiel sieht man einen Spatz in einem herbstliche Laubbaum erst, wenn er sich bewegt.

Bewegungsvektoren

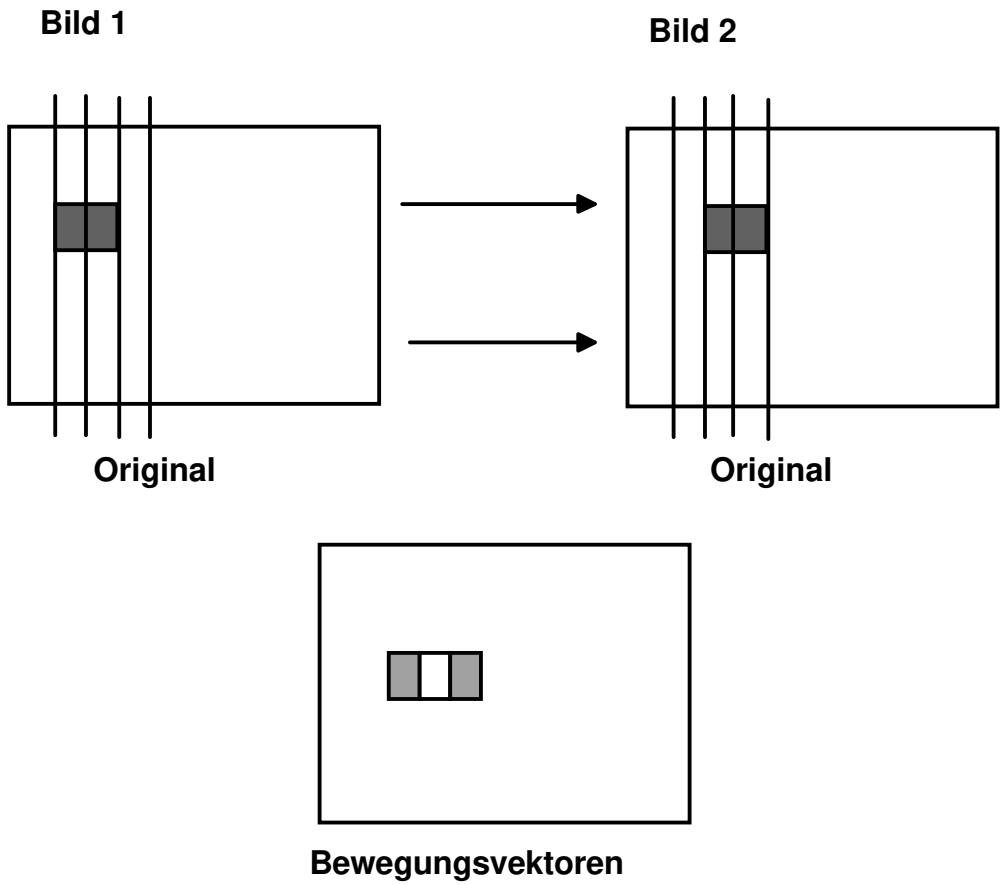
Moderne Kompressionsverfahren für Video berechnen in der Regel **Bewegungsvektoren**, und zwar für Pixelblöcke (Beispiele: MPEG-1, MPEG-2, H.261, H.263). Daraus lässt sich zwar in der Regel eine Bewegung von Einzelobjekten nicht erkennen, aber man kann immerhin Kameraoperationen analysieren. Der Vorteil ist, dass die benötigten Daten ohne eine aufwendige, separate Berechnung zur Verfügung stehen.

Beispiel

Block-basierte Bewegungsvektoren

Wenn ein einfarbiges, flächiges Objekt sich durch das Bild bewegt, funktioniert die Bewegungsdetektion nur an den Kanten des Objekts und auch nur in der Bewegungsrichtung! Deshalb sind block-basierte Bewegungsvektoren für eine semantische Analyse nur eingeschränkt geeignet.

Beispiel

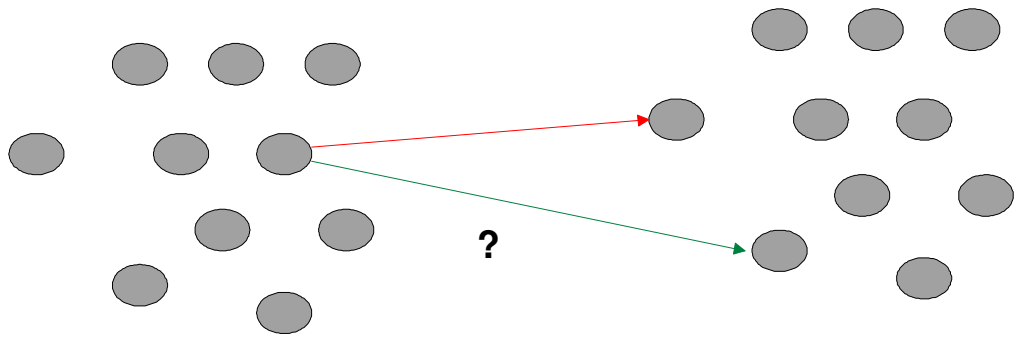


Optischer Fluss

Bewegungen von Objekten der realen Welt stellen sich als Farbänderungen im Bild dar. Um die Berechnung zu vereinfachen, arbeitet man in der Regel auf Grauwertbildern. Unter dem **optischen Fluss** (optical flow) versteht man die Bewegung von Grauwertmustern über die Bildfläche.

In einem ersten Schritt wird an jedem Punkt der Verschiebungsvektor für den Grauwert bestimmt und anschließend ein kontinuierliches Vektorfeld berechnet, das den optischen Fluss darstellt. Beide Schritte sind nur unter einschränkenden Annahmen möglich, und beide sind fehleranfällig. In der Literatur gibt es eine große Zahl von Verfahren zur Berechung des optischen Flusses.

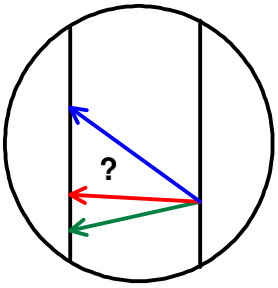
Die größte Schwierigkeit besteht darin festzustellen, wohin ein bestimmter Bildpunkt tatsächlich gewandert ist („physische Korrespondenz“):



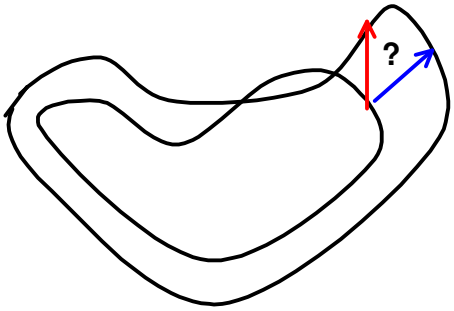
Probleme bei der Berechnung des optischen Flusses

Leider gibt es viele weitere praktische Probleme, die die Berechnung des optischen Flusses erschweren.

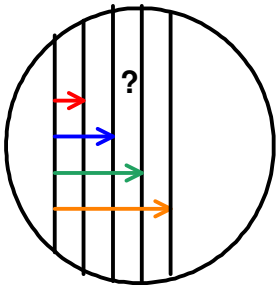
Beispiele



Blendenproblem



deformierbare Körper



periodische Strukturen

Fazit:

Der optische Fluss ist in der Regel ein **unzuverlässiger** Indikator für Objektbewegungen.

Kantenveränderungsrate

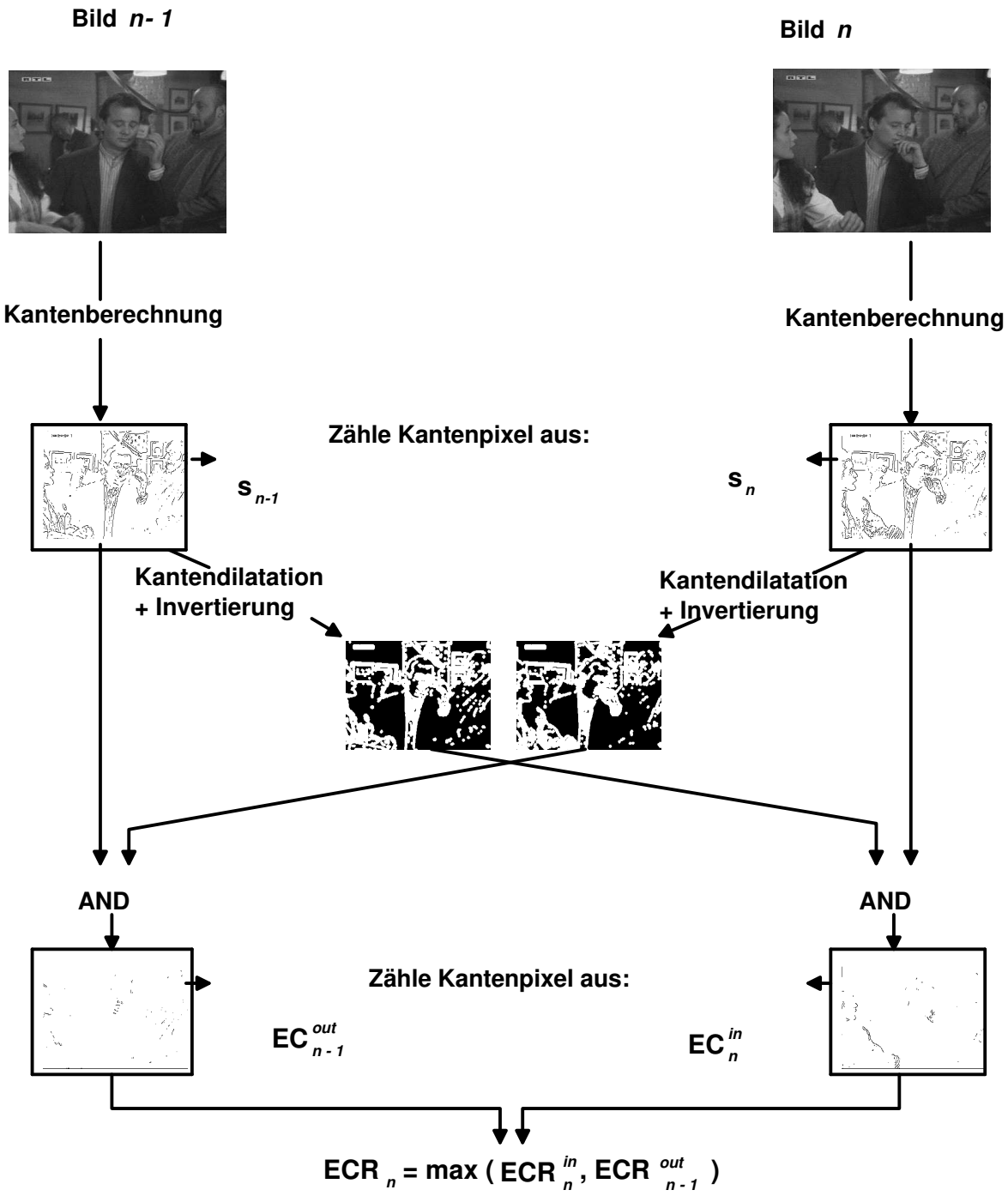
Wenn man die Kanten in einem Bild berechnet, wie weiter oben erläutert, kann man die **Kantenveränderungsrate** (edge change ratio) zwischen zwei Bildern *i* und *i+1* berechnen. Wir ermitteln zunächst die Pixel, die in Bild *i* auf Kanten liegen; ihre Anzahl bezeichnen wir als s_i. Ebenso ermitteln wir die Pixel, die in Bild *i+1* auf Kanten liegen; ihre Anzahl bezeichnen wir als s_{i+1}. Dann ermitteln wir die Anzahl der Pixel, die in Bild *i* auf einer Kante liegen, in Bild *i+1* aber nicht mehr (verschwindende Kanten, E_{out}), und umgekehrt diejenigen, die in Bild *i+1* auf einer Kante liegen, aber noch nicht in Bild *i* (hinzukommende Kanten, E_{in}).

Wir definieren dann die Kantenveränderungsrate ECR (edge change ratio) zwischen Bild *i* und *i+1* als

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

Um die Unempfindlichkeit des Maßes gegenüber leichtem Rauschen oder Bildzittern zu erhöhen, werden die Kanten vor der Berechnung künstlich verbreitert (z.B. auf sechs Pixel). Die ECR kann beispielsweise als ein einfaches Maß für die Intensität von Bewegung eingesetzt werden.

Algorithmus zur Berechnung der ECR



4.2 Ermittlung von semantischen Eigenschaften aus der Video-Spur

4.2.1 Schnitterkennung

Ein sehr einfaches und zugleich zuverlässiges Verfahren zur Ermittlung von Semantik ist die **Schnitterkennung**. Unter einem Schnitt versteht man die Grenze zwischen zwei Einstellungen im Film, in denen jeweils die Kamera ununterbrochen durchläuft.

Man unterscheidet harte Schnitte und Blenden (Einblenden, Ausblenden, Überblenden und Trickblenden wie z.B. Wischblenden).

Die Schnitterkennung dient zugleich dazu, ein Video in einzelne Abschnitte (=Einstellungen) zu zerlegen, denen man dann bestimmte Parameter/ Eigenschaften zuordnen kann. So könnte man zum Beispiel die Einstellungen als atomare Einheiten für Speicherung und Retrieval in einem Video-Archiv ansehen.

