

# Bildverarbeitung

## Antworten zu den Fragen

Manuel Pauli

Sebastian Schweikl

Thomas Lang

12. Juli 2016

## 1 Allgemein

### Erkläre Operationen: Skalierung, Translation

- Translation:  $\tau_k f = f(\bullet + k)$  verschiebt die Funktion  $f$  um  $k$  Einheiten in  $x$ -Richtung ( $k < 0$  Verschiebung nach rechts,  $k > 0$  Verschiebung nach links).
- Skalierung:  $\sigma_h f = \sigma(h \cdot \bullet)$  Multiplikation der Frequenz mit Faktor  $h$  ( $h > 1$  bedeutet Stauchung,  $h < 1$  bedeutet Streckung)

**Erkläre das Prinzip einer Lochkamera** Eine Lochkamera ist der Vorläufer einer modernen Kamera. Dabei fällt das natürliche Licht durch ein kleines Loch (die Blende) ein. Das Bild trifft dann auf der Rückwand der Kamera auf. Bedingt durch die Geometrie steht das Bild auf dem Kopf, dessen Höhe/Breite wird dabei durch den Abstand des Objektes zur Blende, des Abstandes der Blende zur Rückwand und von der Größe des Objektes und der Blende bestimmt.

**Wie kann man das Bild einer Lochkamera modellieren?** Ein Bild ist zunächst ein zweidimensionales diskretes Signal. Es gibt genau drei Möglichkeiten, dies zu modellieren:

- Volldiskret: Abbildung  $I: \{0, \dots, w\} \times \{0, \dots, h\} \rightarrow \{0, \dots, 255\}$ . Das ist genau das Modell, wie wir z.B. Bilder in der Realität messen. Jedes Bild besitzt die Ausmaße  $w \times h$  Pixel und  $I$  bildet Pixel z.B. auf einen Farbwert von 0 bis 255 ab (Grauwertbild).
- Semidiskret:  $I: \{0, \dots, w\} \times \{0, \dots, h\} \rightarrow \mathbb{R}$ . Bei diesem Modell arbeitet man weiterhin mit diskret vielen Pixeln, besitzt aber kontinuierlich viele Werte, die die Farbe angegeben können. Unserer Meinung nach ist dieses Modell nicht sehr sinnvoll für unsere Anwendungszwecke.
- Kontinuierlich:  $I: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ . Dieses Modell ist besonders gut geeignet, um Mathematik damit zu machen. Die ganze Theorie aus der Analysis greift hier (Stetigkeit von Funktionen usw.). Hier besitzen wir kontinuierlich viele Bildpunkte und kontinuierlich viele Farbwerte. Durch Interpolation (Spline-Interpolation, Polynom-Interpolation) können wir unser diskretes Bild in den kontinuierlichen Raum hochlif-

ten, dort unsere mathematische Theorie anwenden, und z.B. durch Diskretisierung wieder diskretisieren (wobei die genauen Verfahren dazu natürlich von der konkreten Problemstellung abhängen).

## 2 Fourier

### 2.1 Fouriertransformation

**Nenne die Gleichung** Für eine Funktion  $f \in L_1(\mathbb{R})$  ist die Fouriertransformation durch

$$\hat{f}(\xi) = \int_{\mathbb{R}} f(t) e^{-i\xi t} dt$$

für alle  $\xi \in \mathbb{R}$  definiert.

**Das ist die Gleichung für  $L_1$ -Funktionen. Wie sieht das im  $L_2$  aus?** Die Fouriertransformation im Signalraum  $L_2$  kann ganz analog zum obigen Fall verwendet werden. Die resultierende Fouriertransformierte liegt dann in  $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ .

Formal beruht dies darauf, dass der Raum  $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$  dicht in  $L_2(\mathbb{R})$  liegt. Deshalb lässt sich also immer eine Funktionenfolge aus  $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$  finden, die im Grenzwert gegen unsere gesuchte Fouriertransformierte konvergiert<sup>1</sup>.

**Was sagt der Satz von Parseval/Plancherel?** Der Satz von Parseval/Plancherel beschreibt den Wechsel zwischen einem Skalarprodukt im Zeit- und einem Skalarprodukt im Frequenzbereich.

Formal gilt für Funktionen  $f, g \in L_1(\mathbb{R})$ :

$$\int_{\mathbb{R}} f(t) g(t) dt = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\vartheta) \overline{\hat{g}(\vartheta)} d\vartheta$$

Im Speziellen für  $f = g$  folgt, dass

$$\|f\|_2^2 = \frac{1}{2\pi} \|\hat{f}\|_2^2$$

gilt, also dass hier (bis auf Normierung) eine Isometrie vorliegt.

**Sie sprachen von Isometrie, warum ist dann der konstante Faktor vor dem rechten Term?**

Dieser Faktor kommt daher, weil wir bei der Definition der Fouriertransformation keinen Faktor dabei hatten. Hätte man dort einen Faktor  $1/\sqrt{2\pi}$  hinzugefügt, so hätte man hier eine perfekte Isometrie, was man ja mit Energieerhaltung<sup>2</sup> identifizieren kann.

**Wie sieht die inverse Fouriertransformation aus?** Unter der Voraussetzung, dass sowohl für  $f$  als auch für  $\hat{f}$  gilt, dass diese Funktionen aus  $L_1$  sind, ist die inverse Fouriertransformation definiert als

$$f(\xi) = (\hat{f})^\vee(\xi) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(t) e^{i\xi t} dt$$

---

<sup>1</sup>Geht immer, zwecks Banachraum und so (d.h. vollständiger Raum, der Grenzwert zu jeder Folge in diesem Raum sitzt wieder in diesem Raum).

<sup>2</sup>Die 2-Norm  $\|\cdot\|_2$  beschreibt ja i.W. Energie.

