
MULTIMEDIATECHNIK

Zusammenfassung

WS 2020/21
UNI TÜBINGEN

Inhalt

Grundlagen	4
Definition Medium	4
Teilaspekte der Multimediatechnik.....	4
Medienarten und konkrete Praxisbeispiele	4
Wahrnehmung.....	4
Farbmischung	4
Farbuntertastung.....	4
Subsampling Verfahren	5
Textkomprimierung	6
Informationsgehalt.....	6
Mittlerer Informationsgehalt (Entropie)	6
Redundanz.....	6
Kompressionsfaktor	6
Laufängerkodierung (RLE).....	6
Huffman-Kodierung.....	7
Lempel-Ziv-Welch-Kodierung (LZW).....	8
Burrows-Wheeler-Transformation.....	9
Move-to-Front Kodierung	10
Signalverarbeitung	11
Analoges vs. Digitales Signal.....	11
Fehler bei Digitalisierung.....	11
Fouriertheorem	11
Abtasttheorem (Nyquist-Grenze).....	11
Bildanalyse.....	12
Anwendungsbereiche.....	12
Operationstypen.....	12
Histogramm	12
Dynamikumfang	12
Abhängigkeit lokaler von globalem Kontrast	12
Globaler Kontrast	12
Lokaler Kontrast	12
Bildbearbeitung	12
Boxcarfilter	12
Min-Filter	13
Max-Filter	13

Median-Filter	13
Gauß-Filter	13
Laplace-Filter	13
Prewitt Filter	14
Kirsch-Operator	14
Sobel-Operator	14
Roberts-Operator	14
Canny-Verfahren	14
Laplacian of Gaussian (LoG)	14
Bildkomprimierung	15
Ortsauflösung (Abtastung)	15
Kontrastauflösung (Quantisierung)	15
JPEG Kompression	16
Diskrete Cosinus Transformation (DCT)	16
Zick-Zack Abtastung	16
Videokompression	17
Verfahrenskategorien	17
Kompressionsverfahren	17
Makroblock	17
Frametypen	17
Group of Pictures (GOP)	17
Folgen von Übertragungsfehlern	17
Audiokompression	18
Maskierungsschwelle	18
Psychoakustisches Modell	18
Modifizierte Diskrete Cosinus Transformation (MDCT)	18
MP3-Kompression	18
Variable Bitrate	18
Videoanalyse	19
Schnitterkennung	19
Edge Change Ratio (ECR)	19
Objektbewegung vs. Kamerabewegung	20
Optischer Fluss	20
Genre-Erkennung	20
Dynamic Time Warping	21
Dynamic Time Warping	21

Levenshtein-Distanz	21
Maximale und minimale Levenshtein-Distanz	21
Gestenerkennung.....	21
Kategorien	21
Tiefendatengenerierung.....	22
Stereo-Vision	22
Kinect.....	22

Grundlagen

Definition Medium

Lat. „Das in der Mitte Befindliche“. Gesellschaftliche Definition: Kommunikationsmittel, in der Technik: Datenform.

Teilaspekte der Multimediatechnik

- Multimedialität
- Multicodalität
- Multimodalität
- Interaktivität

Medienarten und konkrete Praxisbeispiele

- **Perzeptionsmedium** ist für die Sinneswahrnehmung zuständig, also Augen (Sehen), Ohren (Hören), Nase (Riechen/Schmecken), Mund (Schmecken), Haut (Fühlen).
- **Repräsentationsmedium** ist für die Darstellung von Informationen, z.B. einer Bild-datei (TIF, JPEG, PNG, ...), oder eines Textes (ASCII, LaTeX, HTML, ...).
- **Präsentationsmedium** ist für die Ein- und Ausgabe zuständig, z.B. Maus, Tastatur und Kamera (Eingabe), oder Bildschirm und Lautsprecher (Ausgabe).
- **Speicherungsmedium** ist zum Festhalten von Informationen gedacht, z.B. Festplatten oder DVDs.
- **Übertragungsmedium** ist für die Übertragung von Informationen zuständig. Dabei kann man differenzieren zwischen der physischen Form, z.B. Glasfaser oder WLAN, und der Codierung, z.B. Kompression oder Übertragungscodierung.

Wahrnehmung

Farbmischung

- Additiv: durch Hinzufügen von Licht wird Farbe erzeugt (kein Licht = schwarz)
- Subtraktiv: Pigmente erzeugen Farbe (keine Pigmente = schwarz)

Farbuntertastung

TYPISCHES KLAUSURTHEMA

Farbuntertastung bedient sich an der Schwäche des menschlichen Auges, Farben nur mit reduzierter Auflösung wahrnehmen zu können, im Gegensatz zu Helligkeitswerten.

Farbuntertastung ist eine Technik, um Daten zu komprimieren. Dafür werden die Farb- und die Helligkeitsinformationen separat komprimiert. Die Farbinformationen werden dabei auf mehrere Pixel verteilt, man erhält also Blöcke von z.B. 4x4 Pixeln, die die gleiche Farbe haben. Für den Menschen ändert sich aber nichts am Bild, da er den Unterschied in der Auflösung nicht bemerken kann.

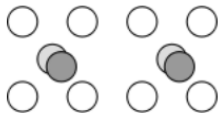
Subsampling Verfahren



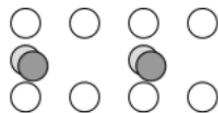
Y'CbCr 4:4:4



Y'CbCr 4:2:2



Y'CbCr 4:2:0



Y'CbCr 4:2:0 MPEG-2-Abtastpositionen



4:1:1



4:2:0



4:2:2



4:4:4

Textkomprimierung

Informationsgehalt

Definition: Der Informationsgehalt eines Zeichens mit der Wahrscheinlichkeit p einer gedächtnislosen Binärquelle lautet:

$$I(p) = \log_2(1/p) \text{ bit} = -\log_2(p) \text{ bit}$$

Mittlerer Informationsgehalt (Entropie)

Definition: Eine diskrete gedächtnislose Quelle X mit dem Zeichenvorrat $X = \{x_1, x_2, \dots, x_N\}$ und den zugehörigen Wahrscheinlichkeiten p_1, p_2, \dots, p_N besitzt den **mittleren Informationsgehalt**, die **Entropie** [Shannon, 1948]

$$H(X) = - \sum_{i=1}^N p_i \cdot \log_2(p_i) \text{ bit}$$

Redundanz

- **Definition:** Die Differenz zwischen Entropie einer Quelle und dem Entscheidungsgehalt ihres Zeichenvorrats nennt man **Redundanz**.

$$R = H_0 - H(X)$$

Kompressionsfaktor

Komprimiert

Original

Lauf­längen­kodierung (RLE)

Die Lauflängenkodierung funktioniert wie folgt:

Wenn in einem zu kodierenden Inhalt mehrere gleiche Zeichen hintereinander vorkommen, dann können diese zusammengefasst werden. Man kürzt dies dann ab mit der Anzahl der Vorkommnisse und dem Zeichen, das vorkam.