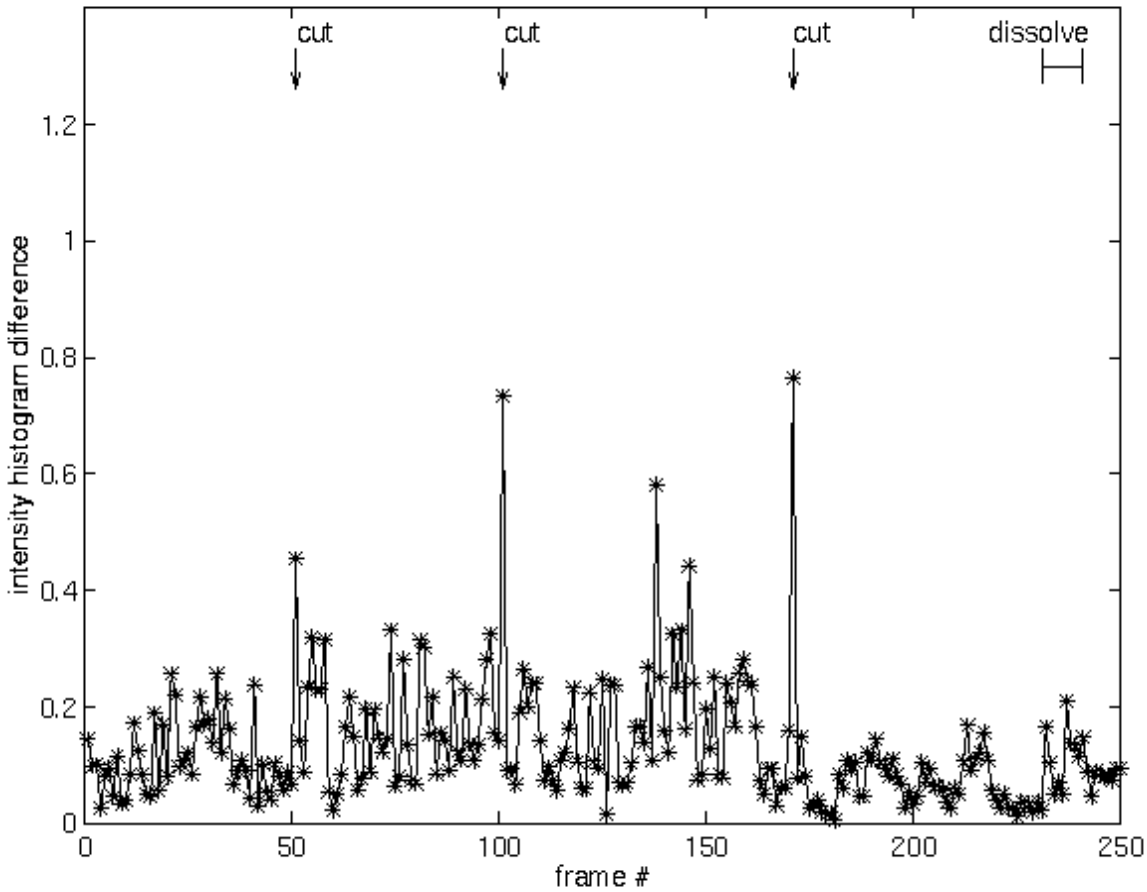
Schnitterkennung mit Farbhistogrammen

Das einfachste Verfahren zur Schnitterkennung basiert auf Farbhistogrammen: Wenn sich die Farbhistogramme zwischen zwei benachbarten Bildern *i* und *i+1* um mindestens einen Schwellwert *T* unterscheiden, wird ein harter Schnitt erkannt.

Sei *H(r,g,b,i)* der Histogrammwert für ein Farbtripel *(r,g,b)* in Bild i. Ein Schnitt wird erkannt genau dann, wenn

$$\sum_{r,g,b} (H(r,g,b,i) - H(r,g,b,i+1))^2 \geq T$$

Beispiel: Schnitterkennung mit Farbhistogramm-Differenzen



Typische Erkennungsfehler

Die Erkennungsrate von harten Schnitten mit Farbhistogrammen liegt in typischen Videos zwischen 90% und 98%.

Sie versagt immer dann, wenn sich die Farbwerte zwischen zwei Bildern plötzlich ändern, ohne dass ein Schnitt im Video vorliegt.

Beispiele

- Einschalten des Lichts in einem Raum
- Explosionen
- gerissene Schwenks

Schnitterkennung mit der Kantenveränderungsrate

Im Allgemeinen werden die Kanten im ersten Bild nach einem harten Schnitt ganz anders verlaufen als im letzen Bild vor dem harten Schnitt. Man kann deshalb im Prinzip die ECR zur Erkennung von harten Schnitten verwenden.

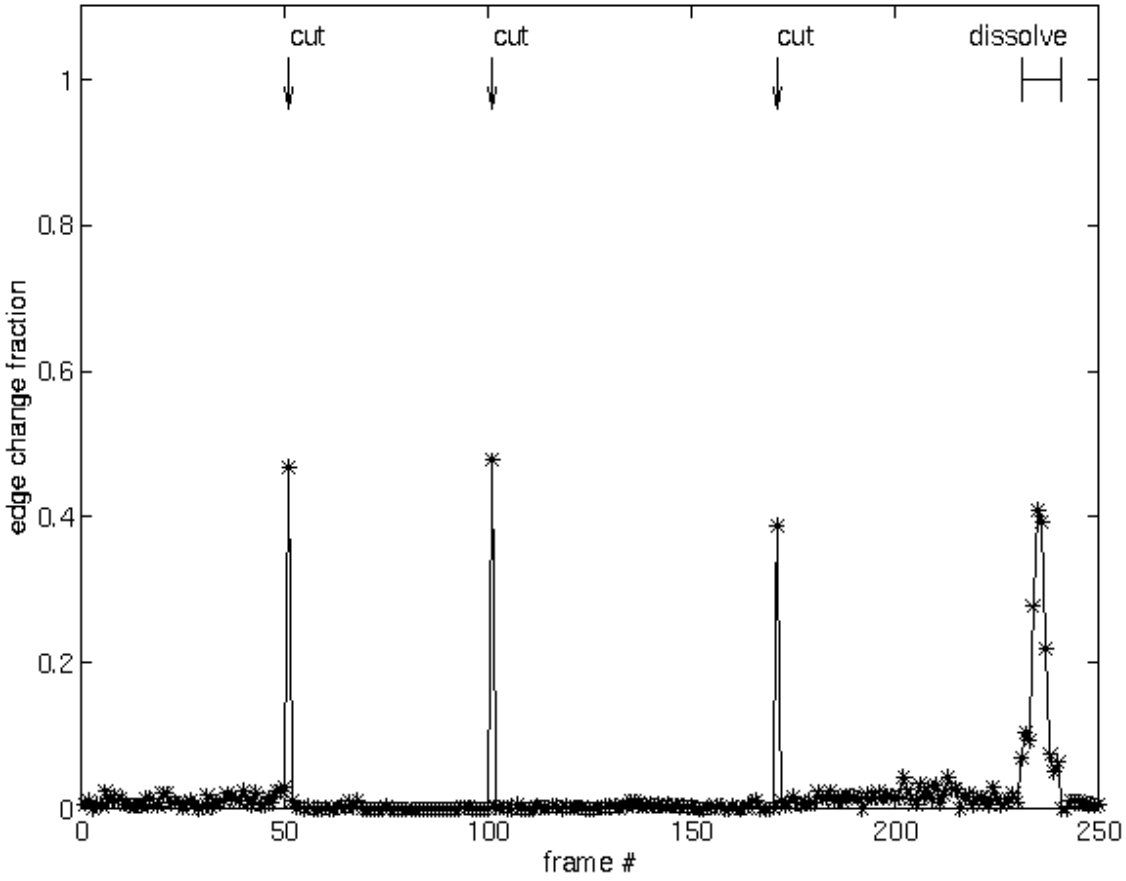
Sei ECR_i die Kantenveränderungsrate zwischen Bild i und Bild $i+1$. Dann wird ein Schnitt erkannt genau dann wenn

$$ECR_i \geq T$$

wobei T ein Schwellenwert ist.

Allerdings muss zum Einsatz dieses Verfahrens zuvor eine Bewegungskompensation auf dem Video gerechnet werden. Denn schnelle Schwenks oder Objektbewegungen in großen Bildbereichen können zu hohen Werten der ECR zwischen benachbarten Bildern führen. Intensive Bewegung kann dadurch von einem harten Schnitt unterschieden werden, dass sie über mehrere Bilder hinweg andauert.

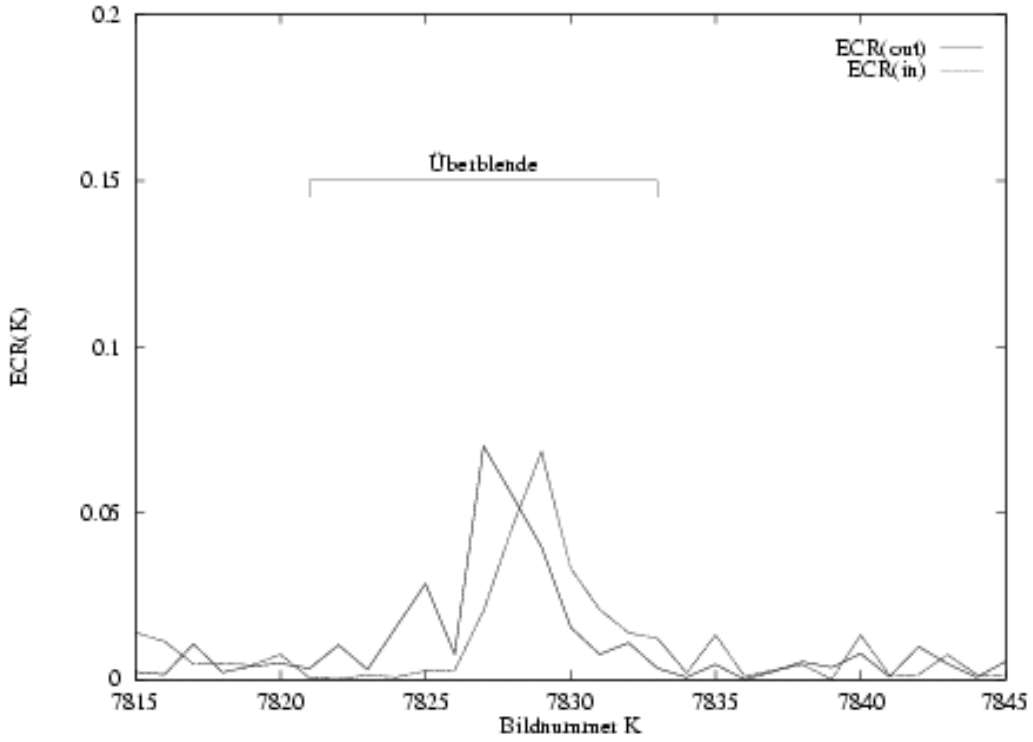
Beispiel: Schnitterkennung mit der Kantenveränderungsrate



Erkennung von weichen Übergängen

Weiche Blenden zwischen Einstellungen sind wesentlich schwerer zu erkennen als harte Schnitte. Man kann beispielsweise versuchen, einen charakteristischen Verlauf der **Kantenveränderungsrate** ECR im Bereich der Blende zu erkennen.

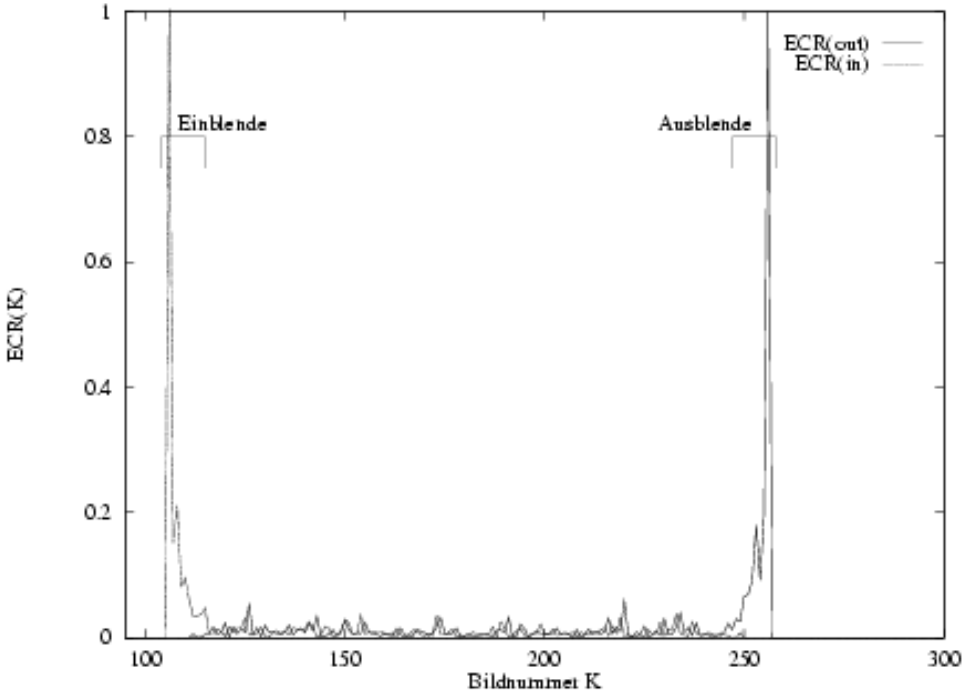
Beispiel: Bei einer **Überblendung** verschwinden zunächst Kanten aus der alten Einstellung mit einer gleichmäßigen Rate; allmählich treten dann Kanten aus der neuen Einstellung hervor. Es ergibt sich ein typischer Verlauf der ECR:



ECR-Verlauf bei Einblenden und Ausblenden

Relativ einfach lassen sich in analoger Weise **Einblenden** und **Ausblenden** im Video lokalisieren. Bei einer Ausblende muss nach dem letzten Bild die Anzahl der Kantenpixel Null sein; bei einer Einblende analog vor dem ersten Bild.

Beispiel: Gemessener Verlauf der ECR beim Ein- und Ausblenden



Wie man sich leicht überlegen kann, sind weiche Übergänge in **Farbhistogrammen** derart häufig **innerhalb** von Einstellungen zu finden, dass man sie nicht als charakteristisch für eine Blende ansehen kann.

4.2.2 Action-Intensität

Die Intensität von Action in einer Einstellung ist ein wichtiger Parameter, der zum Beispiel bei der Genre-Erkennung zur Unterscheidung von Nachrichtensendungen und Musik-Videoclips verwendet werden kann.

Die Action-Intensität lässt sich sehr einfach aus den **Bewegungsvektoren** berechnen: Man berechnet den durchschnittlichen Betrag aller Vektoren über die Länge der Einstellung. Dabei werden sowohl Objektbewegung als auch Kamerabewegung erfasst.

Ebenso kann die **Kantenveränderungsrate** ECR als Indikator für Action dienen. Lange statische Szenen haben eine niedrige ECR, bewegungsintensive Szenen eine hohe ECR.

4.2.3 Erkennung von Kamera-Operationen

Unter Kameraoperationen versteht man **Schwenks, Kamerafahrten und Zooms.** Diese können dadurch von Objektbewegungen unterschieden werden, dass sie sich in einheitlicher, berechenbarer Weise auf die einzelnen Pixel eines Bildes auswirken.

Beispiel 1
Bei einem Schwenk werden alle Pixel beim Übergang von Bild *i* auf Bild *i+1* um denselben Betrag seitlich verschoben.

Beispiel 2
Beim Hineinzoomen werden alle Pixel außer dem Bildmittelpunkt von diesem weg nach außen verschoben. Es ändert sich zwar auch die Größe der Objekte, aber zwischen Bild *i* und Bild *i+1* kann dies bei der Berechnung vernachlässigt werden.