**Zusammenhang zwischen Faltung und Fouriertransformation** Ein wichtiger Zusammenhang ist, dass sich die Faltung zweier Funktion als Produkt ihrer Fouriertransformierten darstellen lässt (vorausgesetzt deren Existenz).  
Es gilt also:

$$\begin{aligned}
 \widehat{(f * c)}(\xi) &\stackrel{\text{def}}{=} \int_{\mathbb{R}} (f * c)(t) e^{-i\xi t} dt \\
 &\stackrel{\text{def}}{=} \int_{\mathbb{R}} \int_{\mathbb{R}} f(t-x) c(x) dx e^{-i\xi t} dt \\
 &\stackrel{\text{Tonelli}}{=} \int_{\mathbb{R}} \int_{\mathbb{R}} f(t-x) c(x) e^{-i\xi t} dx dt \\
 &\stackrel{t=t+x}{=} \int_{\mathbb{R}} \int_{\mathbb{R}} f(t) c(x) e^{-i\xi(t+x)} dx dt \\
 &= \int_{\mathbb{R}} \int_{\mathbb{R}} f(t) e^{-i\xi t} c(x) e^{-i\xi x} dx dt \\
 &\stackrel{\text{Tonelli}}{=} \int_{\mathbb{R}} f(t) e^{-i\xi t} dt \int_{\mathbb{R}} c(x) e^{-i\xi x} dx \\
 &= \hat{f}(\xi) \hat{c}(\xi)
 \end{aligned}$$

**Erkläre den Weg zur FFT** Der Weg von der DFT zur FFT ist der, dass man einen Baumalgorithmus implementiert. Konkret arbeitet ein Teilbaum alle Folgenglieder mit geradem Index ab, während der andere Teilbaum den Rest bearbeitet, wobei sich natürlich die Indexmenge in den Teilbäumen vom Vaterknoten verändert.

Formal habe der Vaterknoten die Indexmenge  $\mathbb{Z}_n = \{0, \dots, n-1\}$  zur Folge  $c$  und sei  $n = 2 * m, m \in \mathbb{N}$ . Dann ergibt sich für die Teilbäume, dass der eine alle Folgenglieder  $c(2k)$  und der andere alle Folgenglieder  $c(2k+1), k \in \mathbb{Z}_m = \{0, \dots, m-1\}$  verarbeitet.

**Wie schnell ist die FFT?** Da man dabei einen Baumalgorithmus implementiert, hat man also nur  $\log n$  Ebenen abzuarbeiten, wobei man immer noch alle Folgenglieder  $c(k), k \in \mathbb{Z}_n$  verarbeiten muss, es entsteht also ein Aufwand in  $\mathcal{O}(n \log n)$ . Nachzuweisen ist dies mit dem Mastertheorem.

**Theorie schön und gut, aber wozu braucht man die Fouriertransformation nun in der Praxis?**  
Mit der ganzen Theorie (insbesondere mit dem Zusammenhang zwischen der Faltung und der Fouriertransformation) können Bildverarbeitungsalgorithmen in verschiedenster Hinsicht optimiert werden. Beispielsweise um die reine Berechnung zu beschleunigen (Multiplikation vs Faltung) oder um die numerische Stabilität zu erhöhen.

Der Standardweg ist dabei, dass man das Signal fouriertransformiert, auf dem Ergebnis die Berechnungen durchführt und anschließend wieder zurücktransformiert.

**Wie funktioniert das JPEG-Kompressionsverfahren mit Hilfe der DFT?** Die Idee ist hierbei, die Tatsache der Abhängigkeitsfreiheit zu nutzen um den Algorithmus massiv parallel zu machen. Dabei ist noch nicht mal die FFT gemeint, sondern das Vorgehen das Bild

in Makroblöcke der fixen Größe  $8 \times 8$  aufzuspalten. Diese fixe Größe kommt daher, dass man also pro Makroblock 16  $1 \times 4$ -Vektoren vorliegen hat, welche z.B. auf Graphikkarten oder modernen CPUs sehr effizient verarbeitet werden können. Falls die Dimensionen des Bildes keine Vielfachen von 8 sind, so können die fehlenden Ränder z.B. auf 0 gesetzt oder das Bild periodisch fortgesetzt werden.

Hat man nun die Makroblöcke vorliegen, so wendet man auf diese die Transformation an (sei es die DFT, die FFT oder die DCT). Das Ergebnis wird dann mit einem Kompressionsverfahren verknüpft, um die Größe jedes Makroblocks (und damit die Bildgröße) dramatisch zu verringern. Beispiele dafür wäre z.B. der Huffman-Code, welcher eindeutige Codes unterschiedlicher Länge generiert, die die ursprünglichen Werte darstellen. Dieses Encoding wäre sogar verlustfrei.

Bei der Dekompression geht man entsprechend umgekehrt vor: Man Ersetzt die gekürzten Codes mit den ursprünglichen Werten und wendet die inverse Transformation (IDFT, IFFT, IDCT) an, um das alte Bild z.B. darstellen zu können.

**Macht das JPEG wirklich so?** Nein, da die DFT/FFT u.U. imaginäre Werte liefert. Diese lassen sich aber nur auf wenigen (heute nicht mehr aktuellen) Maschinen in Hardware darstellen, und die Verarbeitung in Software dauert sehr lange.

Als Alternative wird daher in JPEG (vor Standard JPEG-2000) die *Diskrete Cosinus-Transformation*(DCT) verwendet, welche den selben Effekt wie die DFT erzielt, aber nur reelle Werte liefert.

**Wo liegen die Informationsmaxima?** Die Informationsmaxima liegen am Rand, da die hochfrequenten Anteile dort liegen.

**Nenne die Formel der DFT** Die Formel ergibt sich aus der Abtastung der Fourier-Transformation für Folgen mit Schrittweite  $2\pi/n$  für ein  $n \in \mathbb{N}$ , also

$$\widehat{c_n}(\xi) = \sum_{k \in \mathbb{Z}} c(k) e^{-2\pi i k \xi / n}.$$

**Fouriertransformation im  $\mathbb{R}^2$**  Die Fouriertransformation im zweidimensionalen Raum ist entsprechend die Variante für Bilder.