Man kann zusätzlich einen Marker einführen und eine Untergrenze, ab wie vielen Wiederholungen erst abgekürzt werden soll, damit es sich lohnt.

Hier ein kleines Beispiel:

- Der zu kodierende Inhalt:
wwwwwwwwwwwwwwBwwwwwwwwwwwwwwBBBwwwwwwwwwwwwwwwwwwwwwwwwwwww
wwwwwBwwwwwwwwww
- Es kommen hier: 12 weiße, 1 schwarzer, 12 weiße, 3 schwarze, 25 weiße, 1 schwarzer, 14 weiße Pixel vor
- Also ist die Kodierung: 12w1B12w3B55w1B14w

Hier wurden dann statt 68 Zeichen nur 18 verwendet.

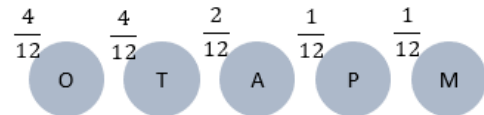
Huffman-Kodierung

(POTATOTOMATO)

1 – Auftreten der Zeichen mit Wahrscheinlichkeiten in Tabelle darstellen

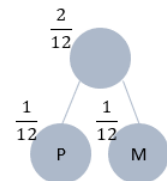
P	O	T	A	M
1/12	4/12	4/12	2/12	1/12

2 – Zeichen der Größe nach sortieren (optional)

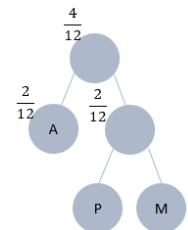


3 - Die zwei Knoten mit den kleinsten Wahrscheinlichkeiten zusammenfassen.

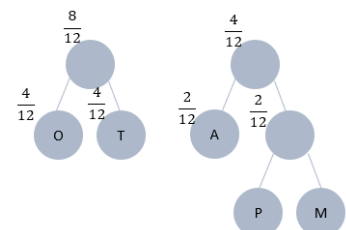
(also hier P und M) Und die Wahrscheinlichkeiten addieren.



4 – Die zwei Knoten/Bäume mit den kleinsten Wahrscheinlichkeiten zusammenfassen und die Wahrscheinlichkeiten addieren. (Hier A und der Baum aus P und M)

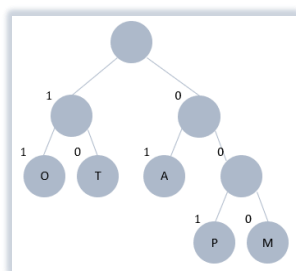
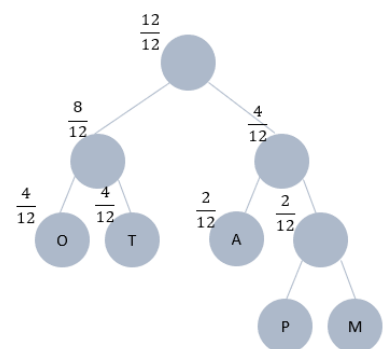


5 - Die zwei Knoten/Bäume mit den kleinsten Wahrscheinlichkeiten zusammenfassen und die Wahrscheinlichkeiten addieren. (Hier O und T)



6 - Die zwei Bäume mit den kleinsten Wahrscheinlichkeiten zusammenfassen und die Wahrscheinlichkeiten addieren. (Hier der Baum aus O und T und der Baum aus A, P und M.)

Dies führt man eben solange durch, bis die Wahrscheinlichkeit beim obersten Knoten bei 1 liegt. Bzw bis alle Knoten zu einem Baum zusammengefasst wurden.



Code-Erstellung: Man verteilt nun an die Knoten Einsen und Nullen, sodass man dann entlang des jeweiligen Pfades die Codierung ablesen kann.

P	O	T	A	M
001	11	10	01	000

Limpel-Ziv-Welch-Kodierung (LZW)

(POTATOTOMATO)

Hier geht man wie folgt vor:

- 1) Notiere dir alle Buchstaben in der linken Spalte (mit Zeichenziffer, jedoch in der Klausur schon gegeben). Man kann hier das Wort ablesen.
- 2) Wenn eine Buchstaben-Kombination (wie in Zeile 2) noch nicht so vorgekommen ist, dann wird diese ins Wörterbuch eingetragen und bekommt eine Zahl zugeordnet. Da hier das Alphabet die Plätze 0-25 belegt, beginnen die neuen Einträge bei 26.
- 3) Die Ausgabe ist immer um eine Zeile verrutscht. So wird erst in Zeile 2 der Buchstabe P aus Zeile 1 ausgegeben. Man kann hier das Wort am Ende ablesen.
- 4) Bei Puffer steht immer der Buchstabe der ganz links in der ersten Spalte steht. (außer Sonderfall)
- 5) Wenn man nun zu dem Sonderfall kommt, dass eine Buchstabenkombination so schon im Wörterbuch steht, geht man wie folgt vor:
 1. Man lässt den Wörterbucheintrag dieser Zeile frei und trägt in die nächste dann die Dreierkombination. Wie beispielsweise in Zeile 7 und 8. (hier eben „OTO“(31))
 2. Auch lässt man die Ausgabe dieser Zeile frei und trägt in die nächste die Kombination, die es schon gegeben hatte. (Hier eben 27(OT))
 3. In den aktuellen Puffer trägt man in dieser Zeile (also der leeren) die Kombination, die es schon gegeben hatte (hier OT) und fährt dann normal fort.
- 6) Wenn das Wort durch ist, muss man den Puffer leeren. Man trägt also ganz links ein „EOF“ ein und gibt in die Ausgabe den letzten Buchstaben des Wortes (siehe linke Spalte). Der Puffer ist dann leer.