Vorgehensweise zur Erkennung von Kamera-Operationen

Algorithmus **Erkenne-Kamera-Operation**

- Verwende die Bewegungsvektoren aus dem Kompressionsalgorithmus (z.B. MPEG) oder berechne den optischen Fluss im Video
- Teste, ob die ermittelten Vektoren nach Betrag und Richtung dem Muster einer vordefinierten Kamera-Operation entsprechen

Anmerkung

Die Erkennung von Kamera-Operationen funktioniert nur dann gut, wenn die analysierte Einstellung wenig Objektbewegung enthält. Bei einer (in der Praxis sehr häufigen) Überlagerung von Kamera-Operation und Objektbewegung ist eine automatische Erkennung kaum noch möglich.

Ermittlung von Raumgeometrien aus Kamerabewegung

Wenn man die Kamerabewegung durch die Analyse der Bewegungsvektoren rückberechnen kann, kann man aus dem Betrag und der Richtung der Vektoren die Geometrie des Raumes rekonstruieren. Sehr leicht lässt sich zum Beispiel aus einem horizontalen Schwenk ein Panorama-Standbild erzeugen.

Beispiel

Berechnung von Panoramabildern aus einem Video, das von einer ständig auf dem Kopf getragenen Kamera aufgezeichnet wird (Steve Mann, MIT Media Lab)

4.2.4 Texterkennung

Ziel

Extraktion von generiertem Text (evtl. auch Szenentext).
Grund: reich an Semantik

Technik

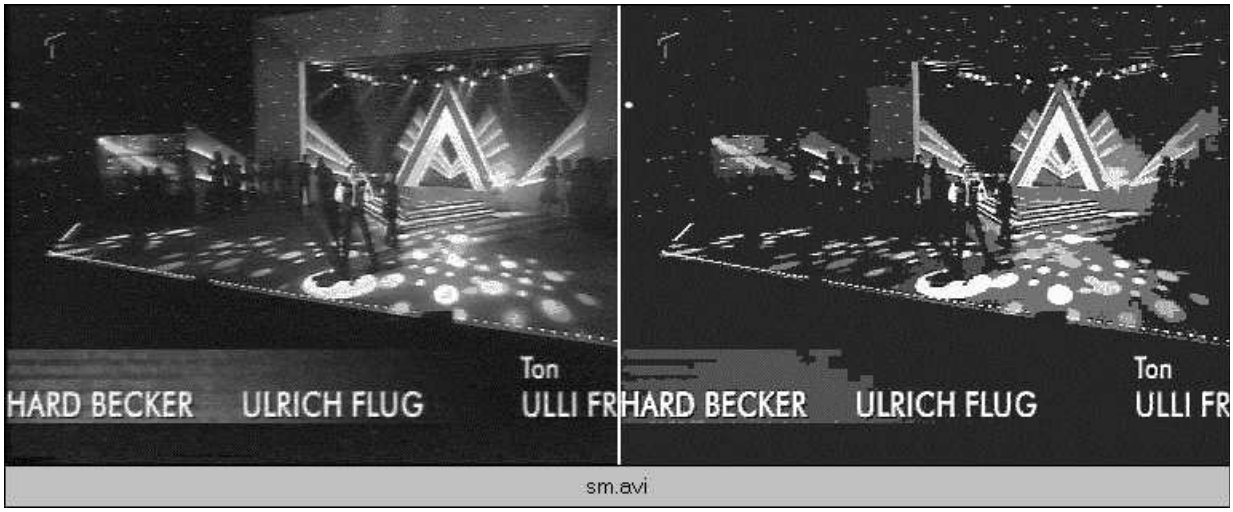
Erkennung von Textregionen, Ausschneiden, OCR

Eigenschaften von Text in Videos

- monochrom
- starr
- im Vordergrund
- minimale und maximale Größe
- stationär oder linear durchlaufend
- hoher Kontrast zum Hintergrund
- erscheint wiederholt in aufeinander folgenden Bildern
- erscheint in Gruppen

Textsegmentierung (1)

Text ist monochrom



Original-Videobild **Zerlegung in Regionen**

Textsegmentierung (2)

Grenzwerte für Zeichengrößen und Kontrast

bisheriges Ergebnis **nach Anwendung der Grenzwerte**

Textsegmentierung (3)

Text muss stillstehen oder sich linear bewegen

bisheriges Ergebnis

nach Anwendung der Bewegungsregel

Experimentelle Ergebnisse

Testvideo	Anzahl Bilder	Anzahl Buchstaben	davon segmentiert
Titelsequenzen	7372	6423	99%
Werbespots	6858	1065	99%
Nachrichten	18624	1411	97%

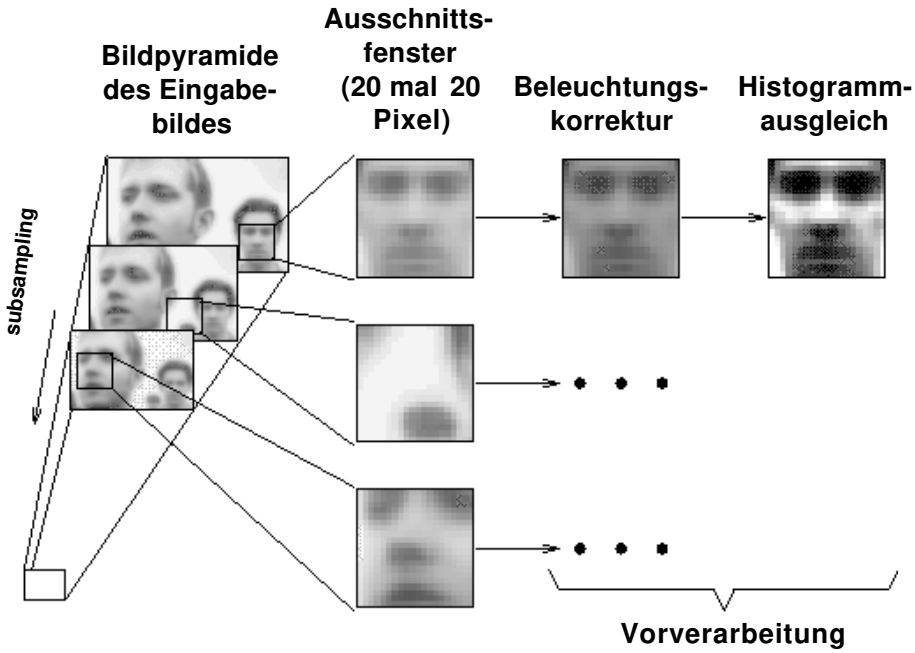
4.2.5 Gesichtserkennung

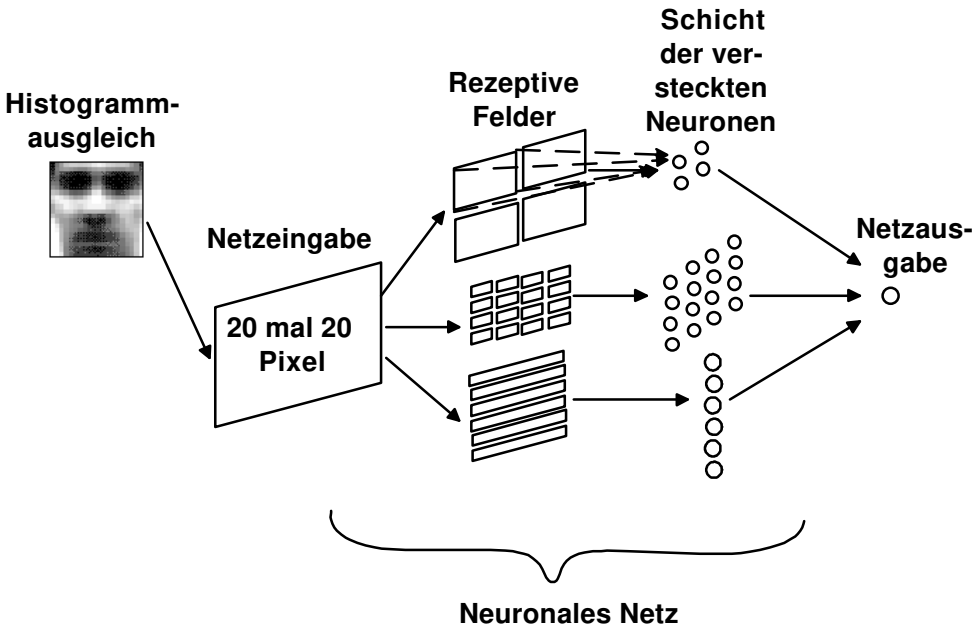
Ziel:
Erkennung von Bildbereichen im Video, die ein Gesicht in Frontalansicht zeigen.

Ansatz:

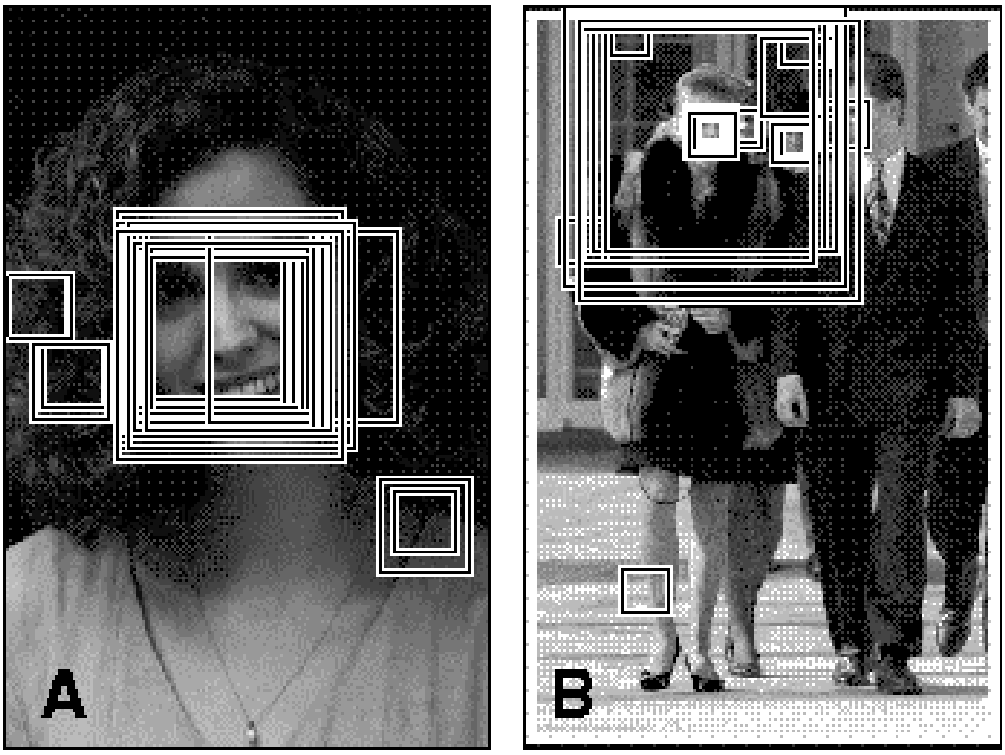
- Aufbau eines neuronalen Netzes zur Mustererkennung
- Trainieren des Netzes mit einigen tausend Gesichtern, bei denen die Linie zwischen den Augen und die Senkrechte von dort auf die Nasenspitze markiert wurden
- Durchlaufen eines unbekannten Bildes mit Auschnittsrahmen in verschiedenen Größen
    - Vorverarbeitung/Filtern/Helligkeitsnormierung des Ausschnitts
    - Test auf “Gesicht“ mit dem trainierten neuronalen Netz

Funktionsweise der Gesichtserkennung





Mehrfach-Erkennung



Visualisierung des Ergebnisses

4.3 Basis-Parameter für die Audio-Analyse

Physikalische Eigenschaften: einfach

Amplitude = Lautstärke

Frequenz = Tonhöhe

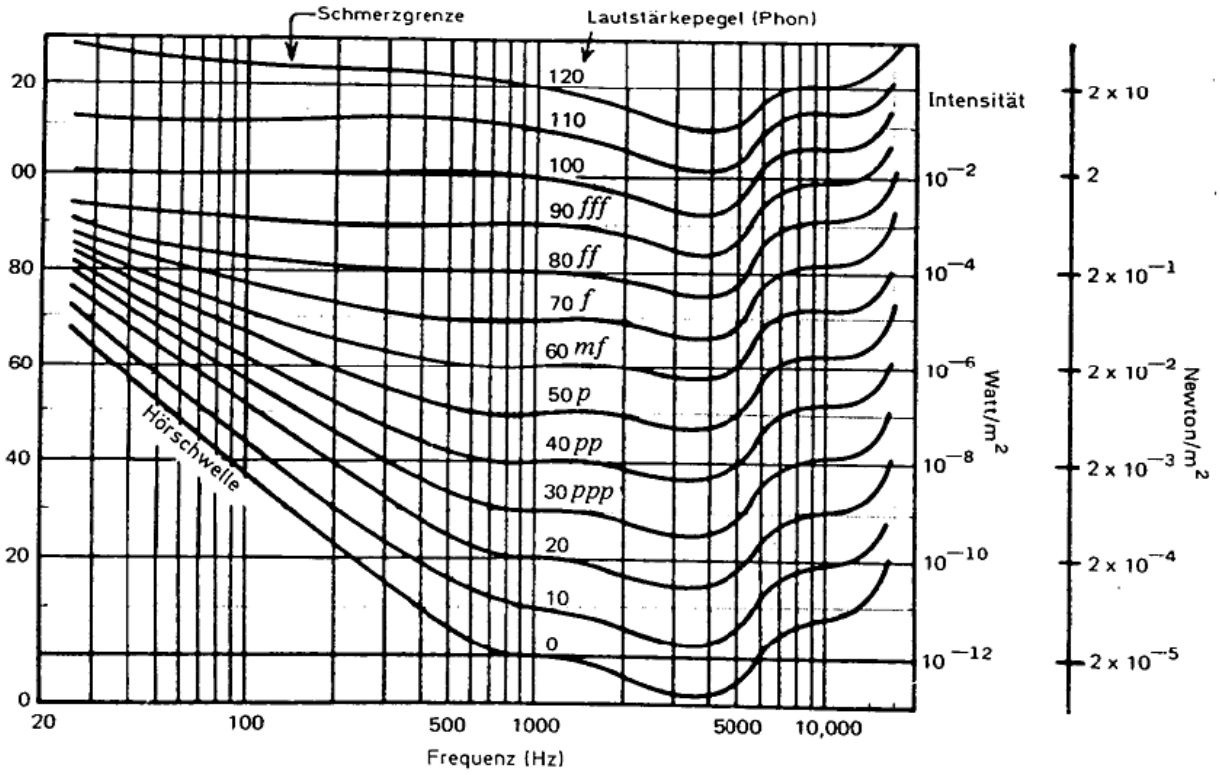
Psycho-akustische Eigenschaften: komplex

