Einführung in die Signal- und Bildverarbeitung

Vorlesung, zuerst gehalten im Sommersemester 2012

Tomas Sauer

Version 1.0pre
Letzte Änderung: 17.7.2012

Statt einer Leerseite ... 0

Chaos is found in greatest abundance whereever order is being sought. It always defeats order, because it is better organized.

T. Pratchett, *Interesting times*

Die wahren Analphabeten sind schließlich diejenigen, die zwar lesen können, es aber nicht tun. Weil sie gerade fernsehen.

L. Volkert, *SZ–Online, 11.7.2009*

When the epoch of analogue (which was to say also the richness of language, of *analogy*) was giving way to the digital era, the final victory of the numerate over the literate.

S. Rushdie, *Fury*

The most incredible thing about miracles is that they happen.

G. K. Chesterton, *The Innocence of Father Brown*

And it didn't stop being magic just because you found out how it was done.

T. Pratchett, *Wee Free Men*

Tomas Sauer
Lehrstuhl für Mathematik mit Schwerpunkt Digitale Bildverabeitung
Universität Passau
Innstr. 43
94032 Passau

$f(x)$
$f(x)$
$f\,(x)$
$f\,(x)$
$f\,(x)$

0 Inhaltsverzeichnis

Ein Image ist das, was man bräucht',
dass die anderen denken, dass man so
ist, wie man gern wär.

E. Pelzig

1 Bilderfassung

Die digitale Bildverarbeitung befasst sich – wenig verwunderlich – mit der Verarbeitung digitaler Bilder. Allerdings ist der englische Name, *Digital Image Processing*, insofern aussagekräftiger, als sich hinter „Image" nicht nur Bilder im klassischen Sinne sondern auch andere Strukturen verbergen können, was die Sache deutlich allgemeiner und interessanter macht. Klassische Aufgaben der Bildverarbeitung lassen sich in den folgenden Kategorien zusammenfassen[1]:

Image Enhancement: Das Bild ist aus welchen Gründen auch immer nicht von der Qualität, die man eigentlich gerne hätte, seien es Defekte in der Aufnahmeapparatur oder zufällig oder bewusst schlecht gewählte Aufnahmeumstände[2]. Solche „Störungen" können Rauschen, partielle Über- oder Unterbelichtung, schlechter Kontrast oder auch ganz andere Artefakte sein, und sollen nun mit geeigneten Verfahren nach Möglichkeit reduziert oder beseitigt werden.

Image Restoration: Teile des Bildes fehlen und sollen aus den vorhandenen Daten ergänzt werden.

Image Compression: Wie speichert man die Unmengen von Daten, die heutzutage erhoben werden effizient und ohne allzu großen Qualitätsverlust. Und was bitte ist Qualität?

Information Extraction: Unser „Bild" enthält Information, die ein Programm automatisch extrahieren soll. Das kann Computersehen genauso bedeuten wie das automatische Auffinden von Anomalien in Computertomographien.

Daß die Behandlung dieser Fragestellungen eine ganze Menge Mathematik benötigt, ist ziemlich naheliegend. Und da die zu verwendenden Verfahren oftmals von der Natur der „Bilder" abhängt, diese aber wieder sehr stark von

[1] Die anglophile Wortwahl ist kein Anfall von Denglisch, sondern es sind gebräuchlich Termini Technici, die man auch wirklich nur schwer eindeutschen kann, weil es keine ähnlich standardisierten Übersetzungen gibt.

[2] Manche Fotographen lieben Gegenlicht und gerade bei Röntgenuntersuchungen wird oft die Bildqualität zugunsten einer reduzierten Strahlenbelastung heruntergefahren.

ihrer Erfassung, ist es vielleicht ganz interessant, sich zuerst einmal ein paar „Bildtypen“ anzusehen.