Tipp:

- Überprüfe, ob in der Ausgabe alle Buchstaben ausgegeben wurden.
- Überprüfe, ob die Wörterbucheinträge aufsteigend nummeriert sind.

		Neuer Wörterbucheintrag	Ausgabe	Aktueller Puffer
1	P(15)			P
2	O(14)	„PO“ (26)	15(P)	O
3	T(19)	„OT“ (27)	14(O)	T
4	A(0)	„TA“ (28)	19(T)	A
5	T(19)	„AT“ (29)	0(A)	T
6	O(14)	„TO“ (30)	19(T)	O
7	T(19)			OT
8	O(14)	„OTO“ (31)	27(OT)	O
9	M(12)	„OM“ (32)	14(O)	M
10	A(0)	„MA“ (33)	12(M)	A
11	T(19)			AT
12	O(14)	„ATO“ (34)	29(AT)	O
13	EOF		14(O)	

Burrows-Wheeler-Transformation

(TOLKIN)

Transformation:

Zuerst werden alle Rotationen erzeugt und in eine Tabelle geschrieben (hier wurde das Wort immer um einen Buchstaben nach vorne verschoben!)

	1	2	3	4	5	6
1	T	O	L	K	I	N
2	O	L	K	I	N	T
3	L	K	I	N	T	O
4	K	I	N	T	O	L
5	I	N	T	O	L	K
6	N	T	O	L	K	I

Dann wird diese Tabelle alphabetisch sortiert:

	1	2	3	4	5	6
1	I	N	T	O	L	K
2	K	I	N	T	O	L
3	L	K	I	N	T	O
4	N	T	O	L	K	I
5	O	L	K	I	N	T
6	T	O	L	K	I	N

Der Ausgabebetext ist „KLOITN“ (letzte Spalte) und der Index ist 6 (Zeile in der das richtige Wort steht).

Rücktransformation:

	1	2	3	4	5	6
Codierter Text (1)	K	L	O	I	T	N
Position (2)	1	2	3	4	5	6
Sortierter Text	I	K	L	N	O	T
Sortierte Pos. (4)	4	1	2	6	3	5

Achtung: Beim Sortieren werden die Großbuchstaben zuerst sortiert und vorneangestellt.

Nun geht man wie folgt vor:

- 1) Man beginnt beim letzten Index der sortierten Position (4). Der Buchstabe ist **T**. Hier steht eine 5. Dann schaut man in der Spalte 5, welcher Buchstabe dort steht. Dies ist das **O**.
- 2) Nun sucht man die sortierte Position (4) von **O**. Hier steht eine 3. Dann schaut man in der Spalte 3, welcher Buchstaben dort steht. Dies ist ein **L**.

- 3) Nun sucht man die sortierte Position (4) von **L**. Hier steht eine 2. Dann schaut man in der Spalte 2, welcher Buchstaben dort steht. Dies ist ein **K**.
- 4) Nun sucht man die sortierte Position (4) von **K**. Hier steht eine 1. Dann schaut man in der Spalte 1, welcher Buchstaben dort steht. Dies ist ein **I**.
- 5) Nun sucht man die sortierte Position (4) von **I**. Hier steht eine 4. Dann schaut man in der Spalte 4, welcher Buchstaben dort steht. Dies ist ein **N**.

Das Wort ist also: TOLKIN

Anschließend „Move-to-Front“-Kodierung. Vorteil von BTW mit MTF: Sortiert Zeichenketten, sodass diese sich einfach mit Huffman oder RLE komprimieren lassen.

Move-to-Front Kodierung

(ahhMMT)

- 1) Alphabet aufstellen: MTah (Großbuchstaben zuerst, dann in alphabetischer Reihenfolge)
- 2) Für jedes Zeichen *z* aus dem Originalwort, folgende Aktionen durchführen:
 - a. Die Position von *z* im Alphabet herausuchen (Index notieren, ab 0)
 - b. Entferne *z* dann aus dem Alphabet und für es vorne an
 - c. Fahre mit diesem Verfahren fort, suche jedoch dann immer im angepassten Alphabet nach dem Index (notiere dieses dann auch in der jeweiligen Iteration)
- 3) Die Ziffern jeder Iteration stellen die Ausgabe dar

Beispieldurchführung:

Zu codierendes Wort: ahhMMT und Alphabet: MTah

- 1) 2, aMTh
- 2) 3, haMT
- 3) 0, haMT
- 4) 3, MhaT
- 5) 0, MhaT
- 6) 3, TMha

Ergebnis: 2 3 0 3 0 3

Signalverarbeitung

Analoges vs. Digitales Signal

analog: kontinuierlicher Strom von Werten. Zu jedem beliebigen Zeitpunkt wird ein Wert zurückgegeben. Nachteil, es lässt sich nicht digital speichern.

digital: diskretes Signal (endliche Menge). Umwandlung analog -> digital ist verlustbehaftet, dafür nicht störanfällig wie analoges Signal.