- Klang entsteht durch eine komplexe Überlagerung von verschiedenen Frequenzen
- wichtig für den akustischen Eindruck ist auch das Einsetzen und das Abklingen eines Klanges (z.B. Klavierton vs. Gitarrenton)

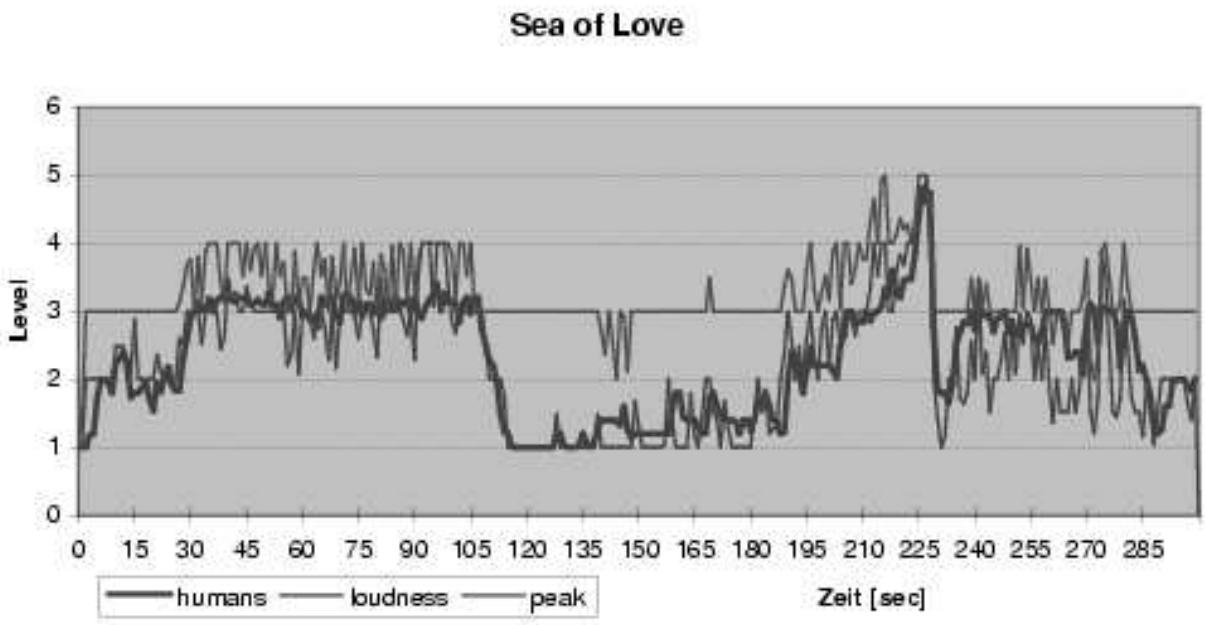
Empfindung der Lautstärke

1. **Das physikalische Maß: der Schalldruckpegel**
   Einheit: Dezibel [db]
2. **Durch den Menschen empfundene Lautstärke**
   Einheit: Phon [phon]

Die Kurvenschar der Isophone stellt die Abhängigkeit der empfundenen Laustärke von der Frequenz dar:



Experimentelle Ergebnisse

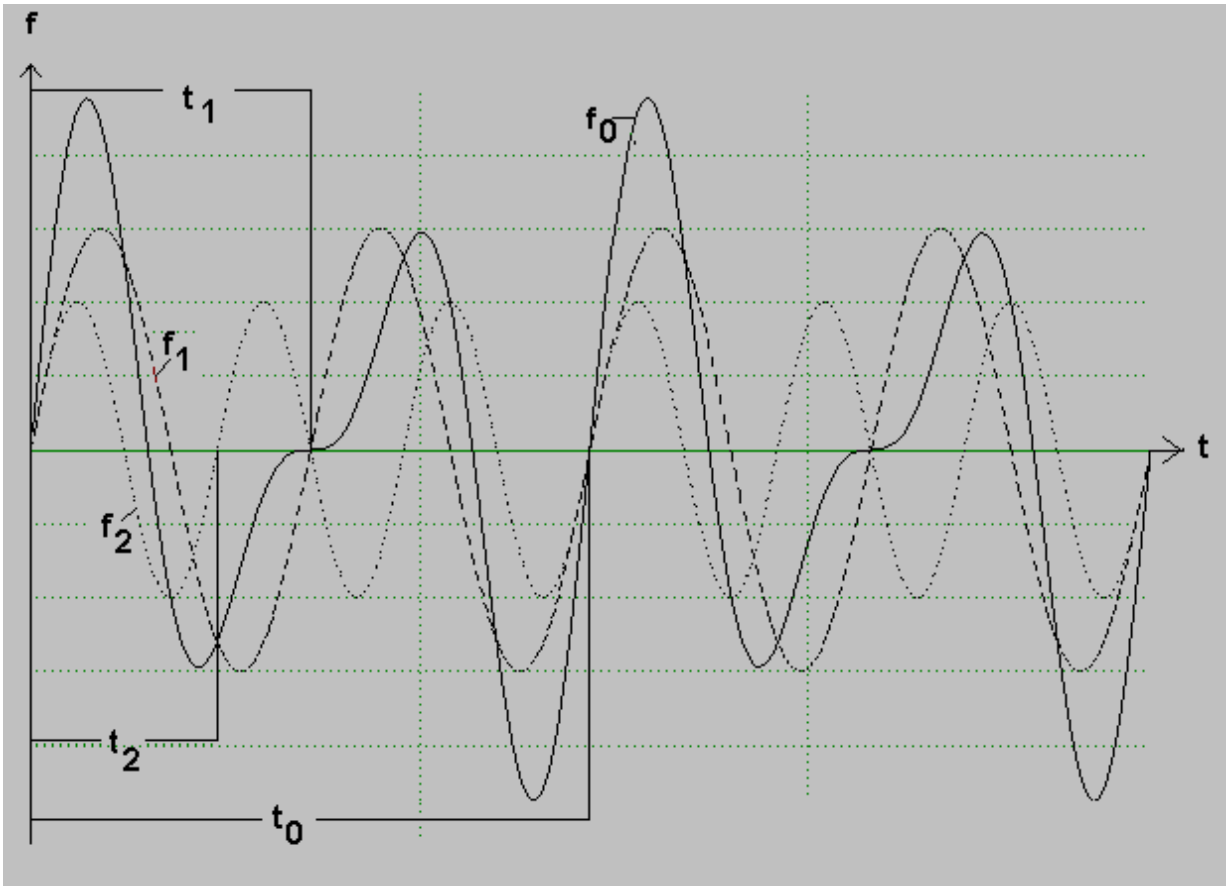


rote Kurve: Schalldruckpegel

blaue Kurve: rechnerische vorhergesagte Lautstärke-Empfindung

schwarze Kurve: von Kontrollpersonen tatsächliche empfundene Lautstärke

Fundamentale Frequenzen in harmonischen Klängen



Die Grundfrequenz (fundamental frequency) des zusammengesetzten Klangs f_0 entspricht dem kleinsten gemeinsamen Vielfachen der Einzelfrequenzen f_1 und f_2.

Frequenztransformation

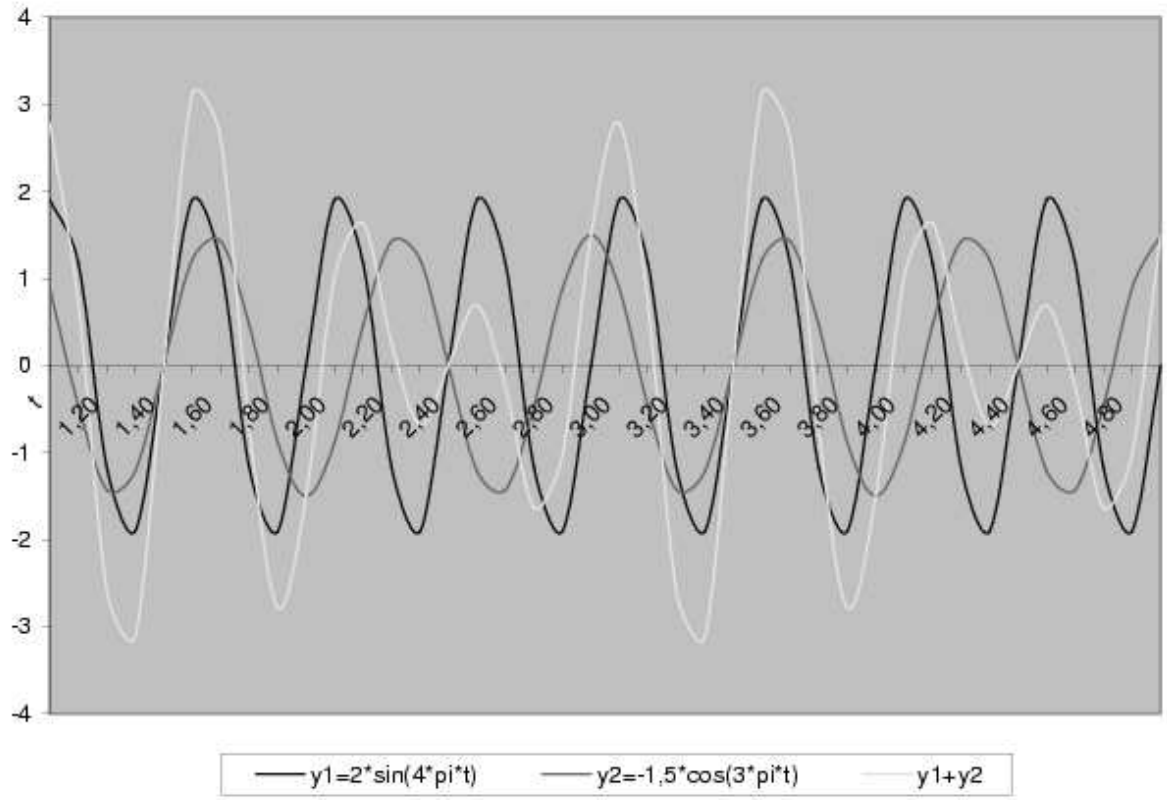
J.B.J. Fourier (1768-1830): Jede Schwingung kann als Summe harmonischer Schwingungen dargestellt werden:

$$s(t) = \frac{B_0}{2} + \sum_{n=1}^{\infty} [A_n \sin(2\pi nft) + B_n \cos(2\pi nft)]$$

f: Grundfrequenz

A_n, B_n: Amplituden

sin(2π nft) = *ganzzahlig Vielfache der Grundfrequenz*



Frequenztransformation auf digitalen Signalen

Instrument:

N-Punkt diskrete Fouriertransformation (DFT)

$$S(f) = \sum_{n=0}^{N-1} s(n) e^{-if\frac{2\pi}{N}n}, f = 0,1,\ldots,N-1$$

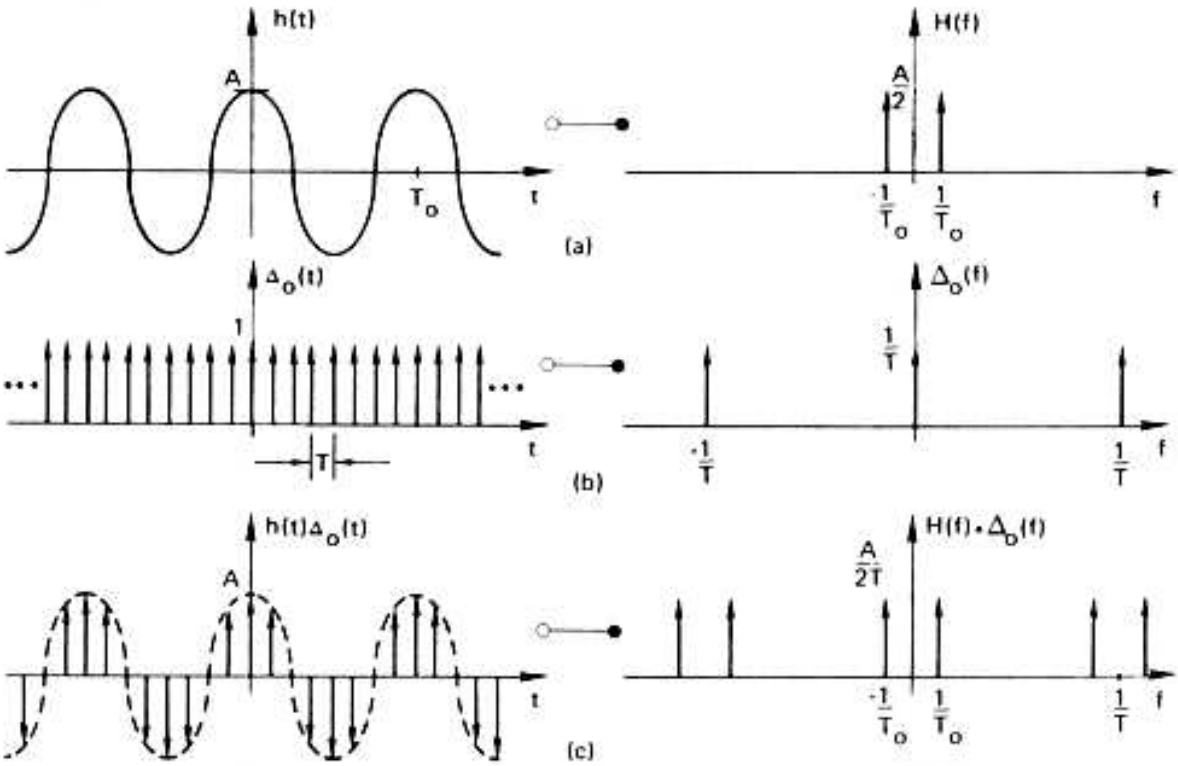
N = DFT-Länge

Schema

	s(t)	kontinuierliches Ausgangssignal
Schritt 1		Abtastung mit $f_s = \frac{1}{T}$
	s(t)	Diskretes Ausgangssignal
Schritt 2		Zeitbegrenzung mit w(t)-Fenster
	s(t)	Diskretes Ausgangssignal mit N Werten [0, NT]
Schritt 3		N-Punkt DFT
	S(f)	Kontinuierliche Fouriertransformierte
Schritt 4		Abtastung mit N Stützpunkten pro T
	S(f)	Diskrete Fouriertransformierte