Hierbei hat man also Punkte  $p = (x, y) \in \mathbb{Z}^2$ , entsprechen betrachtet man auch die Fouriertransformation in zwei unterschiedlichen Richtungen (in  $x$ - und in  $y$ -Richtung). Formal übersetzt ergibt sich also die Fouriertransformation im  $\mathbb{R}^2$  als

$$\hat{f}(\xi) = \int_{\mathbb{R}^2} f(t) e^{-i \xi^T t} dt \quad , \xi, t \in \mathbb{R}^2$$

## 2.2 Faltung

**Was ist formal eine Faltung?** Formal gesehen ist eine Faltung ein Integral. Dieses ist für Funktionen  $f, g \in \mathbb{R}^n \rightarrow \mathbb{C}$  definiert als

$$(f * g) = \int_{\mathbb{R}^n} f(\cdot - t) g(t) dt.$$

Prinzipiell ist dieser Ausdruck definiert für alle Funktionen, die für *fast alle*  $x \in \mathbb{R}$  wohldefiniert sind. In unserem eingeschränkten Kontext der integrierbaren Funktionen ist dies automatisch erfüllt.

**Wie kann man sich so eine Faltung graphisch vorstellen?** Man kann sich das so vorstellen, dass man eine Funktion über die zweite *schiebt* und dort die Übereinstimmung berechnet. Bei einer hohen Übereinstimmung zwischen den Funktionen wird dieses Integral einen hohen Wert liefern, während sich bei wenig Übereinstimmung nur ein kleiner Wert ergibt.

**Was bringt so eine Faltung?** Das im letzten Punkt beschriebene Verhalten der Faltung kann man sich z.B. in der Texterkennung zu nutze machen. Dabei prüft man z.B., ob es eine hohe Übereinstimmung zwischen einem Bild eines Buchstabens und dem vorliegenden Bild gibt, indem man über diese Bildteile die Faltung berechnet. Ein hoher Wert indiziert dabei, dass mit hoher Wahrscheinlichkeit der gesuchte Buchstabe vorliegt. Ganz allgemein ist dieses Prinzip die Grundlage von Filterungen.

### 3 Filter

**Was ist ein Filter?** Ein Filter ist ein Operator, der ein gegebenes Signal in ein anderes Signal transformiert.

**Wie sieht ein Filter allgemein aus?** Ich verstehe die Frage nicht, was ist damit gemeint?

**Wie sieht ein Tiefpass-Filter aus? Zeichne den Graph** Ein Tiefpass-Filter  $f$  lässt wie der Name sagt die tiefen Frequenzen durch, entsprechen wir sich für die obere Frequenz  $\xi$  ergeben, dass  $\hat{f}([0, \xi]) > 0$  ist, während er für die höheren Frequenzen sperrt, also  $\hat{f}((\xi, 2\pi]) = 0$ .

Ein idealer Tiefpass-Filter ist dabei die Rechtecksfunktion, welche direkt an der Frequenzgrenze  $\xi$  abschneidet. Dies ist aber keinesfalls zu empfehlen (schon alleine aus numerischen Gründen), daher wird dieses Rechteck i.d.R. abgerundet und leicht deformiert sein.

**Forderungen an die Transferfunktion für Tiefpass** Siehe vorheriger Punkt.

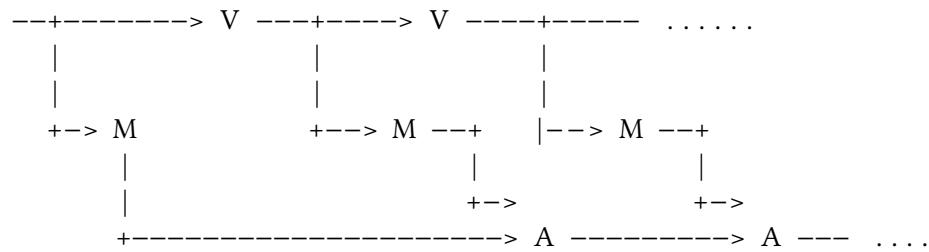
**Erkläre das Gibbs-Phänomen** Das Gibbs-Phänomen sind *Überschwinger*, welche bei der Bildung von Partialsummen auftreten. Dieses Phänomen lässt sich nicht einfach vermeiden, da diese Überschwinger höchstens enger zusammenrücken, aber in der Amplitude nicht kleiner werden. Um dieses Phänomen dennoch zu vermeiden muss man mit Approximationen arbeiten.

**Welche Arten von Filter gibt es? Erklären Sie diese** Es gibt i.W. sechs verschiedene Unterarten von Filtern:

1. Energieerhaltende Filter, wenn sie normiert sind.
2. Lineare Filter, wenn dieser eine lineare Funktion ist.

3. Zeitinvarianter Filter, wenn es egal ist, ob zuerst gefiltert und translatiert oder ob zuerst translatiert und anschließend gefiltert wird.
4. Kausale Filter, wenn der Filter nicht *in die Zukunft sehen* kann, also wenn das Ergebnis nur von den reinen Eingabedaten abhängt. Dementsprechend gibt es auch anti-kausale Filter, die nicht in die Vergangenheit schauen können, sondern nur in die Zukunft.
5. FIR-Filter, wenn die Impulsantwort des Filters endlich ist.
6. IIR-Filter, wenn die Impulsantwort des Filters unendlich ist.

**Zeichen Sie ein Schaltbild zu einem Filter (Addierer, Verzögerer, Multiplizierer)** Es stehe A für einen Addierer, M für einen Multiplizierer und V für einen Verzögerer. Dann hat ein Filter die prinzipielle Form:



**Was ist eine Impulsantwort?** Der Filter angewandt auf den Dirac-Puls. Man kann nachweisen, dass jeder Filter nur durch den Dirac-Puls beschrieben werden kann und man dadurch Informationen über das Verhalten des Filters hat.