Fehler bei Digitalisierung

- zu grobe Abtastrate, Missachtung der Nyquist-Grenze führt zur Rekonstruktion eines falschen Signals
 - zu grobe Quantisierung (z.B. aufrunden), zu starke Abstufung der Werte stellte Werte falsch dar.
- ➔ Fehler nennen sich **Aliasing**

Fouriertheorem

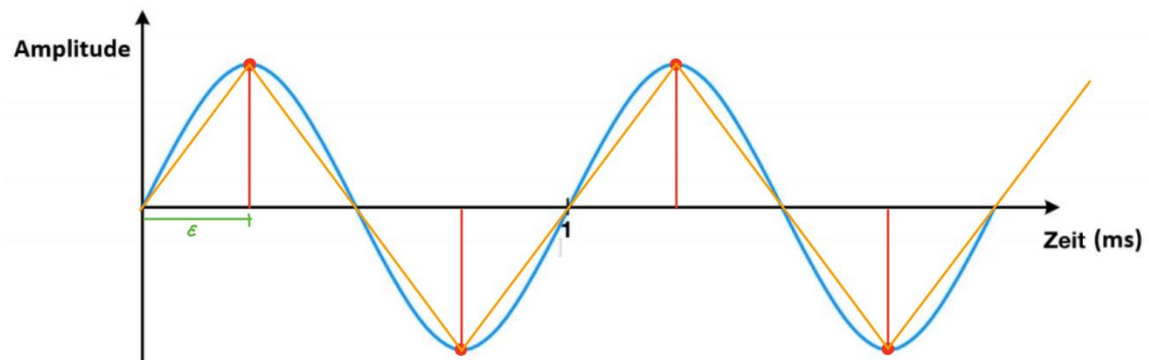
Ein periodisches Signal lässt sich als beliebige Summe von sin- und cos-Funktionen unterschiedlicher Frequenzen darstellen. Diese lassen sich durch Fouriertransformation darstellen.

Abtasttheorem (Nyquist-Grenze)

Quellsignal lässt sich nur aus dem abgetasteten Spektrum rekonstruieren, wenn die Abtastrate größer als die doppelte Grenzfrequenz f_{max} ist.

$$\text{Abtastrate } \frac{1}{T} > 2 \cdot f_{max}$$

Beispiel: Signal mit Grenzfrequenz von $10\text{kHz} \Rightarrow f_{abtast} > 20\text{kHz}$
➔ doppelte Abtastfrequenz bedeutet zwei Abtastpunkte pro Periode.



Zur besseren Visualisierung wird erster Abtastpunkt um ϵ verschoben. Abtastpunkte rot, Ergebnis orange.

Bildanalyse

Anwendungsbereiche

- Zeichenerkennung
- Qualitätsprüfung in industrieller Produktion
- Medizinische Bildanalyse
- Luftbildauswertung
- Gesichtserkennung
- Fahrzeugsteuerung
- Inhaltsbasierende Bildsuche im Internet

Operationstypen

Punktoperator: Eine Operation wird lediglich auf einen bestimmten Pixel ausgeführt, ohne dass diese abhängig von den Nachbarn oder gar dem ganzen Bild ist.

lokaler Operator: Lokale Operatoren beziehen eine begrenzte Region um das betrachtete Pixel mit ein, um z.B. anhand derer zu erkennen, ob es sich um einen fehlerhaften Pixel handelt, oder einen neuen Intensitätswert durch Gauss-Filter zu berechnen.

globaler Operator: Die Operation auf einen Pixel ist vom gesamten Bild abhängig.

Histogramm

Ein Histogramm zeigt die Häufigkeitsverteilung von bestimmten Werten aus einer Quelle. Für Bilder bedeutet das, dass sich zum Beispiel Farbanteile darstellen lassen, oder wie in dieser Übung, zwischen Intensitätswerten differenziert wird. Aus der Verteilung von Intensitätswerten lässt sich Kontrast und Dynamikumfang eines Bildes ableiten.

Dynamikumfang

Anzahl unterschiedlicher Intensitätswerte in einem Bild (Graustufenabstufung)

Abhängigkeit lokaler von globalem Kontrast

Wird der globale Kontrast eines Bildes erhöht, so werden dunkle Stellen noch dunkler und helle Stellen noch heller. Lokal betrachtet bedeutet das dann auch, dass der lokale Kontrast zwischen zwei benachbarten Pixeln, von denen das eine hell und das andere dunkel ist, ebenfalls erhöht ist.

Globaler Kontrast

Größter Grauwertunterschied in einem Bild

Lokaler Kontrast

Durchschnittliche Grauwertunterschied zwischen Nachbarn. Lässt sich durch log-Transformation oder Gamma-Korrektur verbessern.

Bildbearbeitung

Boxcarfilter

Mittelwert (Mean), glättet linear, Kanten werden verwaschen.

$$H(i, j) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Min-Filter

Eliminiert weiße Punkte und fördert dunkle Stellen

Max-Filter

Eliminiert dunkle Punkte und fördert weiße Stellen.

Median-Filter

Sortiert alle Werte in der Filterregion und wählt den mittleren Wert. Eliminiert Salt&Pepper Rauschen.

Gauß-Filter

Glättung, örtliche Intensitätsunterschiede werden abgeflacht.

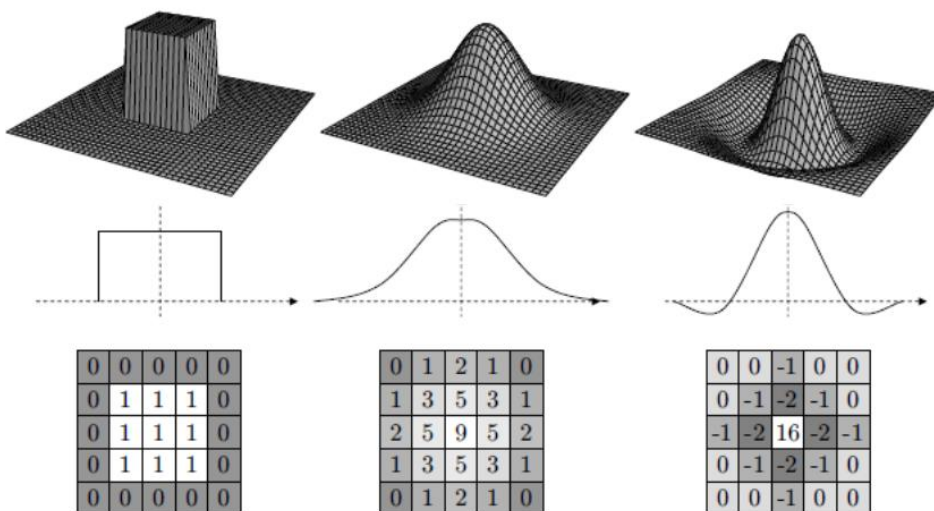
$$F_{Gau\beta} = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Salt&Pepper Rauschen kann nicht eliminiert werden, da zentrales Pixel höchste Gewichtung hat.