1.1 Bildmodelle

Bei „Bilder“ denkt man natürlich intuitiv zuerst an Fotographien[3], also Aufnahmen mit einer Kamera. Für die **digitale** Bildverarbeitung müssen diese Aufnahmen natürlich erst einmal digitalisiert werden, aber dennoch bestimmt genau diese Vorstellung auch ganz massiv unser mathematisches Modell eines Bildes, nämlich als zweidimensionales, rechteckiges Objekt. Und da wird es dann auch schon interessant, denn je nach Wunsch und Anwendung können wir Bilder unterschiedlich modellieren:

Realistisch: Ein Bild ist ein **Array**, dessen Elemente als **Pixel** bezeichnet werden und in denen *diskrete* Farbwerte codiert sind:

$$B = \left[b_{jk} \; : \; \begin{matrix} j = 1, \ldots, m \\ k = 1, \ldots, n \end{matrix} \right], \qquad b_{jk} \in \{0, \ldots, M\}.$$

Wie genau die Farben codiert sind, ob über eine Farbtabelle oder über diskrete RGB–Werte, spielt hier erst einmal keine Rolle.

Halbkontinuierlich: Wir betrachten das Bild lediglich als Array von *unquantisierten* reellwertigen Einträgen, die Grauwerte oder Farbanteile in einem der Standardfarbsysteme wie RGB oder YUV, siehe (Foley *et al.*, 1990), in stufenloser Form angeben:

$$B = \left[b_{jk} \; : \; \begin{matrix} j = 1, \ldots, m \\ k = 1, \ldots, n \end{matrix} \right], \qquad b_{jk} \in \mathbb{R} \text{ oder } b_{jk} \in \mathbb{R}^3.$$

Der Übergang von diesem Modell zum volldiskreten erfolgt durch **Quantisierung** der Werte, also eine Aufteilung auf endlich viele diskrete Werte.

Vollkontinuierlich: Manchmal ist es auch hilfreich, sich ein Bild als eine Funktion von einem Rechteck nach $\mathbb{R}$ oder nach $\mathbb{R}^3$ vorzustellen, $B : [0, m] \times [0, n] \to \mathbb{R}^3$, wenn die Pixel also sozusagen so dicht geworden sind, daß sie ein Kontinuum bilden. Jetzt kann man Methoden aus der Analysis anwenden, was gerade bei der Kantenbestimmung in Bildern die Grundlage vieler leistungsfähiger moderner Verfahren geworden ist[4]. Das heisst aber auch, daß man bei der numerischen Rechnung oder der Anwendung auf endliche diskrete Daten erst einmal wieder diskretisieren muss, was auch nicht immer so ganz einfach ist.

[3] Wörtlich „Lichtzeichnerei“, eines der vielen griechischen Wörter, die es im Griechischen nie gab und deren Transkription immer zu einer gewissen Ratlosigkeit führt, ob nun „f“ oder „ph“ zu verwenden ist. Die Schreibweise „Fotographie“ erscheint in diesem Zusammenhang etwas inkonsistent (und ist es wohl auch), aber „Photographien“ erschien mir zu eklektisch und „Fotografien“ beissen sich mit der „Tomographie“, „Tomografie“ sieht aber wieder seltsam aus. Gut jedenfalls, daß wir darüber geredet haben.

[4] Wir kommen noch dazu.

Obwohl eigentlich nur das diskrete Modell wirklich realistisch ist, gibt es doch auch gute Gründe, die anderen Modelle zu betrachten, einfach deswegen, weil das Weglassen von „Realität“ den Einsatz leistungsfähigerer mathematischer Modelle erlaubt. Daher die folgende Definition, die in ihrer Allgemeinheit vollkommen nichtssagend ist, aber eben auch schon zeigt, daß Bildverarbeitung nicht unbedingt gleich Bildverarbeitung sein muss.

Definition 1.1 (Bild) *Ein **Bild*** B *ist eine Abbildung von einem Definitionsbereich* D *nach* E^d *für ein passendes* d.

Ein andere, mindestens ebenso wichtige Frage ist, wie das Bild eigentlich erzeugt wurde. Bei einem Digitalfoto werden einfallende Lichtstrahlen von der Optik transformiert und dann auf einem Chip erfasst, bei Röntgenbildern werden unterschiedliche Absorptionsraten aufgezeichnet, in der Tomographie werden eigentlich erst einmal ganz andere Dinge gemessen und so weiter. Das sorgt dafür, daß Bild nicht gleich Bild ist, selbst wenn am Ende vielleicht immer ein Raster von 512×512 Farbwerten herauskommt, die RGB mit jeweils 8 Bit quantisieren. Denn:

> Die für ein Bild zu verwenden Verfahren hängen von den Methoden der Bilderfassung und dem zugrundeliegenden Bildmodell ab.