**Spielt die Laufzeit von Filtern in der Praxis eine Rolle?** Das kommt natürlich immer auf die Anwendung an. Will ich z.B. ein Audio- oder Videosignal in Echtzeit filtern, so muss der Filter natürlich sehr schnell arbeiten. Ein anderes Beispiel ist, wenn Video- und Audiosignal miteinander synchronisiert werden müssen.

### 3.1 Was ist ein Gradientenfilter?

**Was ist ein Gradientenfilter?** Das ist ein Filter, der die Differenzen in den Farbwerten der Pixeln erkennt. Im kontinuierlichen Fall könnte man sagen, er erkennt die Steigung zwischen zwei Pixeln. Nachteil: Anfällig auf Rauschen, wenn sehr feine Strukturen in den Bildern vorhanden sind. Abhilfe: Erst entrauschen, um die feinen Strukturen loszuwerden, und dann Laplace-Filter: Gradientenfilter, der mit Mittelwertfilter kombiniert wurde und dann diskretisiert wurde.

**Wozu ist er gut?** Mit einem Gradientenfilter können Kanten in Bildern erkannt/hervorgehoben werden.

**Wie funktioniert er?** Ist eine solche Kante in einem Bild, so muss dort ein signifikant anderer Farbwert vorhanden sein. Dies macht man sich zu nutze, indem man sich die Steigung zwischen zwei benachbarten Farbwerten ansieht: Ist diese Steigung groß, so sind die Farbwerte grob unterschiedlich (beispielsweise eine schwarze Kante auf einem weißen Hintergrund), was ein Indikator für eine Kante ist.

**Wie sieht so ein Filter aus? (Gradient + Filtermatrix)** Wie schon erwähnt betrachtet man die Steigungen. Kontinuierlich betrachtet ist dies aber gerade die erste Ableitung, was im mehrdimensionalen der Gradient

$$\nabla f = \begin{pmatrix} \frac{\partial}{\partial x_1} f \\ \vdots \\ \frac{\partial}{\partial x_n} f \end{pmatrix}$$

ist. Um diesen nun auf ein diskretes Signal (z.B. ein Bild) anwenden zu können, muss man diesen geeignet diskretisieren. Dies ergibt die Filtermatrizen in  $x$ - und  $y$ -Richtung

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

**Nachteile + mögliche Gegenmaßnahmen** Ein wesentlicher Nachteil eines reinen Gradientenfilters ist, dass dieser gegen Rauschen extrem empfindlich ist, da auch diese feinen Unterschiede unterschiedliche Farbwerte und damit potenzielle Kanten liefern. Um dies zu beheben, kann man den Gradientenfilter mit einem Mittelwertfilter kombinieren, welcher das Bild glättet und damit Rauschen entfernt, gefolgt vom eigentlichen Gradientenfilter.

**Wie sieht so ein Filter aus?** Obiges Vorgehen liefert z.B. diese neuen Filtermatrizen:

$$\frac{1}{9} \begin{pmatrix} -1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 \end{pmatrix}, \quad \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

**So viel zu Gradientenfilter, was ist aber dann ein Laplace-Filter?** Ein Laplace-Filter ist eine Diskretisierung des Laplace-Operators

$$\Delta = \langle \nabla, \nabla \rangle,$$

welcher offensichtlich beide Richtungen kombiniert. Eine Diskretisierung ergibt z.B. die Filtermatrix

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

### 3.2 Filterbänke

**Skizziere eine Filterbank und beschreibe den Vorgang** Skizze siehe Abbildung 1.

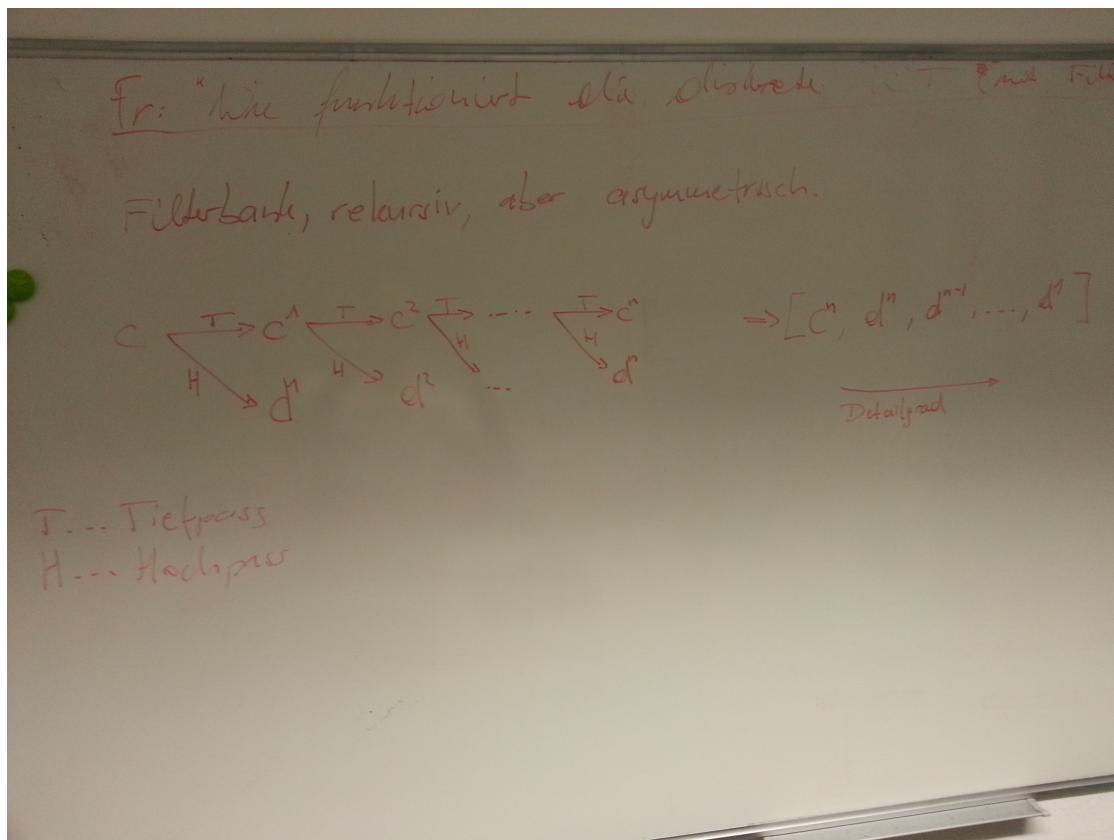


Abbildung 1: Skizze einer Filterbank.