Laplace-Filter

Differenzbildung, Intensitätsunterschiede werden verstärkt.

$$\nabla^2 \approx \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



Boxcar-Glättungsfilter

Gauß-Glättungsfilter

Laplace-Differenzfilter

Durchschnittsbildung → Glättung
örtlicher Intensitätsunterschiede

Differenzbildung → Verstärkung
örtlicher Intensitätsunterschiede

UNTERSCHIED GAUSS LAPLACE TYPISCHES KLAUSURTHEMA

Prewitt Filter

Kantendetektion durch Rotation der Maske in 45° Schritten.

$$p_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad p_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad M \approx \sqrt{P_x^2 + P_y^2}$$

Kirsch-Operator

Kantendetektion durch Rotation.

$$h_1 = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{pmatrix}, \quad h_2 = \begin{pmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{pmatrix}, \quad h_3 = \begin{pmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{pmatrix} \dots \quad h_8 = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 0 & -5 \\ 3 & -5 & -5 \end{pmatrix}$$

Sobel-Operator

Kantendetektion durch Filter in x- und y-Richtung. S_x deckt vertikale Kanten auf, S_y deckt horizontale Kanten auf.

$$s_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad s_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad M \approx \sqrt{S_x^2 + S_y^2}$$

Roberts-Operator

Kantendetektion von diagonalen Kanten.

$$R_x = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad R_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Canny-Verfahren

Bessere Kantendetektion durch Verkettung mehrerer Verfahren.

1. Bild in Graustufen umwandeln
2. Gauß Filter anwenden
3. Sobel Operator anwenden
4. Non-Maximum Suppression (schmäleren der Kanten)
5. Hysterese (Schwellenwertfilterung)

Laplacian of Gaussian (LoG)

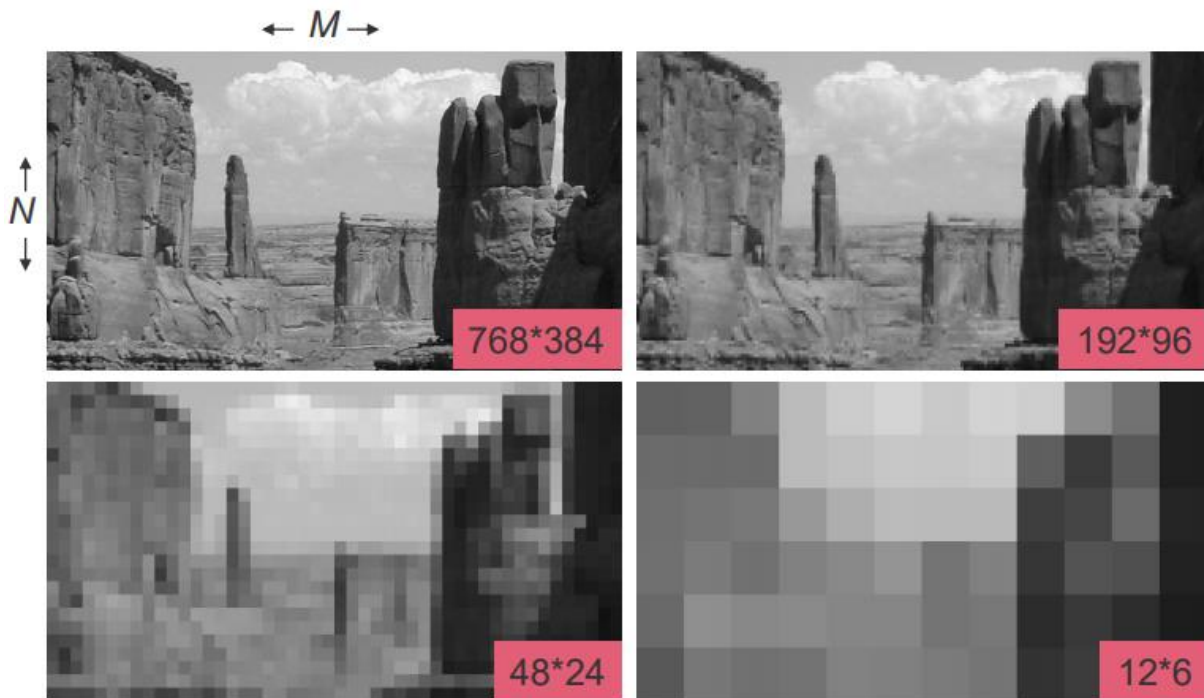
Kantendetektion durch Mexican Hat.

1. Gauß Filter
2. Laplace Operator

Bildkomprimierung

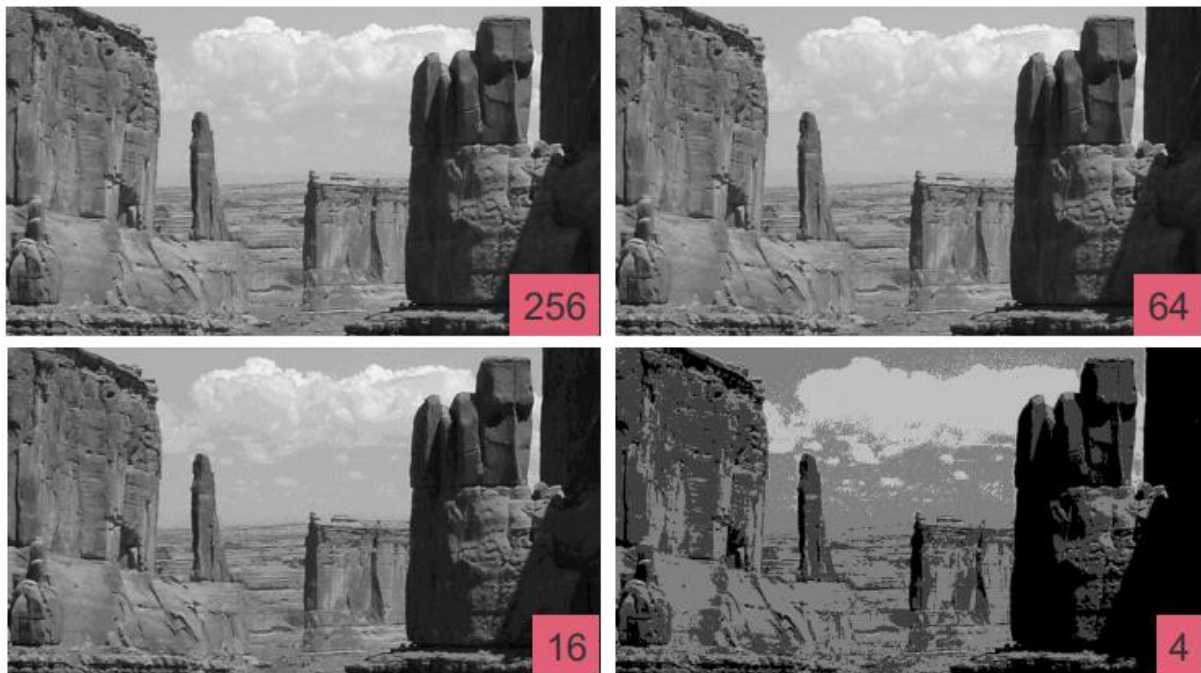
Ortsauflösung (Abtastung)