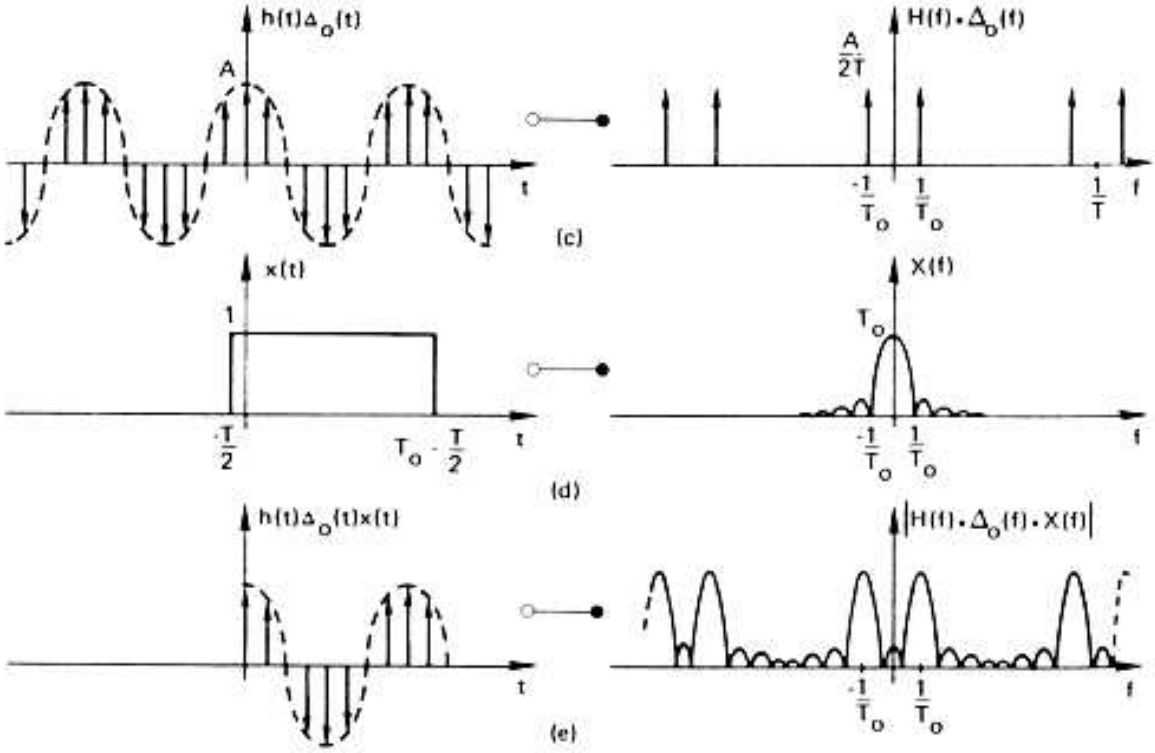
Die Schritte 3 und 4 werden durch die FFT (Fast Fourier Transform) erheblich beschleunigt.

Schritt 1: Abtastung im Zeitbereich



Zeitbereich **Frequenzbereich**

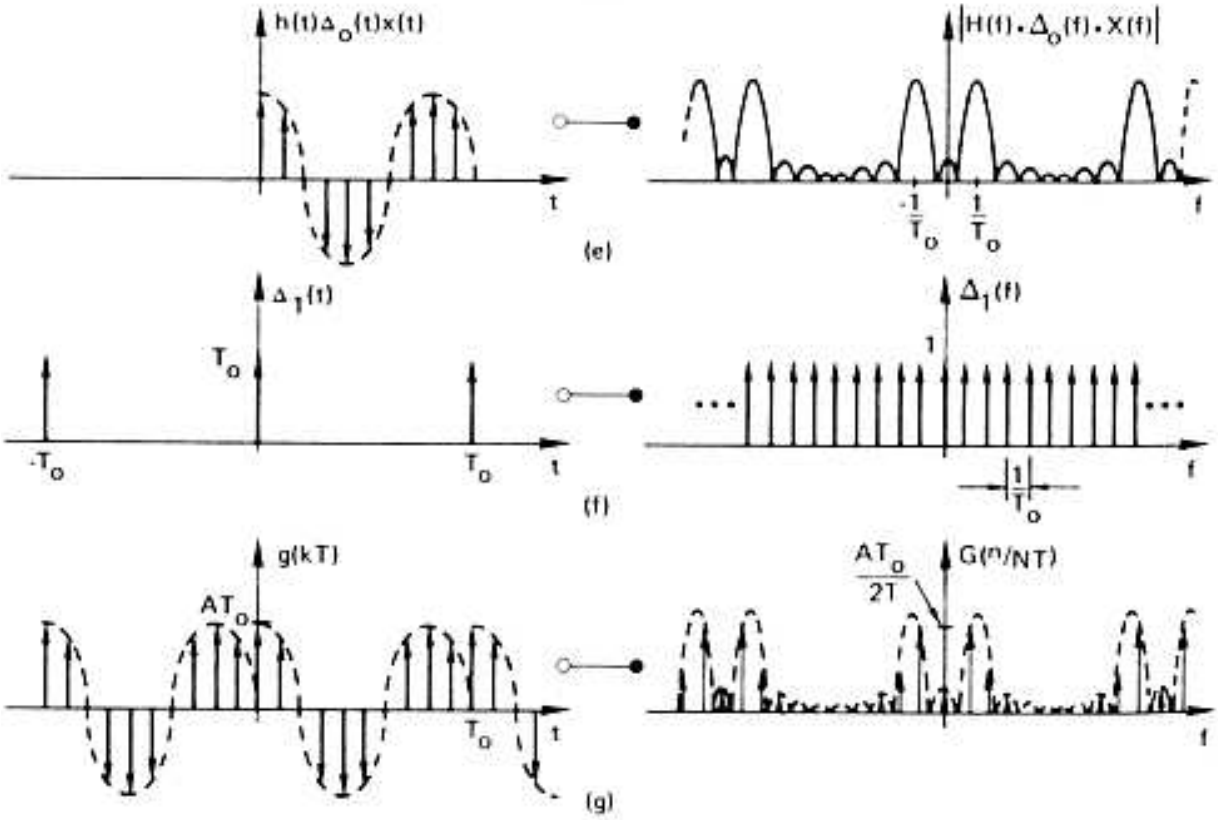
Schritt 2: Zeitbegrenzung auf [0, *NT*]



Zeitbereich **Frequenzbereich**

Schritt 3: Abtastung im Frequenzbereich

Ziel: Digitalisierung der Daten auch im Frequenzbereich zwecks Darstellung auf dem Rechner



Zeitbereich **Frequenzbereich**

Literatur:
E.O Brigham: *FFT: Schnelle Fourier-Transformation,* Oldenburg Verlag 1995

Signalanalyse mit der DFT

Gegeben

Natürliches Audiosignal der Länge M, z.B. M = 5 min monophones Musiksignal

Ziel

Extraktion von Eigenschaften, z.B. musikalischeTöne (Tonhöhe, Lautstärke, Anklingzeit, Abklingzeit)

Methode

Festlegung eines Rahmens der Größe *N*, der zur Analyse über das Audiosignal verschoben wird und jeweils ein „Fenster“ auf dieses bildet. Auf diesem Rahmen wird die DFT durchgeführt.

Im Beispiel: Da Töne auf mindestens 10 ms stationär bleiben, wird *N* = 10 ms gewählt.

Die Verschiebung des Rahmens geschieht überlappend, um Tonübergänge besser erfassen zu können. Im Beispiel werde gewählt: Überlappung = 2 ms

=> es gibt $\frac{5x60x100}{8} = \frac{30.000}{8} = 3.750$ Rahmen.

Signalanalyse – Eigenschaften (1)

Auf den Rahmen können semantische Eigenschaften berechnet werden.

1. Energie

$$E_s(m) = \sum_{n=m-N+1}^{m} s^2(n)$$

m = Endezeitpunkt des Rahmens

E_s ist ein Maß für die akustische Energie des Signals auf dem Rahmen. Es entspricht dem Quadrat der Fläche unter der Kurve im Zeitbereich.

Die Energie kann auch auf dem frequenztransformierten Signal berechnet werden und stellt dann ein Maß für die **spektrale Energieverteilung** dar.

Signalanalyse – Eigenschaften (2)

2. Nulldurchgangsmaß

$$sign(s(n)) = \begin{cases} 1: & s(n) \geq 0 \\ -1: & s(n) \prec 0 \end{cases}$$

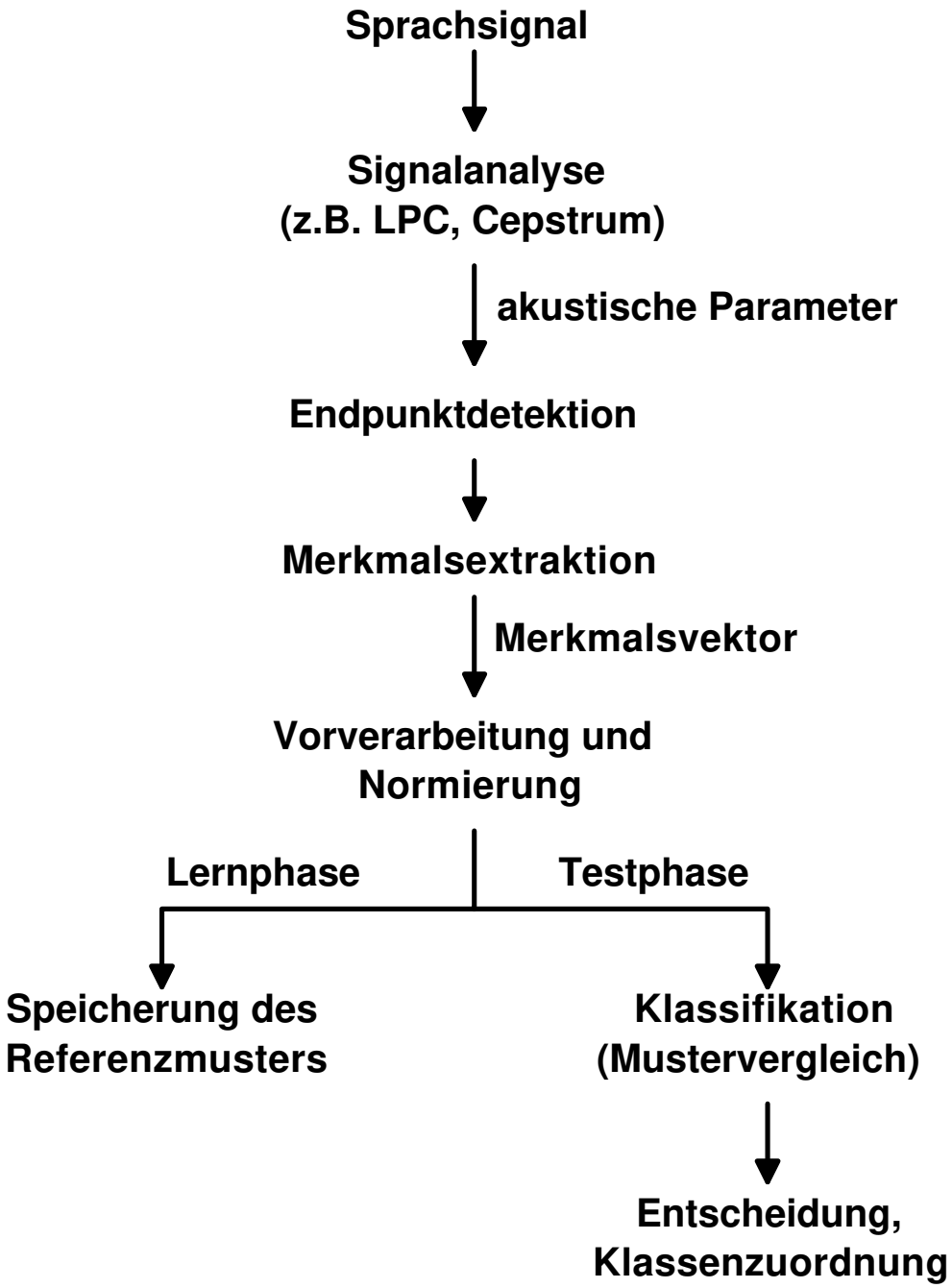
$$Z_s(m) = \frac{1}{N} \sum_{n=m-N+1}^{m} \frac{\left| sign(s(n)) - sign(s(n+1)) \right|}{2}$$

- Zählt die Anzahl der Nulldurchgänge (bzw. Vorzeichenwechsel) des Signals
- Hohe Frequenzen führen zu hohem Z_s, niedrige Frequenzen zu niedrigem Z_s
- Eng verwandt mit der Grundfrequenz.

Es gibt noch viele weitere geeignete syntaktische Eigenschaften aus der Signalanalyse.

4.4 Ermittlung von semantischen Eigenschaften aus der Audio-Spur

4.4.1 Spracherkennung

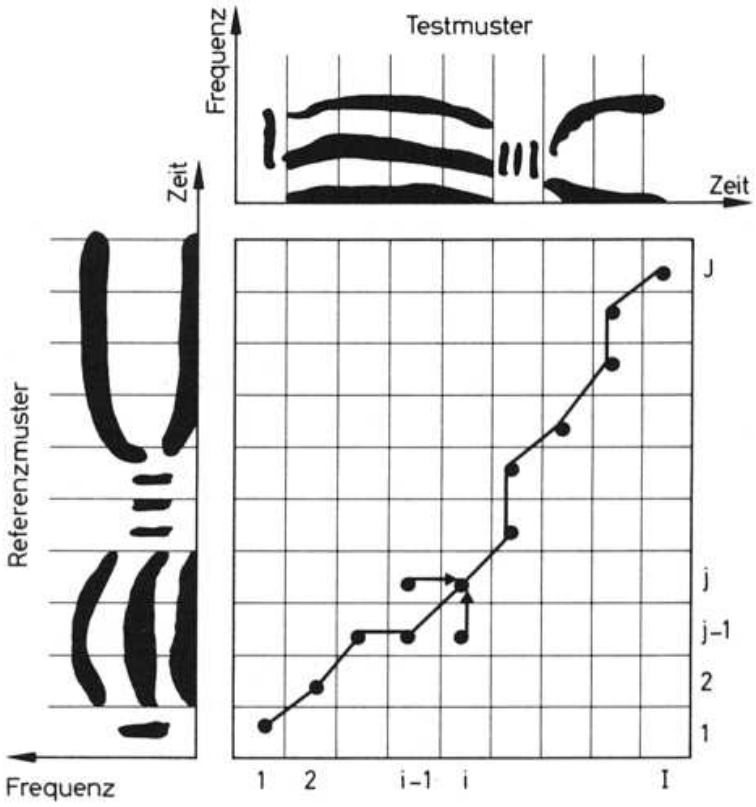


Spracherkennungsmethoden (1)

Nichtlineare Anpassung (Dynamic Time Warping)

Kernproblem: zeitlich unterschiedlicher Aufbau von verschiedenen Wörtern unterschiedlicher Sprachsituationen

Einsatz von Methoden der dynamischen Programmierung zur Abbildung zweier Wörter aufeinander.