**Nenne 3 typische Filter** Binomialfilter (Glättung), Medianfilter (Glättung), Laplace-Filter (Kantendetektion).

**Was ist die zentrale Eigenschaft von Filterbänken? Welche Voraussetzung muss dazu gelten?**

Man startet mit einem Signal  $c$ , generiert daraus den Polyphase-Vektor  $P$ , stecke diesen in die Analyse-Filterbank. Das Ergebnis davon kommt in die Synthese-Filterbank, daraus bekomme ich einen weiteren Polyphase-Vektor, den ich dann ein Ausgangssignal  $c'$  umwandeln kann. Der Polyphase-Vektor ist sozusagen eine vektorwertige Abtastung des Signals auf dem Einheitsmatrix. Die Modulationsmatrix  $M$  ist eine darstellende Matrix der Analyse-Filterbank, und  $\tilde{M}$  die darstellende Matrix der Synthese-Filterbank. Soll Perfect Reconstruction gelten, d.h. dass  $c = c'$  gilt, dann muss ich alle Änderungen, die  $M$  gemacht hat, auch wieder rückgängig machen können durch  $\tilde{M}$ . Mathematisch:  $M\tilde{M} = I$ . Voraussetzungen an die Filter in der Filterbank werden bereits durch die Modulationsmatrizen  $M$  und  $\tilde{M}$  gegeben.

Mathematische Formeln:

- Polyphase-Vektor

$$P = \left( c^*(e^{2i\pi k/n} z) : k \in \mathbb{Z}_n \right) \in \mathbb{C}^n.$$

- Darstellende Matrix

$$M(z) = \frac{1}{n} \left( f_j^*(e^{2\pi i k/n}) : j, k \in \mathbb{Z}_n \right) = \frac{1}{n} \begin{pmatrix} f_0^*(z) & \dots \end{pmatrix} \in \mathbb{C}^{n \times n}.$$

Dabei ist  $f_j$  jeweils die Impulsantwort eines Filters  $F_j$

**Kann man Filterbänke hierarchisch aufbauen?** Natürlich kann man das! Dieses Ding nennt man dann Subband-Coding. Beispiel:  $n = 2$ . Dann starte ich mit meinem Signal, filtere das mit einem Tiefpassfilter  $F_0$  und Hochpassfilter  $F_1$ . Dann mache ich ein Downsampling um den Faktor 2. Dies liefert mir im Falle des Tiefpassfilters ein Singal  $c_0$ . Dieses Signal kann ich nun auf die gleiche Art und Weise wieder rekursiv zerlegen. Dabei verwende ich wieder den Filter  $F_0$  (Tiefpass) und ein Downsampling um Faktor 2, um ein Singal  $c_{00}$  zu bekommen. Analog erhalte ich  $c_{01}$  durch den Hochpassfilter. Das war die Analysephase.

In der Synthesephase nehme ich meine Signale  $c_{00}$  und  $c_{01}$ , wende den Upsampling-Operator um Faktor 2 an, wende dann die Filter  $G_0$  und  $G_1$  an, fasse die beiden Ergebnisse zu einem Signal  $c_0$  zusammen. Und mit  $c_0$  und  $c_1$  kann ich dann das gleiche wieder rekursiv machen. Zum Schluss erhalte ich das Signal  $c$ , welches mit dem ursprünglichen Eingabesignal übereinstimmen kann, aber nicht muss. Signal

**Wie funktioniert die Diskrete Wavelet-Transformation mit Filterbänken?** → siehe Abbildungen 2 und 3.

## 4 Abtastsatz

**Welche Eigenschaften müssen für eine Abtastung gelten?** Funktion muss integrierbar sein.

Schrittweite der Abtastung muss kleiner sein als  $h^*$ , um Unterabtastung zu vermeiden, und die Funktion muss bandbeschränkt sein, weil man nicht unendlich viele Frequenzen rekonstruieren kann.

**Erkläre den Shannonschen Abtastsatz** Der Shannon'sche Abtastsatz sagt, dass wenn man eine  $T$ -bandbeschränkte  $L_1$ -Funktion fein genug abtastet, dann kann man aus dieser Abtastung durch eine semi-diskrete Faltung mit der *sinc*-Funktion die Originalfunktion eindeutig rekonstruieren.

**Wie funktioniert Abtasten überhaupt?** Beim Abtasten mit Faktor  $h$  nimmt man nur jeden  $h$ -ten Funktionswert in die Evaluation auf. Formal nimmt man also nur jeden  $c(h \cdot)$ -ten Wert auf.