1.2 Fotographie

Die allereinfachste Vorstellung einer „fotographischen“ Bilderzeugung wäre die einer **Parallelprojektion** auf die Bildebene, wobei wir es entweder mit von einem Objekt ausgesandten bzw. reflektierten oder durch das Objekt hindurchgehenden Strahlen zu tun hätten, siehe Abb 1.1. Die Parallelprojektion hat einen großen Vorteil: Es gibt keinerlei optische Verzerrungen und der Abbildungsmaßstab ist überall gleich, weswegen man Fernröntgenbilder einfach ausmessen kann. Daher werden diese gerne in der Kieferorthopädie oder bei der Planung orthodontischer Eingriffe[5] genutzt. In der Optik macht man auch gerne mal die Annahme des parallelen Lichteinfalls, hat es dann aber formal mit unendlich weit entfernten Objekten zu tun[6].

Ein etwas realistischeres Bild der Fotographie ergibt sich mit dem einfachen Modell der **Lochkamera**, die mathematisch auf dem Konzept der **Zentralprojektion** basiert. Legt man wie in Abb 1.2 den Nullpunkt in die **Lochblende**, so erhält man als Bild des Punktes $\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ den **Bildpunkt** $\mathbf{x}' = \frac{d}{z}\begin{bmatrix} -x \\ -y \end{bmatrix}$ in der

[5] Zur Ausrichtung der Zahnstellung und für eventuelle ästhetische Korrekturen verwendet. Man kann sich vorstellen, daß eine Ganzkörper–Fernröntgenbild einen großen technischen Aufwand darstellen würde.

[6] Wobei in vielen Anwendungen die Unendlichkeit sehr viel früher beginnt, als man so landläufig erwarten würde.

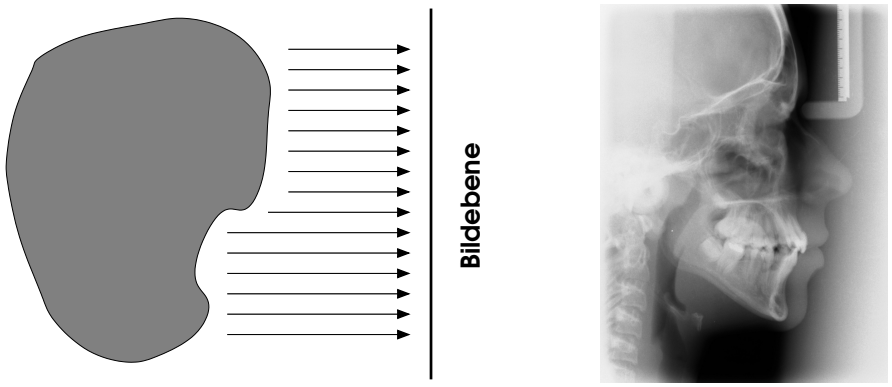


Abbildung 1.1: Parallelprojektion auf die Bildebene *(links)*. Derartige Bilderfassungsmethoden gibt es tatsächlich, nämlich beim **Fernröntgenbild** *(rechts, Quelle: `Wikimedia commons`)*, wo die Strahlenquelle so groß ist wie das spätere Bild.

Bildebene. Da es sich hierbei um projektive Geometrie handelt, verwendet man geschickter **homogene Koordinaten**, bei der ein Punkt $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3$ mit der Menge *aller* Punkte[7]

$$\widehat{\mathbf{x}} = \begin{bmatrix} w\,x_1 \\ w\,x_2 \\ w\,x_3 \\ w \end{bmatrix} \in \mathbb{R}^4, \qquad w \neq 0,$$

identifiziert wird, wobei $w = 1$ die **kanonische Einbettung** darstellt. Den Fall $w = 0$ schliesst man tunlichst aus, denn dieser unendlich ferne Punkt liegt in allen Äquivalenzklassen zu allen Punkten aus dem $\mathbb{R}^3$. Umgekehrt ordnet man einem Punk $\widehat{\mathbf{x}} \in \mathbb{R}^4$ dann wieder den Punkt

$$\widehat{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \mapsto \begin{bmatrix} x_1/x_4 \\ x_2/x_4 \\ x_3/x_4 \end{bmatrix} \in \mathbb{R}^3$$

zu. Das schöne daran ist, daß sich alle geometrischen Transformationen in homogenen Koordinaten als Matrixmultiplikationen darstellen lassen:

Translation um $\mathbf{y} \in \mathbb{R}^3$:

$$T_{\mathbf{y}}\widehat{\mathbf{x}} := \begin{bmatrix} \mathbf{I} & \mathbf{y} \\ \mathbf{0}^T & 1 \end{bmatrix} \widehat{\mathbf{x}}.$$

[7] Formal spricht man dann von einer **Äquivalenzklasse**.

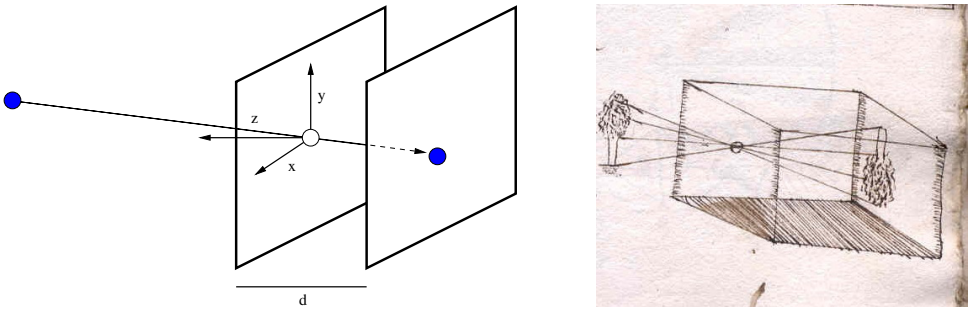


Abbildung 1.2: Zentralprojektion in einer (idealen) Lochkamera, dem Prinzip der Camera Obscura *(rechts eine Schemazeichnung aus dem 17. Jhdt., Quelle: `Wikipedia`)* Der Punkt mit den Koordinaten $\mathbf{x}$ wird nach den Regeln des Strahlensatzes auf einen Punkt $\mathbf{x}'$ in der Bildebene abgebildet. Dabei ist d der Abstand von der Lochblende zur Bildebene.

Lineare Abbildung [8] mit Matrix $\mathbf{A} \in \mathbb{R}^{3\times3}$:

$$L_{\mathbf{A}}\widehat{\mathbf{x}} := \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \widehat{\mathbf{x}}.$$

Projektion mit Projektionsfaktor $d \neq 0$:

$$P_d\widehat{\mathbf{x}} := \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -d^{-1}\,\mathbf{e}_3^\top & 0 \end{bmatrix}.$$

Übung 1.1 Gilt $L_{\mathbf{A}}T_y = T_y L_{\mathbf{A}}$? Wenn nein, wie lautet die korrekte Formel? ◊

Das mit der Projektion sehen wir uns etwas genauer an:

$$\begin{aligned} P_d\widehat{\mathbf{x}} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} wx_1 \\ wx_2 \\ wx_3 \\ w \end{bmatrix} = w \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ -\frac{x_3}{d} \end{bmatrix} = w\,\frac{x_3}{d} \begin{bmatrix} -\frac{x_3}{d}\,x_1 \\ -\frac{x_3}{d}\,x_2 \\ -\frac{1}{d} \\ 1 \end{bmatrix} \\ &\simeq \frac{d}{x_3} \begin{bmatrix} -\frac{x_3}{d}\,x_1 \\ -\frac{x_3}{d}\,x_2 \\ -\frac{1}{d} \end{bmatrix}, \end{aligned}$$

was nichts anderes als der Bildpunkt von $\mathbf{x}$ auf der Bildebene $x_3 = -1/d$ ist. Alle geometrischen Operationen und die Projektion in der Lochkamera lassen sich durch Multiplikationen mit 4×4–Matrizen realisieren[9], was auch der Grund ist, warum moderne Grafikkarten Hardware zur Durchführung dieser Matrizenmultiplikationen integriert haben. Auf der Basis von Brechungsgesetzen lässt sich so auch noch eine Optik, also das Verhalten einer oder meherer Linsen integrieren, was auch heute noch in der medizinischen Optik zur Planung von implantierbaren Kunstlinsen genutzt wird, (Langenbucher *et al.*, 2004).