Pixelverhältnis in einem Bild. Umso kleiner die Ortsauflösung, desto schwerer Inhalte erkennbar.



Kontrastauflösung (Quantisierung)

Bittiefe der Intensitätswerte. Niedrige Quantisierung führt zu Verschwinden von Oberflächendetails, Formen allerdings immer noch gut erkennbar.



JPEG Kompression

1. **Farbraumumrechnung** (Zerlegung in RGB oder YCrCb) [optional]
2. **4:2:0 Subsampling** (verlustbehaftet)
3. **Aufteilung** Bild in 8x8 Blöcke
4. **Diskrete Cosinus Transformation** der 8x8 Blöcke.
5. **Quantisierung**, Werte aus DCT werden mit Quantisierungsmatrix gerundet (verlustbehaftet)
6. **Abtastung** durch Zick-Zack-Verfahren
7. **Entropiekodierung** der Werte durch RLE und Huffman

Diskrete Cosinus Transformation (DCT)

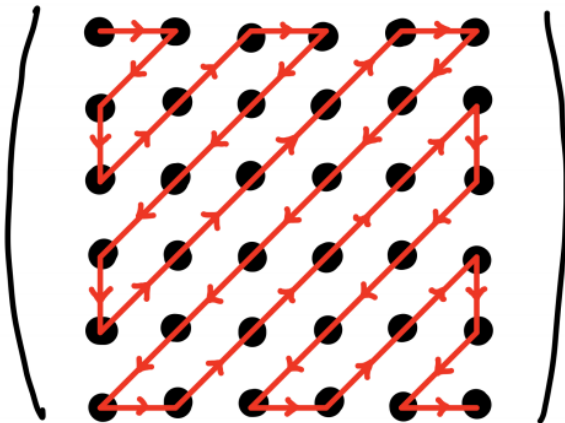
Im Gegensatz zur DFT werden nur reelle Werte erzeugt und die DCT wird in der Regel nur auf quadratische Matrizen angewandt. Im Ergebnis befindet sich dann links oben in der Ecke an Position (0,0) der DC-Koeffizient, welcher die am häufigsten auftretende Frequenz im Block repräsentiert. Die Restlichen Werte werden AC-Koeffizienten genannt.

Beliebte Prüfungsfrage: Vorteile DCT gegenüber DFT?

„Energieverdichtung“ in Richtung oben links (häufige Frequenzen sammeln sich links oben im Eck) und schnellere Berechnung (da nur reelle Parts berechnet werden).

Zick-Zack Abtastung

Durch das Abtastmuster werden zuerst die Höheren Frequenzen aufgenommen. Dadurch bekommen diese mein RLE weniger Bits als Kodierung.



Videokompression

Verfahrenskategorien

- **Intra-Bild-Codierung:** Operationen werden auf einen einzelnen Frame ausgeführt
- **Inter-Bild-Codierung:** Operationen werden durch Abhängigkeiten zwischen mehreren Frames ausgeführt

Kompressionsverfahren

Vektorquantisierung (intra): Ein Bild wird in kleine Blöcke aufgeteilt und dann nach Ähnlichkeiten zwischen den Blöcken gesucht. Durch die Ähnlichkeiten wird eine Art Durchschnittsblock berechnet, der die ähnlichen Blöcke dann jeweils ersetzt. Die Durchschnittsblöcke werden in einer Palette gesammelt und das Bild durch darauf verweisende Indizes kodiert.

Konturbasierte Kodierung (intra): Ein Bild wird in Konturen und Texturen aufgeteilt. Konturen, also Kanten, werden durch Bezier-Kurven beschrieben. Texturen, also Flächen, werden durch DCT kodiert.

Differenzcodierung (inter): Ausgehend von einem intracodierten Startbild werden für die darauffolgenden Bilder lediglich die Differenzen zum Startbild erfasst und jeweils komprimiert.

Bewegungskompensation (inter): Bewegungen von einzelnen Objekten im Bild werden identifiziert. Für eine Bewegung entsteht dann ein Differenzbild und ein Verschiebungsvektor, durch die sich die Frames zwischen Beginn und Ende der Bewegung mit einem passenden Algorithmus rekonstruieren lassen.

Makroblock

Besteht aus DCT-codierten Farbkanälen:

4x 8x8 Blöcke mit Intensitätswerten (Y)

1x 16x16 Block mit blauen Chrominanzwerten (Cb)

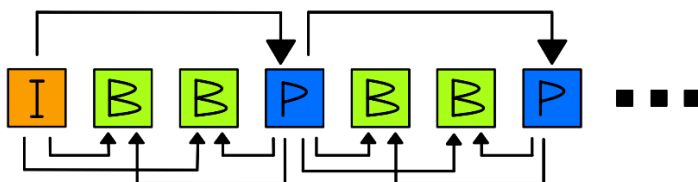
1x 16x16 Block mit roten Chrominanzwerten (Cr)

Frametypen

- **I-Frame:** Intracodiertes Bild, welches einzeln betrachtet werden kann.
- **P-Frame:** Interkodiertes Bild, Differenzbild vom vorherigen I- oder P-Frame.
- **B-Frame:** bidirektional interkodiertes Bild, Differenzbild zwischen vorherigem und nächsten I- oder P-Frame. → erreichen höchste Kompression.