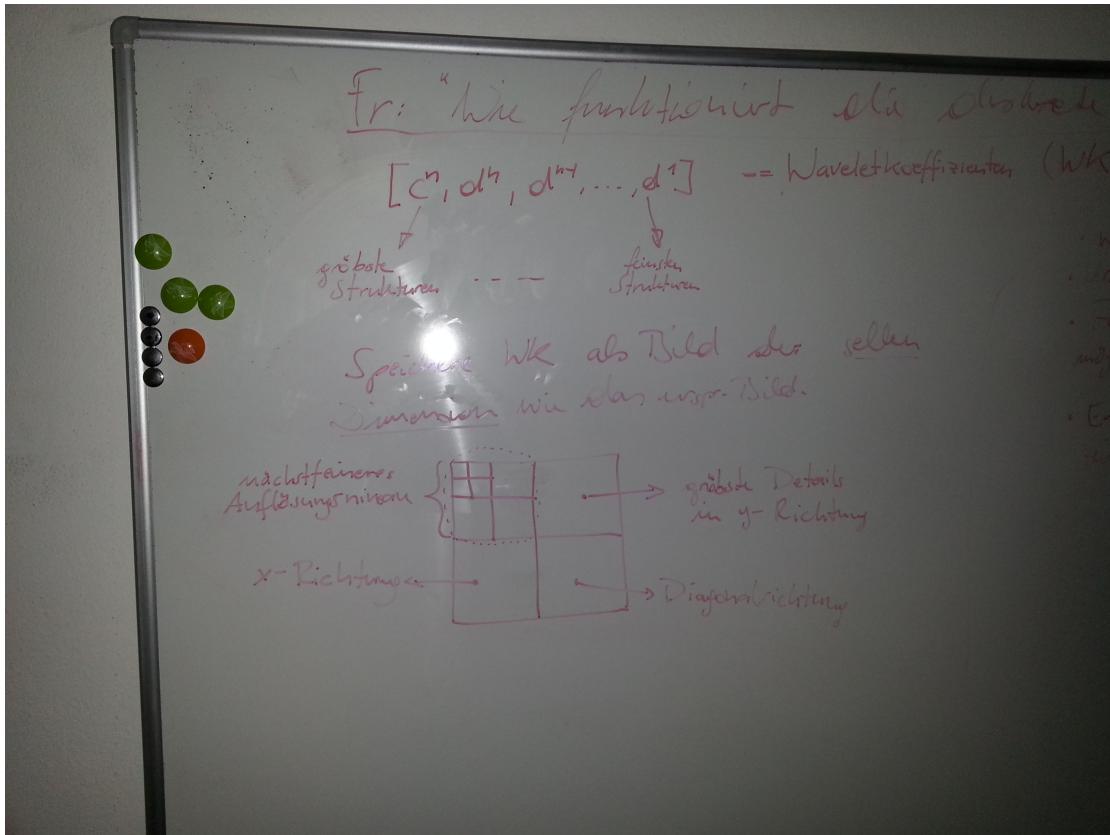


Abbildung 2:

**Was ist die kritische Abtastrate?** Die kritische Abtastschrittweite ist der Kehrwert der so genannten Nyquist-Frequenz. Das bedeutet, dass man mit einer höheren Frequenz als der Nyquist-Frequenz (und damit mit einer kleineren Abtastschrittweite) abtasten muss, um die Funktion (sofern überhaupt möglich) eindeutig aus der Abtastung rekonstruieren zu können.

Tastet man zu grob ab, so kommt es zu *Überlagerungen*, wo man im Nachhinein nicht mehr sagen kann, zu welchem Teil des Signals welcher Teil der Überlagerung gehört.

**Wie berechnet man die kritische Abtastrate?** Die Nyquist-Frequenz einer  $T$ -bandbeschränkten Funktion ist gegeben durch  $T/\pi$ , entsprechend ergibt sich die kritische Abtastschrittweite durch deren Kehrwert, also  $h^* = \pi/T$ . Um nun die angesprochene eindeutige Rekonstruktion zu bewerkstelligen, muss man mit einer Schrittweite von  $h < h^*$  abtasten.

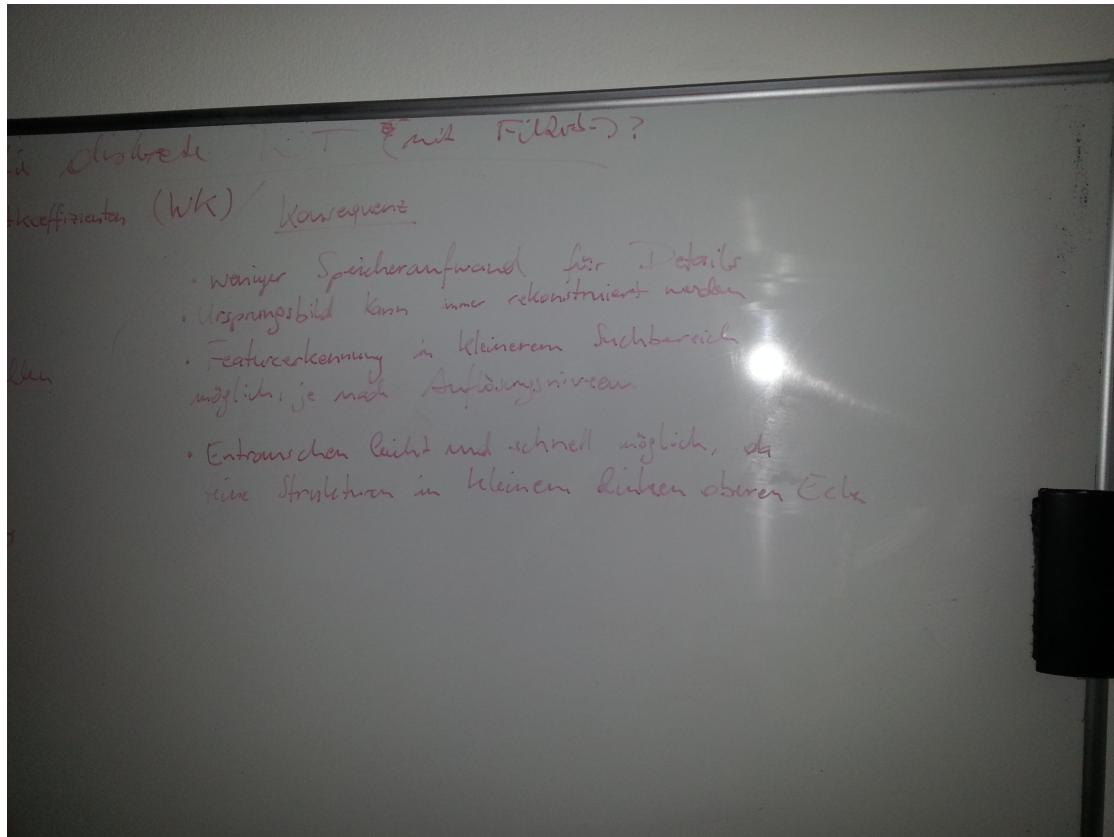


Abbildung 3:

**Was ist ein bandbeschränktes Signal?** Ein bandbeschränktes Signal  $f \in L_1(\mathbb{R})$  ist ein Signal, welches nur ein *beschränktes Band* an Frequenzen  $([-N, N], N \in \mathbb{R})$  beinhaltet<sup>3</sup>, formal:

$$\hat{f}(\xi) = 0 \Leftrightarrow \xi \notin [-N, N] \quad , \xi \in \mathbb{R}$$

**Was ist der Träger eines Signals?** Der Träger eines Signals ist umgangssprachlich definiert als der Bereich auf der Zeitachse, auf der das Signal *lebt*. Formal ist der Träger

$$\text{supp } f = \{t \in \mathbb{R}: f(t) \neq 0\} \subset \mathbb{R}$$

In einem realen Anwendungsfall wir ein Signal immer endlichen Träger  $[a, b] \subset \mathbb{R}$  haben, also gilt für ein passendes Maß  $d\mu$ , dass  $d\mu([a, b]) < \infty$  ist.

---

<sup>3</sup>Was ja auch durchaus realitätsnah ist.

## 5 Transformationen

### 5.1 Hough-Transformation

**Was ist die Hough-Transformation?** Die Hough-Transformation ist ein allgemeines Verfahren, um in einem **binarisierten** Bild Formen zu erkennen.

**Wie funktioniert sie?** Bei der Hough-Transformation wird für alle möglichen Variationen einer Form im gesamten Bild gesucht, ob diese vorkommt. Dies wird dadurch realisiert, dass man bei der Test-Form alle Pixel gezählt werden, bei der in der binarisierten Form ein Farbwert (z.B. 1 für Schwarz) steht. Hat man nun eine hohe Anzahl gefunden, so hat man ziemlich sicher auch die gesuchte Form im Bild erkannt.

**Was ermöglicht sie?** Mit diesem Verfahren kann man nun nach *allen möglichen* Formen in Bildern suchen, die Formen müssen sich entsprechen parameterisieren lassen. Dies ist u.a. für Linien, Kreise, Ellipsen u.v.m. möglich.

Man beachte aber, dass man z.B. bei Linien zwei Freiheitsgrade hat. D.h. man muss für alle Möglichkeiten dieser Freiheitsgrade das Bild absuchen, was hier schon quadratischen Aufwand liefert. Will man nun z.B. Kreise finden, so hat man schon drei Freiheitsgrade (die  $x$ - und die  $y$ -Koordinate des Mittelpunktes und den Radius  $r$ ), was entsprechend kubischen Aufwand liefert.

**Was sind mögliche Optimierungsmöglichkeiten?** Mögliche einfache Optimierungen sind z.B., dass man die Freiheitsgrade einschränkt. Sucht man z.B. in einem Bild nur nach Kreisen mit einem festen Radius, so besteht die einzige Variationsmöglichkeit in den Koordinaten des Kreismittelpunktes, was insgesamt also quadratischen Aufwand verursacht.

**Welcher Filter spielt dabei eine Rolle?** Man muss das Bild vorher z.B. durch einen entsprechenden Filter binarisieren.

**Gibt es weitere Filter zur Kantenerkennung?** Der wohl einfachste (und schnellste) Filter zur Kantenerkennung ist ein Bildshift. Damit verschiebt man das Bild in die Richtung, in der man Kanten finden möchte um eine Position. Ist der Unterschied hoch, so unterscheiden sich die Farbwerte signifikant, also liegt vmtl eine Kante vor.

Eine andere Möglichkeit ist die Betrachtung eines Differenzbildes.

Eine nette Alternative, die sich massiv parallelisieren lässt, ist der Sobel-Operator. Dieser Filter nimmt die Farbwerte benachbarter Pixel gewichtet und errechnet daraus einen neuen Wert<sup>4</sup> und findet so Kanten.

---

<sup>4</sup>Stencil und so.

Eine weitere, jedoch ziemlich unbrauchbare Möglichkeit ist die Verwendung der Fouriertransformation. Dabei erhält man zwar die Information, dass es Kanten im Bild gibt, aber nicht wo, da man keine Lokalitätsinformationen hat<sup>5</sup>.

Eine weitere, moderne Alternative ist die Verwendung von Wavelets. Damit erkennt man Kanten durch die schnellen Änderungen von Frequenzen. Zusätzlich weiß man auch, wo sich diese Kanten befinden, da man bei der Wavelettransformation ja Lokalitätsinformationen hat.

## 5.2 Gabor-Transformation

**Was ist die Gabor-Transformation?** Die Gabor-Transformation ist eine *gefensterte* Fouriertransformation.

Formal gilt also für  $f, g \in L_2(\mathbb{R})$ ,  $g$  gerade und  $\|g\|_2 = 1$ :

$$Gf(u, \xi) = \int_{\mathbb{R}} f(t) e^{-i\xi t} g(t-u) dt$$

**Was ist ein Spektrogramm?** Ein Spektrogramm ist ein Diagramm, auf dem die  $x$ -Achse die Zeit und auf der  $y$ -Achse die Frequenz aufgetragen ist. Das spiegelt also wider, wieviel Energie sich zu einer Zeit in einer gewissen Frequenz konzentriert.

**Wie hängt die Gabor-Transformation mit der Fouriertransformation zusammen?** Durch die Fensterfunktion bekommt man eine gewisse Lokalitätseigenschaft, da man dann Aussagen über die Frequenzanteile in dem spezifizierten Zeitfenster treffen kann, während sich die reine Fouriertransformation diese Informationen global ansieht.

## 5.3 Wavelets

**Was ist ein Wavelet?** Ein Wavelet  $\Psi$  ist eine beliebige  $L_2$ -Funktion, die die Eigenschaft der Mittelwertfreiheit erfüllt, also dass gilt

$$\int_{\mathbb{R}} \Psi(t) dt = 0.$$

Der Name kommt daher, dass diese Funktionen wie Wellen aussehen müssen (also Anteile über und unter der  $x$ -Achse), um in der Fläche 0 zu sein.