[8] Dies beinhaltet Rotationen, Skalierungen und vieles anderes mehr.
[9] Dasselbe gilt auch in der Kinematik und Robotik, siehe (Paul, 1981).

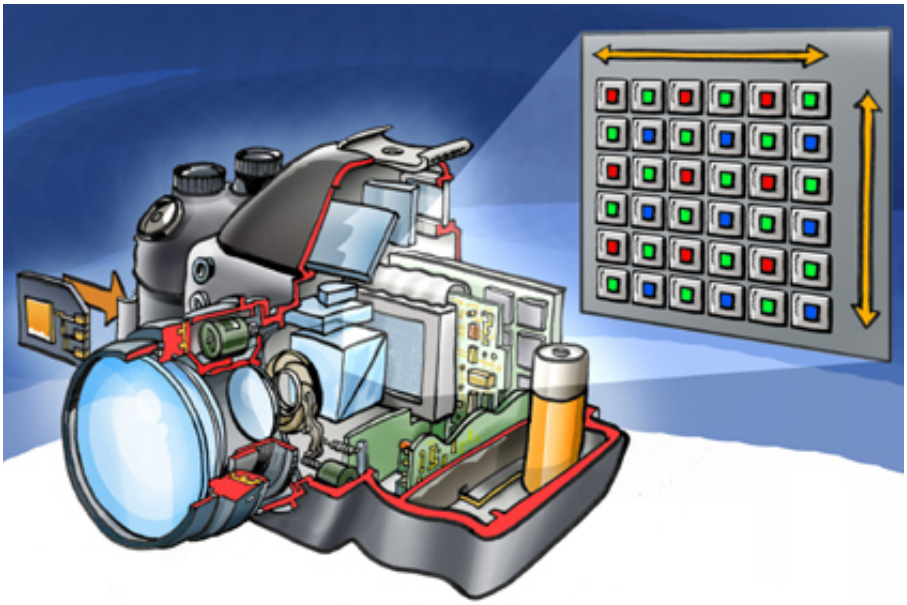


Abbildung 1.3: Digitalkamera mit Vergrößerung des Chip.
Quelle: `Peter Welleman @Wikimedia commons`

Das ist aber noch bei weitem nicht das Ende der Geschichte, denn die Lochkamera stellt nur ein sehr unvollkommenes Modell eine Kamera dar. Als nächstes müssten nämlich auch noch die Optik der Kamera, siehe Abb. 1.3, und die dadurch erzeugten Verzerrungen des Bildes modelliert werden. Diese Verzerrungen sind umso größer, je kürzer die Brennweite des Objektivs ist, siehe Abb. 1.4, bei Teleobjektiven mit langer Brennweite ist das Bild wesentlich homogener. Weitere Details zur Bilderfassung mit Kameras finden sich beispielsweise in (Jähne, 2002).

Wenn man in Abb. 1.3 genau hinsieht, erkennt man ein weiteres witziges Detail von Digitalkameras: Jedes Pixel des Sensors unterteilt sich in vier Subpixel, die den roten, blauen und *zweimal* den grünen Farbanteil aufnehmen. Die stärkere Gewichtung von Grün ist physiologisch bedingt, denn das menschliche „Auge“ ist im Grünbereich wesentlich sensibler.