Group of Pictures (GOP)

Dateneinheit aus mehreren Frames. Erster Frame muss ein I-Frame sein (klassisches Pattern: I, B, B, B, P, B, ...).



Folgen von Übertragungsfehlern

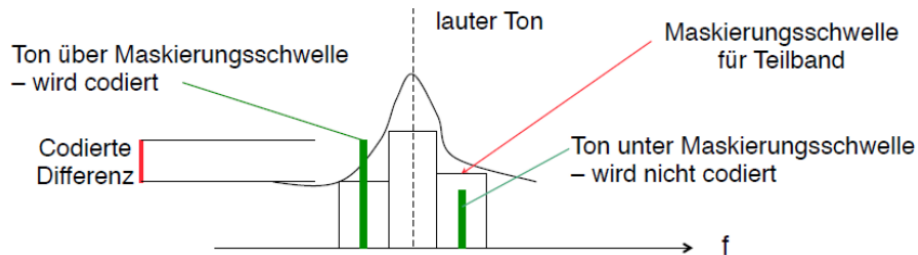
verlorenes I- oder P-Frame: erst nächste GOP kann wieder korrekt dekodiert werden.

verlorenes B-Frame: einzelnes Frame verloren, leichtes Ruckeln im Bild.

Audiokompression

Maskierungsschwelle

Vergleich einer Akustischen Grenze, ab der eine Frequenz hörbar ist. Liegt die Frequenz unter der Schwelle, ist sie nicht zu hören.

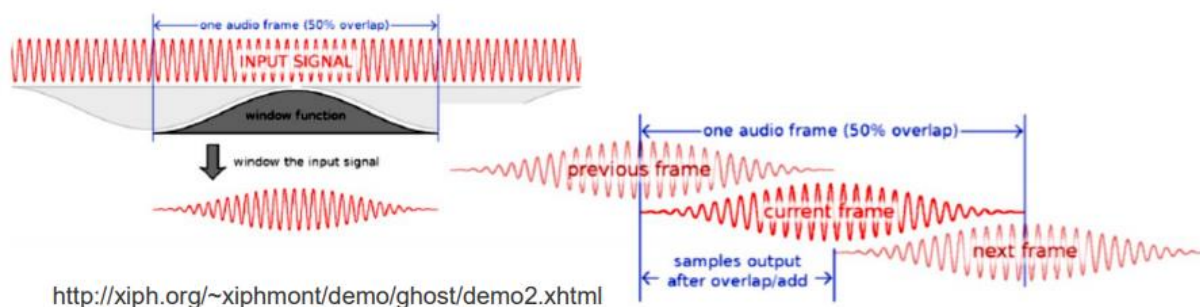


Psychoakustisches Modell

Unterteilung eines Signals in Frequenzbänder und Berechnung der jeweiligen Maskierungsschwelle anhand der Nachbarbänder.

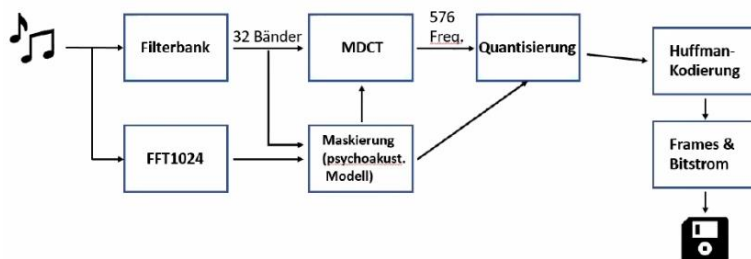
Modifizierte Diskrete Cosinus Transformation (MDCT)

Artefakte werden durch Überlagern von Signalblöcken vermieden.



MP3-Kompression

1. Subband-Transformation (Einteilung in 32 Subbänder)
2. MDCT-Transformation des Signals und Psychoakustisches Modell daraus bilden
3. Quantisierung (Maskierte Frequenzen werden entfernt)
4. Huffman Kodierung



Variable Bitrate

Bitrate wird dynamisch zu den kodierenden Daten angepasst. D.h. bei Signalen mit wenig Frequenzumfang (menschliche Stimme) wird sie runter gesetzt, bei hohem Frequenzumfang (Orchester) erhöht. Dadurch wird Bandbreite nicht verschwendet und ein optimales Qualität-Speicher-Verhältnis erreicht.

Videoanalyse

Schnitterkennung

Pixelbasierte Verfahren: Man berechnet, wie stark sich die einzelnen Pixel benachbarter Bilder verändern. Wenn die Summe der Veränderungen über eine gewisse Grenze gelangt, dann entscheidet man, dass ein harter Schnitt geschehen ist.

Vorteil: geringe Komplexität, robuste Ergebnisse

Nachteil: hohe Fehlerrate bei starken Bewegungen

Histogrammbasiertes Verfahren: Man berechnet, wie stark die Farbhistogramme benachbarter Bilder sich unterscheiden. Wenn Summe der Veränderungen wieder über einer Schwelle liegt, dann entscheidet man, dass ein Schnitt geschehen ist. Hiermit werden besonders gut Schnitte erkannt, in denen sich die Farben bzw. Helligkeiten stark unterscheiden, somit also eher harte Schnitte.

Problem: Unterschiedliche Szenen können ähnliche Histogramme haben.

Kantenbasiertes Verfahren: Die einzelnen Bilder einer Videosequenz werden mit einem Kantenfilter (Sobel, Canny) gefiltert. Die daraus resultierenden Kanteninformationen werden mit einem Schwellenwertverfahren quantisiert, dann zwischen den Bildern mit der ECR-Methode verglichen (betrachtet hinzugekommene, bzw. verschwundene Pixel zwischen zwei Kanten-Bildern). Dadurch kann man harte Schnitte und auch Bewegungen erkennen, sowie Einblendungen, Ausblendungen und Überblendungen, diese jedoch mit einer hohen Fehlerquote.