Punkte im Schema identifizieren als gleich erkannte Muster.

Spracherkennungsmethoden (2)

Hidden-Markov-Modelle

Der Prozess der Spracherzeugung wird durch einen stochastischen endlichen Automaten modelliert. Spracherkennung wird dadurch realisiert, dass der Automat eine Wahrscheinlichkeit berechnet dafür, dass er selbst ein vorliegendes Sprachsignal erzeugt hat. D.h., für jedes Wort wird ein Automat in einer Lernphase erzeugt und trainiert und kann dann in der Erkennungsphase die gewünschte Wahrscheinlichkeit berechnen.

Literatur

B. Eppinger, E. Herter „Sprachverarbeitung“ Carl Hauser Verlag München, Wien 1993

J.R. Deller, J.G. Proakis, J.G.H. Hansen „Discrete-Time Processing of Speech Signals“ Prentice Hall 1987

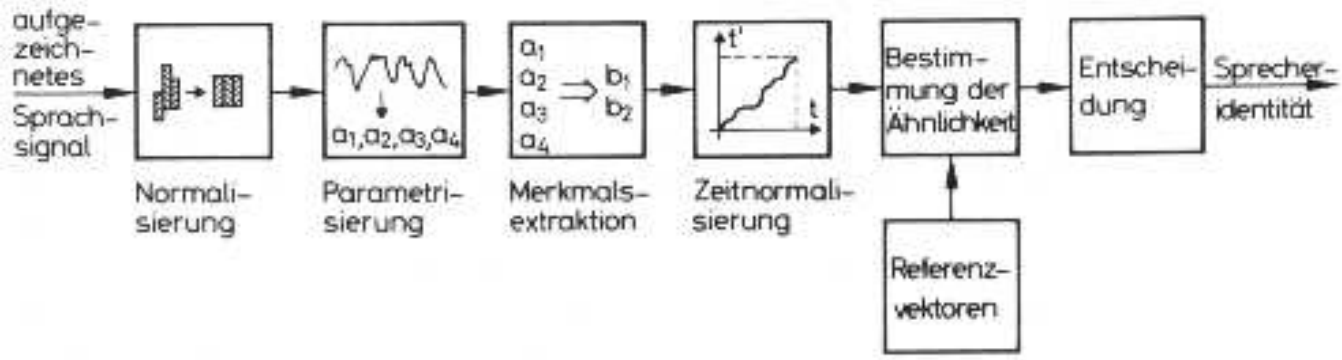
Sprechererkennung

Anwendungen

- Sprecher-Verifikation, z.B. für polizeiliche Ermittlungen; Problem: nicht-kooperativer Sprecher (Stimme verstellen, keine vorgegebenen Sätze)
- Sprecheridentifikation z.B. für die Zugangskontrolle

Methoden

- Mustererkennung



Sprecherindividuelle Merkmale

Die sprachlichen Unterschiede zwischen Menschen beruhen auf unterschiedlichen Dimensionen des Vokaltraktes und der Stimmbänder, sowie auf einem entwickelten Sprachverhalten. Letzeres kann kopiert werden, so dass Ersteres wichtiger ist für die Sprechererkennung.

- Berechnung von Langzeit-Mittelwerten und –Standardabweichungen von Merkmalen
- Berechnung von Histogrammen aus Merkmalen

4.4.2 Erkennung von Stille

Ziel

Erkennung von Phasen relativer Stille in der Audio-Spur

In natürlichen Schallsituationen gibt es Momente, die der Mensch als „Stille“ identifiziert. Sie sind dadurch charakterisiert, dass ein dominantes Vordergrundgeräusch (z.B. Sprache) fehlt und nur noch Hintergrundgeräusche existieren.

Ansätze zu ihrer Bestimmung

1. Auf der Basis von Lautheitsmessungen.

Leisere Teile werden als relative Stille erkannt. Problem: Was sollte der Schwellenwert sein?

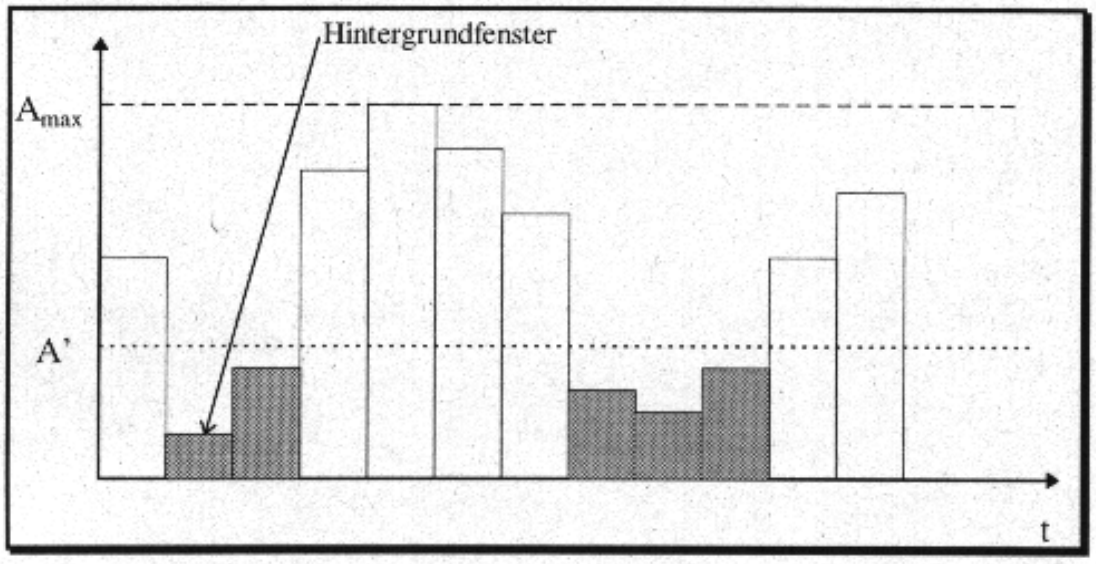
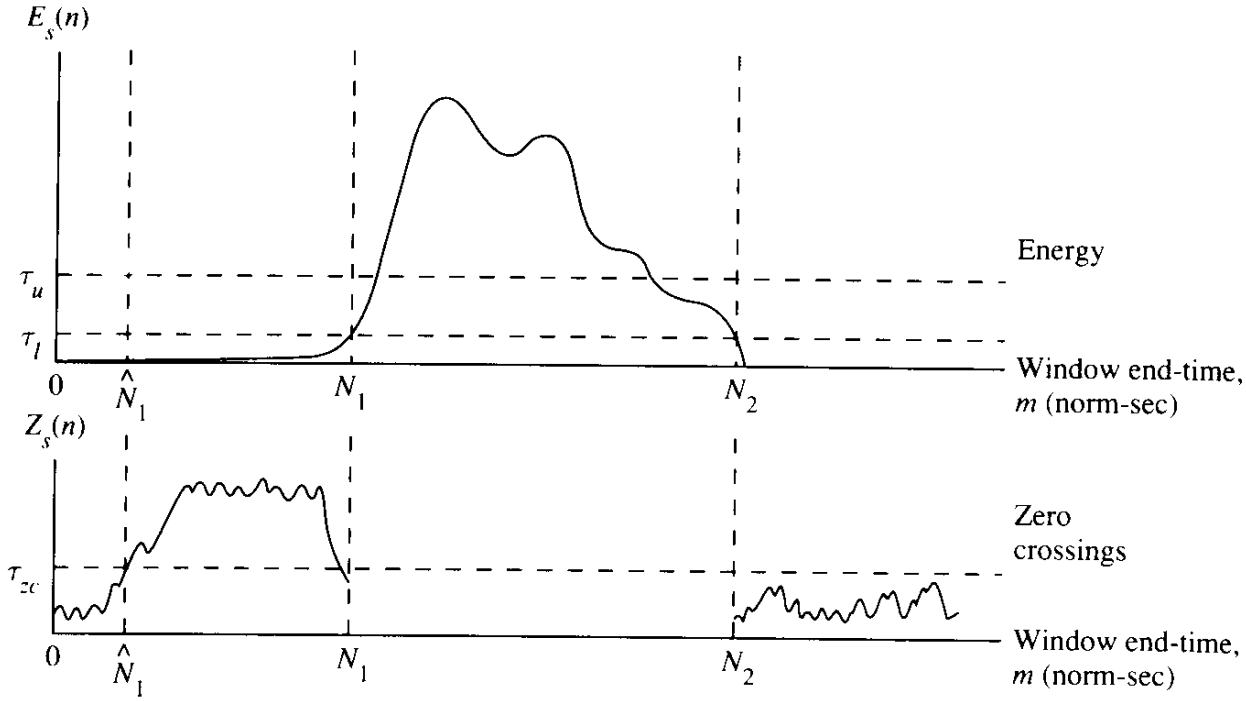


Abb. 12: Ermittlung der Hintergrundfenster mittels Pegelanalyse.

Erkennung von Stille (2)

2. Auf der Basis von Energiemessungen

Wird in Kombination mit dem Nulldurchgangsmaß zur Identifikation von Wortgrenzen eingesetzt



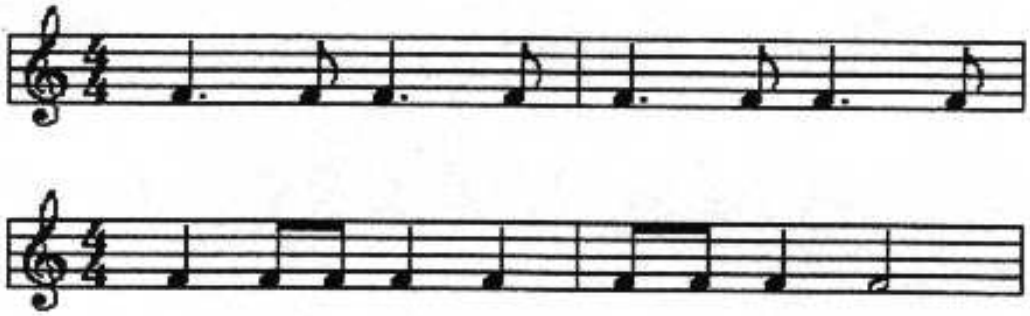
Zeitliche Strukturen

Takt, Rhythmus

Takt = regelmäßig wiederkehrende Folge von betonten und unbetonten Schlägen ("beats"); z.B. ¾ -Takt

Rhythmus = einprägsame, regelmäßig wiederkehrende Folge von Notenwerten; z.B.

Typischer Rhythmus einer Baßdrum zu einem 8-Beat-Rhythmus



Beispiel für einen Rhythmus, der über mehrere Takte geht.

Erkennung von Schlagzeug-Schlägen

- Im Zeitbereich: durch Amplitudenstatistiken
  - Erkennen von Amplitudenspitzen über Schwellenwerte, wobei relative oder absolute Maximalwerte gewählt werden können.
  - Ergebnisse: für „richtige“ Musikstücke unbrauchbar, da die Amplitudenspitzen nicht ausgeprägt genug sind
- Im Frequenzbereich: durch Schlagzeugerkennung

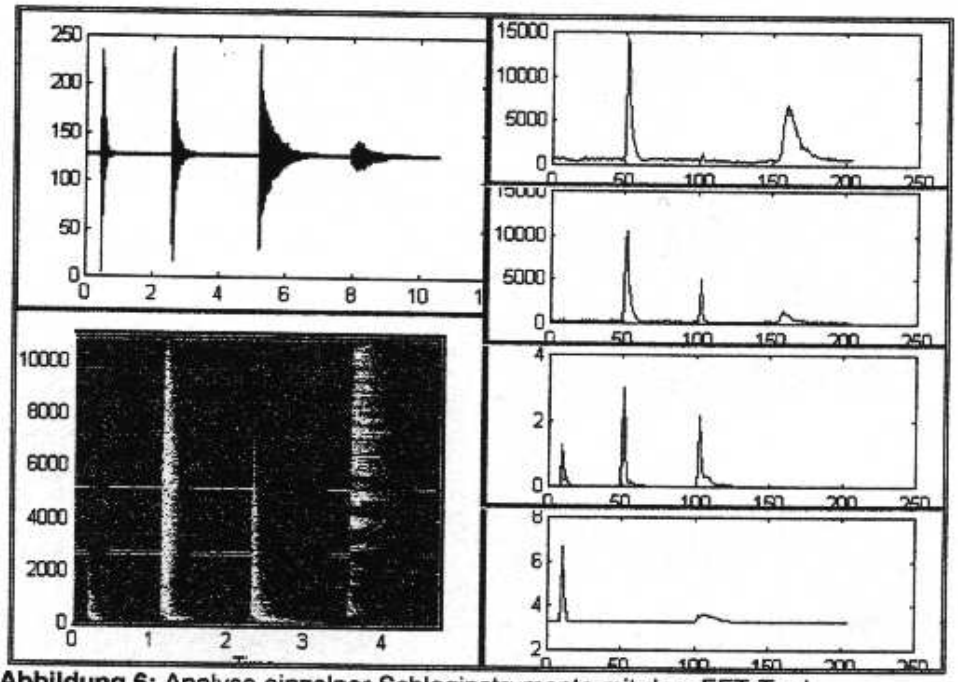


Abbildung 6: Analyse einzelner Schlaginstrumente mit dem FFT-Tool (Baßdrum, Snare, Tom und Becken)

Ergebnisse zur Rhythmus-Erkennung

Identifikation bei „richtigen“ Musikstücken zuverlässig: in 15 von 20 Stücken wurde der Rhythmus korrekt erkannt