**Was bedeutet Zulässigkeit bei Wavelets?** Die Zulässigkeit bei Wavelets ist eine mathematische Bedingung, dass

$$C_\Psi := \int_{\mathbb{R}} \frac{|\hat{\Psi}(\xi)|^2}{|\xi|} d\xi < \infty$$

---

<sup>5</sup>Das ist ein Statement vom Matheman, das ich für ziemlich unbrauchbar halte, aber der Vollständigkeit halber sei es hier erwähnt.

gelten muss. Um dies zu gewährleisten sollte die Funktion an der 0 insbesondere 0 sein, weil man selten gut durch Null dividieren kann.

Die Zulässigkeit ist insbesondere eine Voraussetzung dafür, dass man die Inverse Wavelet-Transformation berechnen kann, weil sonst der Term  $\frac{1}{C_\psi}$  null wäre und dann würde man als Rekonstruktion die konstante Nullfunktion erhalten.

**Was ist die Wavelet-Transformation?** Die Wavelet-Transformation ist eine integraltransformation, welche das Signal und das Wavelet quasi als Fensterfunktion miteinander in Beziehung setzt:

$$\mathcal{W}_\Psi f(u, s) := \int_{\mathbb{R}} f(t) \frac{1}{\sqrt{|s|}} \overline{\Psi\left(\frac{t-u}{s}\right)} dt$$

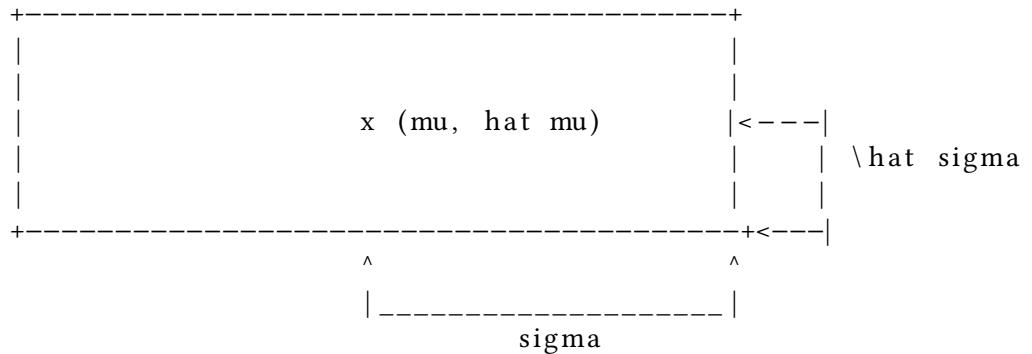
**Was ist ein Skalogramm?** Ein Skalogramm ist das Analogon zum Sprektrogramm der Gabor-Transformation, mit dem Unterschied, dass nun auf der vertikalen Achse nicht die Frequenzen sondern die *Skalen* aufgetragen sind, was in etwa dem Kehrwert der Frequenzen entspricht. Die (Farb-)Werte im Diagramm selbst wird dann die Intensität da sein.

**Was sind Heisenberg-Boxen?** Die Heisenberg-Boxen sind die graphischen Elemente des Skalogramms. Diese werden durch insgesamt vier Parameter definiert:

1. Zeitauflösung  $\mu$
2. Frequenzauflösung  $\hat{\mu}$
3. Zeitvarianz  $\sigma$
4. Frequenzvarianz  $\hat{\sigma}$

Die Zeit- und die Frequenzauflösung bilden zusammen den Mittelpunkt der Box im Diagramm, während die Varianzen die halbe Ausdehnung der Boxen in die jeweiligen Achsenrichtungen darstellen.

Graphisch (oder semi-graphisch) dargestellt:



**Was besagt die Heisenberg'sche Unschärferelation?** Die Heisenberg'sche Unschärferelation besagt, dass diese Heisenberg-Boxe eine Mindestfläche besitzen. Das bedeutet insbesondere, dass man nicht beliebig genau in Zeit und Frequenz auflösen kann. Dies ist aber auch der Natur nachempfunden, da z.B. ein Ton mit einer niedrigen Frequenz relativ lange schwingen muss, um gehört zu werden.

**Warum findet sich bei vielen Wavelets der Gauß-Kern als Term wieder?** Dies liegt daran, dass man mit dem reinen Gauß-Kern prinzipiell die bestmögliche Zeit-/Frequenz-Auflösung bekommen würde (also hat die Heisenberg-Box dann minimale Fläche). Das Problem ist nur, dass der Gaußkern kein Wavelet ist, da nicht mittelwertfrei.

**Was ist der Vorteil der Wavelet-Transformation gegenüber der Fourier- bzw Gabor-Transformation?**

Der klare Vorteil hierbei liegt darin, dass sich die Heisenberg-Boxen (und damit die Auflösung) dynamisch an die Frequenz oder Zeit anpasst. Bei Gabor legt man durch die Fensterfunktion aber eine fixe Frequenz-/Zeit-Auflösung fest. Man bekommt mit der Wavelet-Transformation also eine dynamische Zeit-/Frequenz-Auflösung mit einer gewissen Lokalitätseigenschaft. Bei der Wavelet-Transformation gilt: Je »unglatter« die Funktion ist, desto höher ist die Abklingrate der Wavelet-Koeffizienten.

**Wo spiegelt sich bei der Wavelet-Transformation Redundanz wider?** Man kann<sup>6</sup> mathematisch nachweisen, dass man die Originalfunktion aus der Wavelet-Transformation mit nur einer einzigen Skala rekonstruieren kann.

**Gibt es praktische Anwendungen für die Wavelets?** Ein prominentes Beispiel<sup>7</sup> ist z.B. Bildkompression mit JPEG-2000.

---

<sup>6</sup>Man kann, ich kann es nicht ...

<sup>7</sup>Ein prominentes Beispiel, was dennoch kein Schwein hennimmt (facepalm).