Kantenorientiertes Kontrast-Verfahren: Die einzelnen Bilder einer Videosequenz werden mit einem Kantenfilter gefiltert, der auch die Kantenstärke zurückgibt. Die Kanten werden nach Schwellwerten in starke oder schwache Kanten sortiert. Diese stellt man mit dem EC in ein Verhältnis, welcher zwischen 0 und 2 ist. Wenn der EC unter 1 ist, dann gibt es mehr schwache als starke Kanten und wir können von einer Überblendung ausgehen. Da es vor allem nur bildweise operiert und benachbarte Bilder nicht miteinbezogen werden, kann dieses Verfahren nur Überblendungen erkennen.

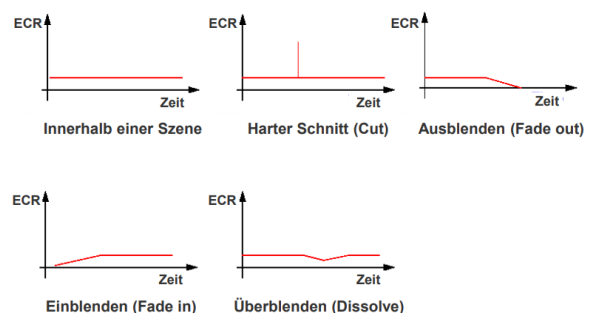
Edge Change Ratio (ECR)

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 zur Verfolgung von Bewegungsintensität verwendet werden



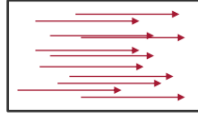
Objektbewegung vs. Kamerabewegung

Objektbewegung: einzelne Pixel verändern sich

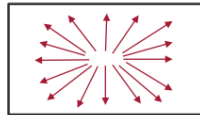
Kamerabewegung: alle Pixel verändern sich gleichmäßig

• Beispiele der Pixelbewegung von Bild i zu $i+1$

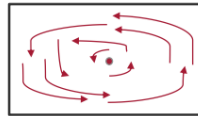
- Schwenk (nach links):



- Zooming (in):



- Kippen (nach rechts 90°):



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)

Genre-Erkennung

Erkennung von z.B. Werbung oder unterschiedlichen Sendungen in einer Aufnahme anhand von syntaktischen und semantischen Merkmalen.

Dynamic Time Warping

Dynamic Time Warping

Anpassen, bzw. Ausrichtung zeitlicher oder geometrischer Sequenzen.

1. Lokale Kostenmatrix berechnen $c(x_i, y_j) = |x_i - y_j|$
2. akkumulierte Kostenmatrix berechnen

$$d_{i,j} = \begin{cases} 0 & \text{wenn } i = j = 0 \\ \min(d_{i-1,j-1}, d_{i-1,j}, d_{i,j-1}) + c(x_i, y_j) & \text{wenn } i > 0 \text{ und } j > 0 \\ \infty & \text{sonst} \end{cases}$$

3. Warping Pfad visualisieren: günstigster Pfad von (1,1) bis (n,n), Kosten des Pfades stehen in der akkumulierten Matrix im Feld (n,n).

Levenshtein-Distanz

Distanz zwischen zwei Zeichenketten in Tabelle ermitteln.

$$d_{i,j} = \min \begin{cases} d_{i-1,j} + 1 & \text{(Löschen)} \\ d_{i,j-1} + 1 & \text{(Einfügen)} \\ d_{i-1,j-1} + 1 & \text{falls } a_i \neq b_j \text{ (Ersetzen)} \\ d_{i-1,j-1} & \text{falls } a_i = b_j \text{ (Übernehmen)} \end{cases}$$

Levenshtein-Distanz steht in Feld (n,n).

<i>L</i>	6	5	4	3	3	3	3	2	3
<i>A</i>	5	4	3	2	2	2	2	3	4
<i>S</i>	4	3	2	1	1	2	3	4	5
<i>T</i>	3	2	1	1	2	3	4	5	6
<i>U</i>	2	1	0	1	2	3	4	5	6
<i>F</i>	1	0	1	2	3	4	5	6	7
	0	1	2	3	4	5	6	7	8
		<i>F</i>	<i>U</i>	<i>S</i>	<i>S</i>	<i>B</i>	<i>A</i>	<i>L</i>	<i>L</i>

Maximale und minimale Levenshtein-Distanz

- **minimal** kann die Levenshtein-Distanz 0 betragen, wenn $A = B$.
- **maximal** kann die Levenshtein-Distanz $\max(|A|, |B|)$ betragen, wenn beide Wörter in ihren Buchstaben komplett verschieden sind.

Gestenerkennung

Kategorien

Sensorik am Körper

- Vorteil: hohe Präzision
- Nachteil: Sensoren am Körper (eingeschränkte Bewegung)

Externe (optische) Sensorik

- Vorteil: Bewegungsfreiheit
- Nachteil: größere Fehlerrate, Merkmalspunkte müssen algorithmisch erkannt werden

Tiefendatengenerierung

Stereo-Vision

1. **Kamerakalibrierung:** Ausrichtung der Kamera wird ermittelt, um Verzerrung zu erkennen
2. **Eliminierung der Verzerrungen:** geschwungene Kanten werden gerade
3. **Rektifizierung:** Stereo-Bilder werden ausgerichtet, dass beide in eine Bildebene gesetzt sind
4. **Disparitäten berechnen:** Versatz eines Objektes berechnen, betrachtet durch zwei Okulare (Unterschied linkes – rechtes Auge)
5. **Tiefenkarte berechnen:** Abstand eines Objekts durch Disparität und Brennweite berechnen
6. **Tiefenkarte optimieren:** Löcher in Tiefenkarte durch Median Filter oder Interpolation beseitigen

Kinect

1. generieren eines zufälligen Specklemusters (heterogenes Punktmuster)
2. Muster wird auf Objekt projiziert
3. Muster auf Objekten wird detektiert
4. matching zwischen detektiertem und gespeichertem Muster
5. Berechnung von Tiefeninformationen anhand der resultierenden Verzerrungen