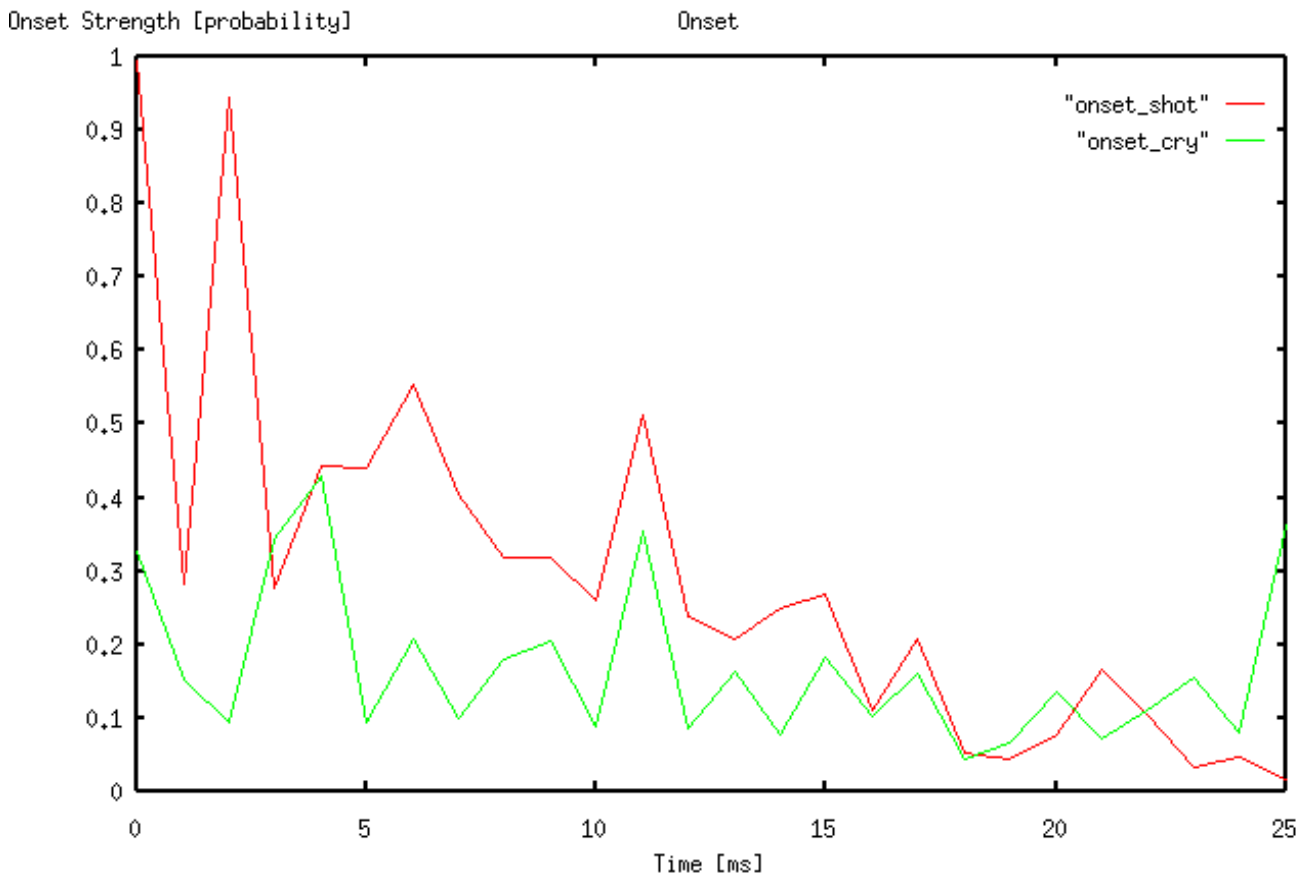
Identifikation von Musikstücken

Wichtige Variable

- Spektrale Energieverteilung
  - Frequenzbandbreite des Instruments
  - Spezifische Obertöne und ihre Ausprägung
- Zeitliche Struktur der Frequenzkomponenten
  - beim Anspielen
  - beim Halten
  - beim Ausklingen
  - z.B. bei Blasinstrumenten bleiben Frequenzkomponenten über die Zeit konstant, während sie sich bei Streichern ständig verändern (Vibrato)
- Anklingzeit des Tons
  - z.B. Trompete, Horn: kurze Anklingzeit
  - Klarinette, Saxophon: lange Anklingzeit
- Übergänge zwischen Tönen
- Abklingfrequenzen der Töne:
  - Eigenfrequenz des Instruments tritt hervor

Einsetzen und Abklingen des Tons

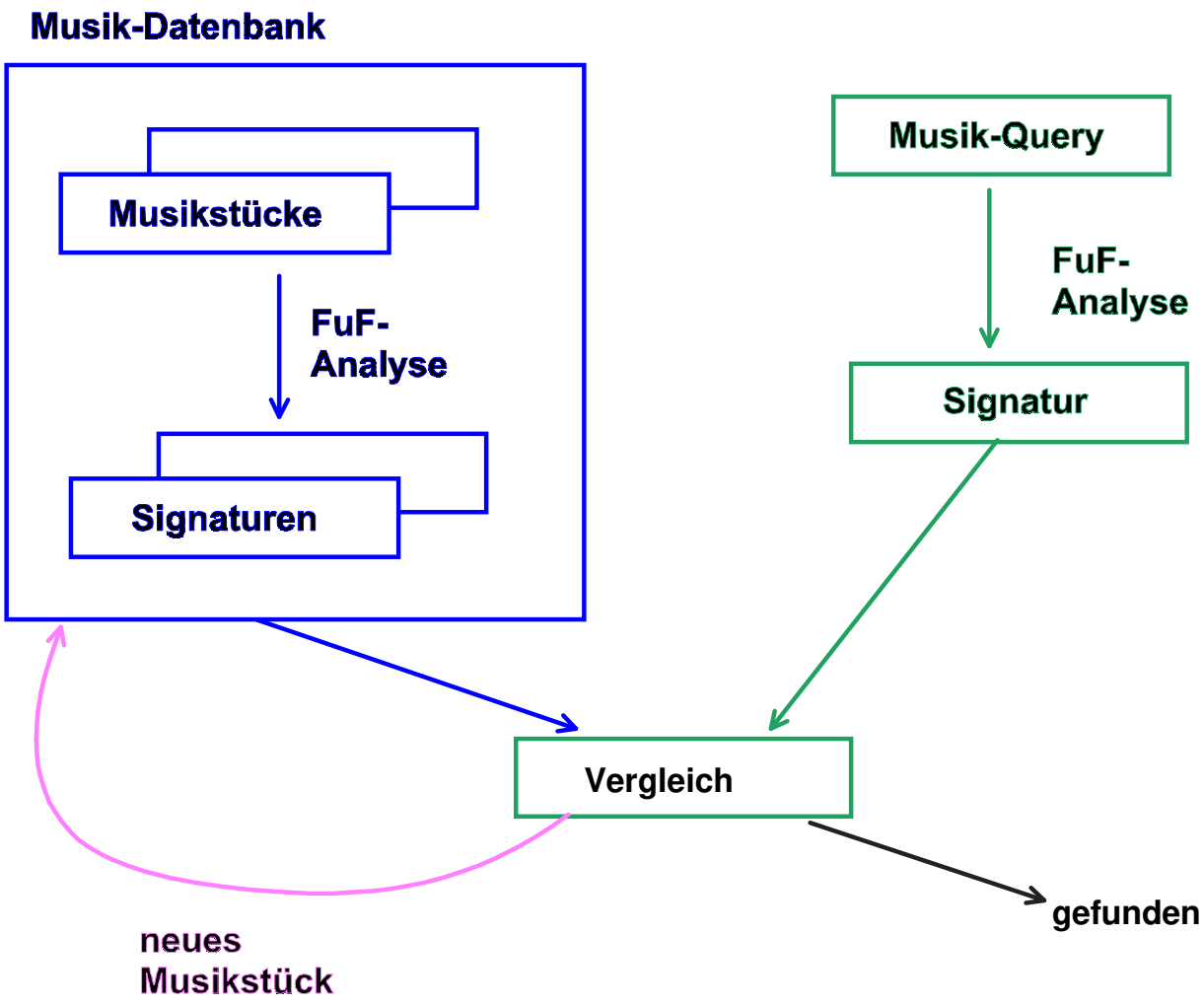
Beispiel: Schuss und Schrei



4.4.3 Anwendungsbeispiele (Audio)

- Unterscheidung von Schweigen, Sprache, Musik und Geräusch auf der Tonspur eines Videos
- Transkription von Sprache nach ASCII zur nachfolgenden automatischen Indexierung (heute noch nicht ganz machbar)
- Direkte Inhaltserkennung in sehr typischen Fällen: Tennis, Schüsse, Explosionen, Tierlaute (Bellen, Wiehern)
- “Query-by-Example“ für Musikstücke

Suche in Musik-Datenbanken



4.5 Anwendungsbeispiele

4.5.1 Genre-Erkennung

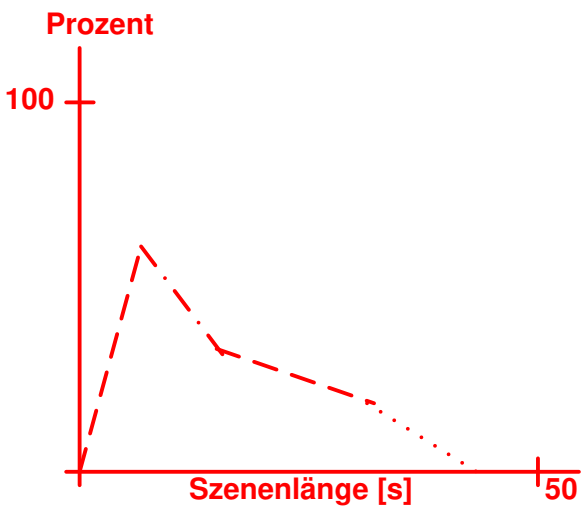
Ziel

Zuordnung eines Videos zu einem Genre (Spielfilm, Nachrichten, Werbung, Musik-Clip etc.)

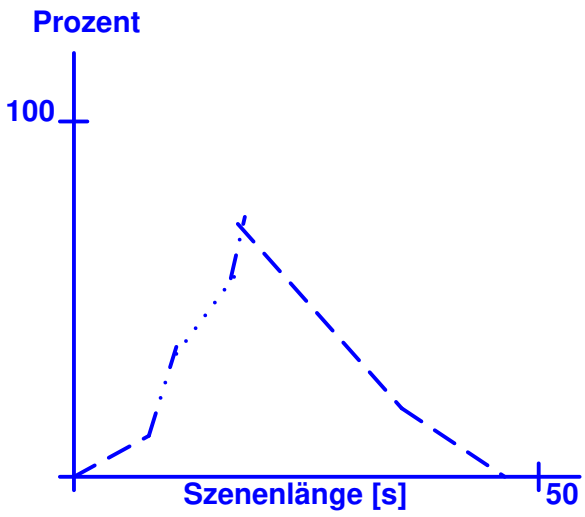
Technik

Kombination von vielen Parametern der Ebene 2 zu einem charakteristischen “Fingerabdruck“

Beispiel: Szenenlängen-Verteilung



Musikclip



Nachrichtensendung

Fazit

Allein auf Grund der Szenenlängen-Verteilung kann ein Musik-Clip von einer Nachrichtensendung unterschieden werden!

Experimentelle Ergebnisse

Ermittelt am Lehrstuhl für Praktische Informatik IV.

- 140 Clips aus 7 Genres: Nachrichten, Fußball, Tennis, Talkshow, Musik-Clip, Zeichentrickfilm, Werbung
- Der MoCA-Prototyp klassifizierte zwischen 87% (Werbe-Clips) und 99% (Nachrichten) richtig.

Problem:
Die Berechnungszeit für ein Video von 3 min beträgt 28 h auf einer SUN SPARC20!

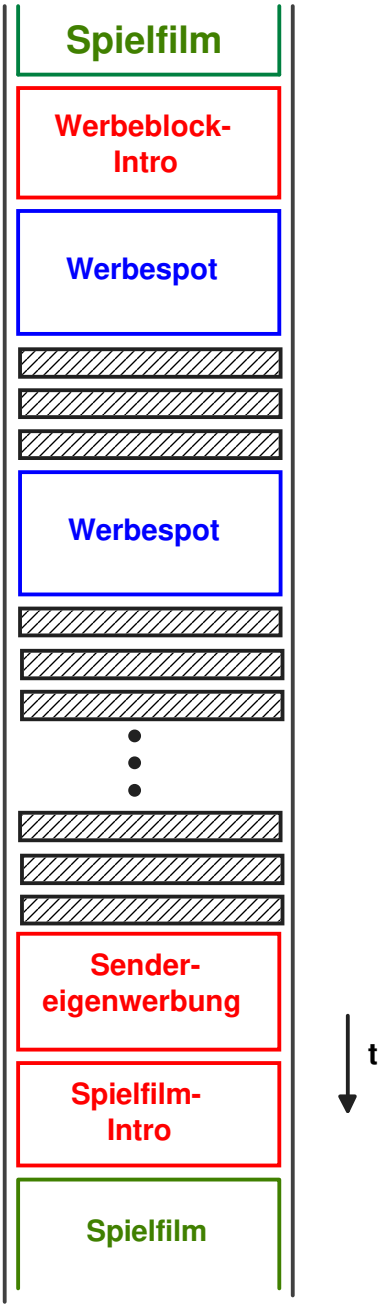
4.5.2 Erkennung von Werbe-Spots in einem Videostrom

Motivation

- Überprüfen, ob Werbespots tatsächlich gesendet werden, wie vertraglich vereinbart
- Automatisch beobachten, was die Konkurrenz tut
- Korrelieren der messbaren Eigenschaften der verschiedenen Werbespots zu ihrem Erfolg auf dem Markt (z.B. Farbstimmung, Bewegungsintensität)
- Entfernen unerwünschter Werbespots aus einem Video-Datenstrom

Eigenschaften von TV Werbespots

Struktur eines Werbeblocks



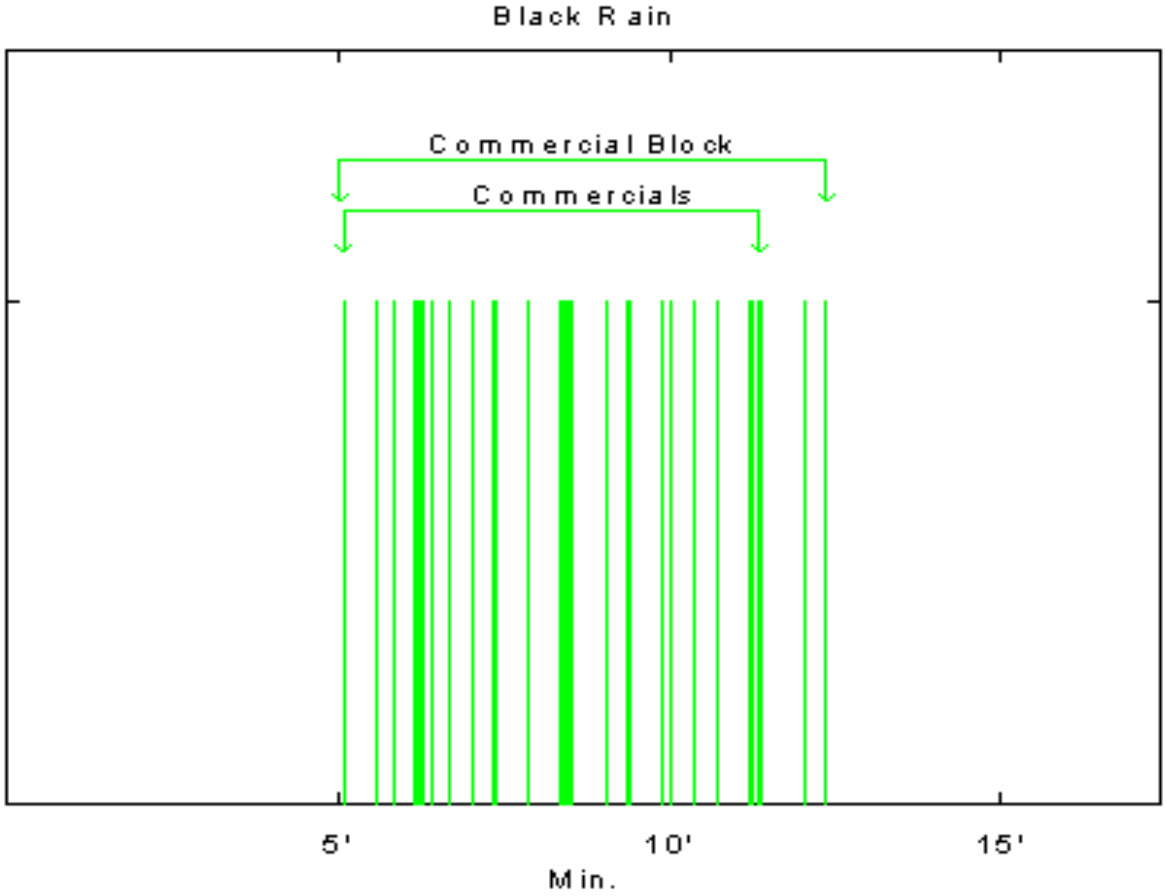
Inhaltsbasierte Lokalisierung von Werbespots

Erkennung auf der Basis von

- dunklen, einfarbigen Frames
- hoher Bewegungsintensität (hohe ECR, viele lange Bewegungsvektoren)
- Häufigkeit von harten Schnitten.

Schwarze Frames

Auftreten von einfarbigen, dunklen Frames in einem Spielfilm, der von einem Werbeblock unterbrochen wird:



Lokalisierung in zwei Schritten

1. Schritt: schnelle Vorauswahl

Monochrome Bilder und häufige harte Schnitte dienen als Vorauswahl und bestimmen die Zeitbereiche des Videos, die als Kandidaten angesehen werden.

2. Schritt: Präzise Grenzen bestimmen

Beginn und Ende jedes Werbeblocks werden durch Bewegungsvektoren und Kantenveränderungsmaß (ECR) genauer bestimmt.

Experimentelle Ergebnisse

Getestetes Material

Vier Stunden deutsches Fernsehen auf Video (nur Spielfilme mit eingebetteten Werbeblöcken)

Ergebnis

Alle Werbeblöcke wurden korrekt erkannt.

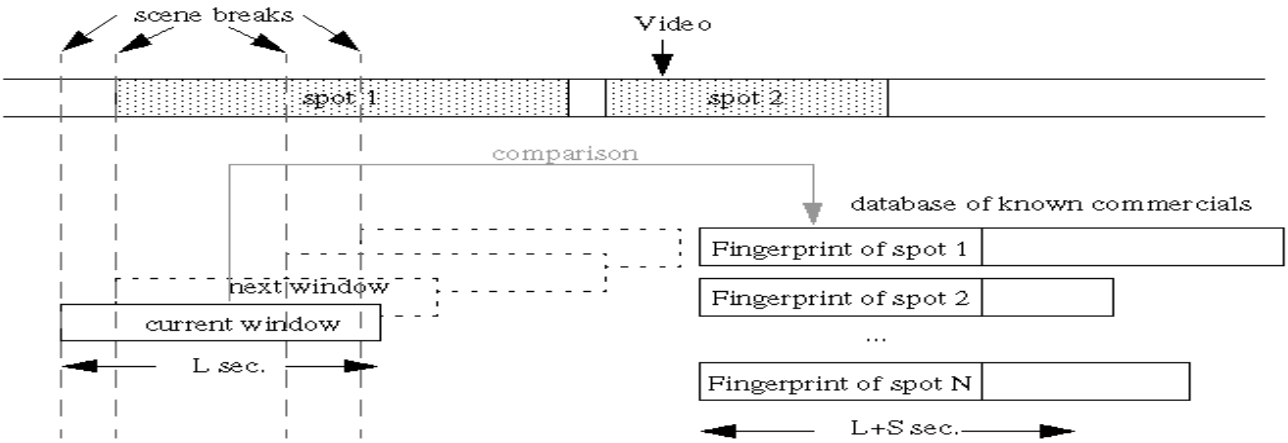
Wiedererkennung bekannter Werbespots

Vorgehensweise

- Berechnung eines “Fingerabdrucks“ für jeden Werbespot
- Aufbau einer Datenbank der Fingerabdrücke bekannter Werbespots
- Vergleich des Werbespots im Videostrom mit den Werbespots in der Datenbank

Vergleich

- Schnelles Vorspringen bis zum nächsten harten Schnitt
- Ermittlung des “Fingerabdrucks“ auf eine Länge von *L* Frames
- Vergleich mit der Datenbank



Experimentelle Ergebnisse

Getestet an 200 zusätzlichen Werbespots

Ergebnisse

- alle bekannten Werbespots wurden erkannt
- es erfolgte keine falsche Erkennung
- der durchschnittliche Unterschied zwischen den genauen und den berechneten Grenzen der Spots betrug fünf Bilder

4.5.3 Eine intelligente Alarmanlage

Experiment

- Installation einer Kamera zur Raumüberwachung
- Ständige Ermittlung der Bewegungsvektoren aus dem Kamerabild
- Beim Erkennung von Bewegung Starten des digitalen Video-Rekorders, Digitalisierung, Kompression, Aufzeichnung des Videos auf der Festplatte
- Programmierbar! Beispiel: Leute in weißen Kitteln lösen keinen Alarm aus

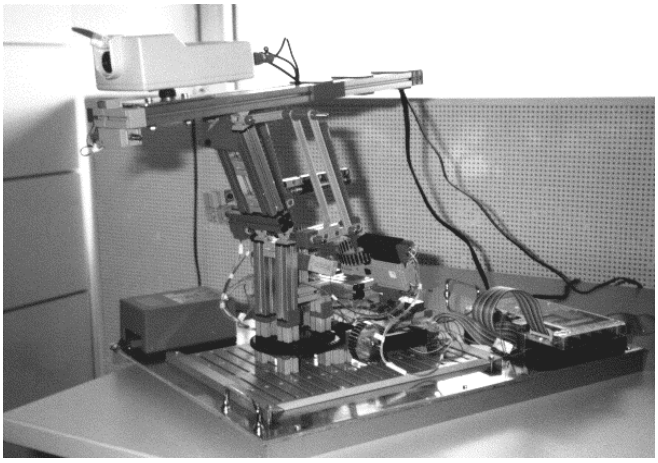
Anwendungsbeispiele

PC-Pool, Intensivstation im Krankenhaus, Bankräume, Baustellen u.v.m.

4.5.4 Automatische Verfolgung von bewegten Objekten

Experiment

Aufbau des Kamera-Roboters CaRo:



Segmentierung von Objekten aus dem Blickfeld der Kamera durch Kantenerkennung und einfache Farbmuster. Im Experiment: ein ferngesteuertes Auto auf grauem Fußboden

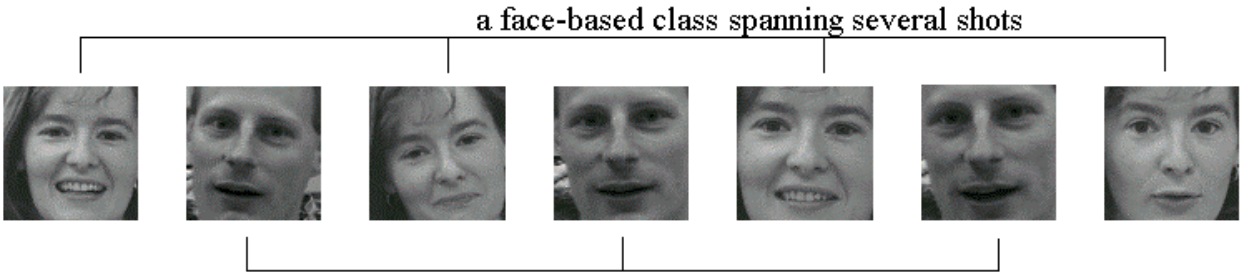
Anwendungsmöglichkeiten

- Sprecherverfolgung auf dem Podium
- Kameraführung bei Sportereignissen
- Video-Überwachung

4.5.5 Erkennung von Dialogen

Algorithmus

- Gesichts-Lokalisierung im Video mit dem neuronalen Netz
- Ermittlung eines “Fingerabdrucks“ von jedem erkannten Gesicht
- Wenn ein Muster von Gesichtern in der Form a,b,a,b,... vorkommt, wird ein Dialog erkannt



Anmerkung: Dialoge in dieser Form sind in Spielfilmen sehr häufig zu finden. Man bezeichnet diese Kameraführung als “Schuss und Gegenschuss“.

4.5.6 Video-Abstracting

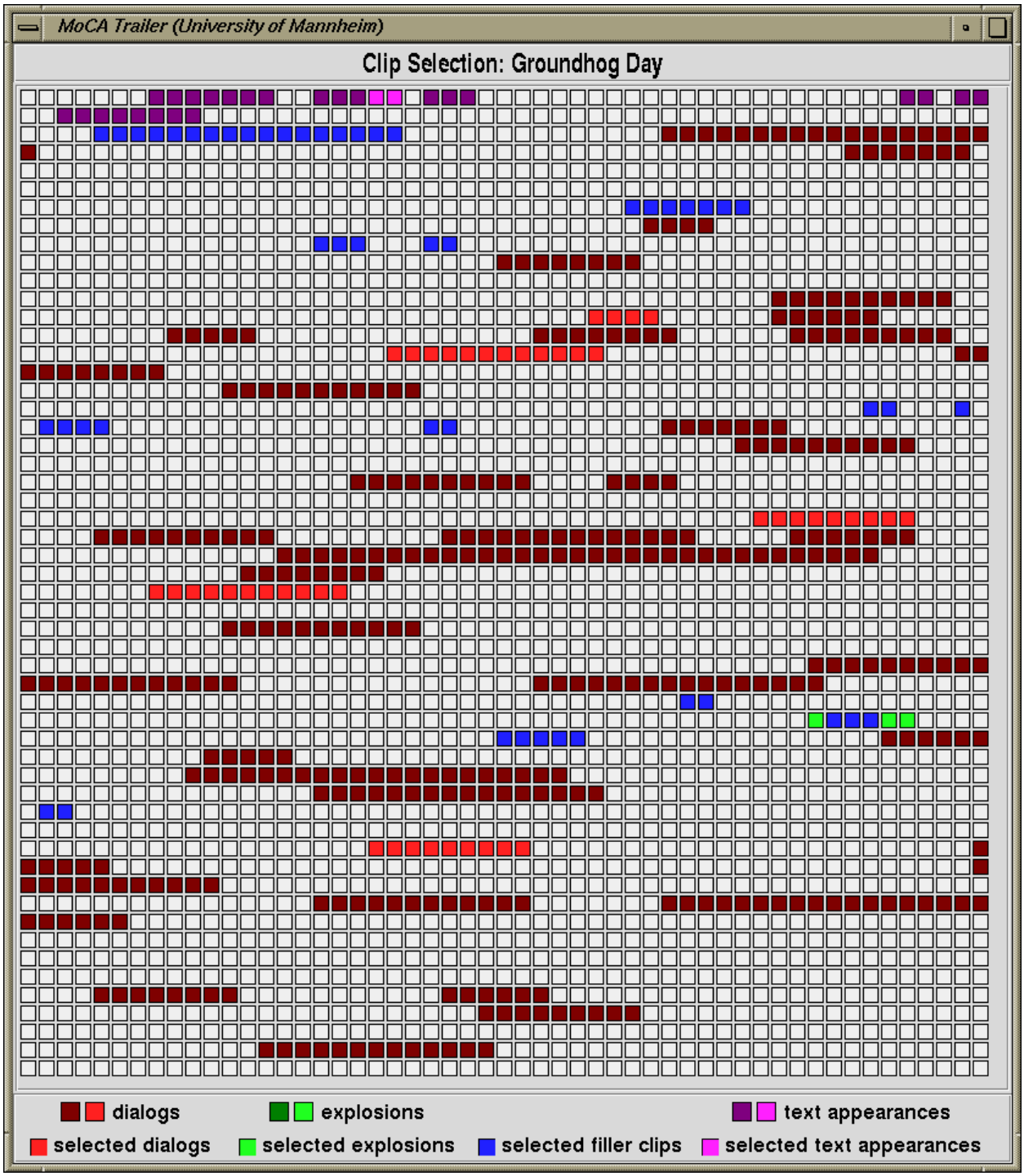
Ziel

Automatische Erstellung einer Video-Kurzfassung (Trailer) aus einem vollen Video

Vorgehensweise

- Zerlegung des Videos per Schnitterkennung
- Charakterisierung der Einstellungen mit den vorhandenen Tools, so weit wie möglich
- Definition einer Heuristik, was in das Abstract hinein soll und in welcher Reihenfolge
- Zusammenfügen des Abstracts aus den einzelnen Einstellungen

Beispiel für ein Video-Abstract



4.5.7 Automatische Erkennung von Gewalt oder Pornographie?

Mit heutigen Mitteln ist die automatische Erkennung von Gewalt in Videos nicht möglich. Die Semantik von Gewalt ist zu tief und zu kompliziert.

Dasselbe gilt für pornographische Darstellungen.