1.3 Computertomographie

Eine sogenannte **Integraltransformation** mit kaum zu unterschätzender Bedeutung von der medizinischen Bildverarbeitung bis hin zur Materialprüfung ist die **Radontransformation**, die im Zusammenhang mit der **Computertomographie** zu Ruhm und Ehre gekommen ist. Wird ein (Röntgen)–Strahl durch ein inhomogenes Material geschickt, so wird ein Teil des Strahls vom Material absorbiert, der Rest setzt seinen Weg durch das Material fort, Brechung, Beugung

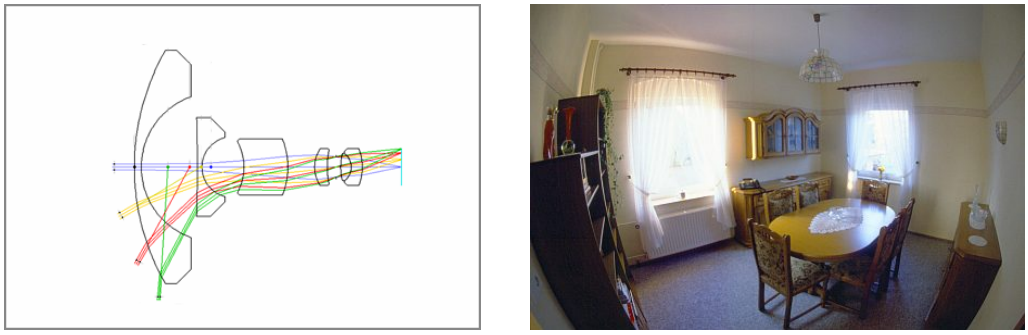


Abbildung 1.4: Querschnitt durch ein heute übliches Kamera–Objektiv *(links, Quelle: `Jos. Schneider Optische Werke GmbH`)* und der berühmte Fischaugen–Effekt bei extremen Weitwinkelobjektiven *(rechts, Quelle: `Wikimedia Commons`)*

und ähnliche Effekte wollen wir hier erst einmal ignorieren[10]. Bezeichnen wir diese Absorptionsrate[11] mit $f(x)$, $x \in \mathbb{R}^2$, und mit $I(x)$ die Intensität des Strahls an dieser Stelle, dann gilt für zwei Punkte x, $x + \delta$ auf dem Strahl laut (Olafsson & Quinto, 2006) näherungsweise[12] die Beziehung

$$I(x+\delta) - I(x) \approx -f(x)\,\delta\,I(x), \qquad \text{d.h.} \qquad I(x+\delta) \approx I(x)\,(1 - f(x)\,\delta)\,.$$

Um aus dieser multiplikativen Beziehung eine additive zu machen, logarithmieren wir beide Seiten, so daß wir mit ein klein bisschen Rechnerei[13]

$$\begin{aligned} \log I(x+\delta) &= \log I(x) + \log\,(1 - f(x)\,\delta) = \log I(x) - f(x)\,\delta + O\left(\delta^2\right) \\ &\approx \log I(x) - f(x)\,\delta \end{aligned}$$

erhalten – schließlich ist die Taylorentwicklung von $\log(1 - ax)$ an $x = 0$ ja

$$\log(1-ax) = -\sum_{j=1}^{\infty} \frac{a^j}{(1-ay)}\bigg|_{y=0} x^j = -\sum_{j=1}^{\infty} (ax)^j\,.$$

Zerlegen wir die Strecke von der Quelle x_S zum Detektor x_D in $N+1$ Stückchen der Länge δ, dann ist

$$\begin{aligned} \log \frac{I\,(x_S)}{I\,(x_D)} &= -\left(\log I\,(x_D) - \log I\,(x_S)\right) \\ &= -\sum_{j=0}^{N} \log I\,(x_S + (j+1)\delta) - \log I\,(x_S + j\delta) \approx \sum_{j=0}^{n} f\,(x_S + j\delta)\;\delta, \end{aligned}$$

[10] Das kann man dadurch rechtfertigen, daß man annimmt, daß der Strahl hinreichend "hart", also energiereich ist. Ob das immer den Patienten freut, sei an dieser Stelle einmal dahingestellt.

[11] Die zumindest bei monochromatischen Röntgenstrahlen proportional zur Dichte ist, siehe (Olafsson & Quinto, 2006, S. 2).

[12] Das ist Physik – da ist zwischen näherungsweise und exakt eigentlich kein wirklicher Unterschied.

[13] Taylor an der Stelle 0 nach δ und das ist klein!

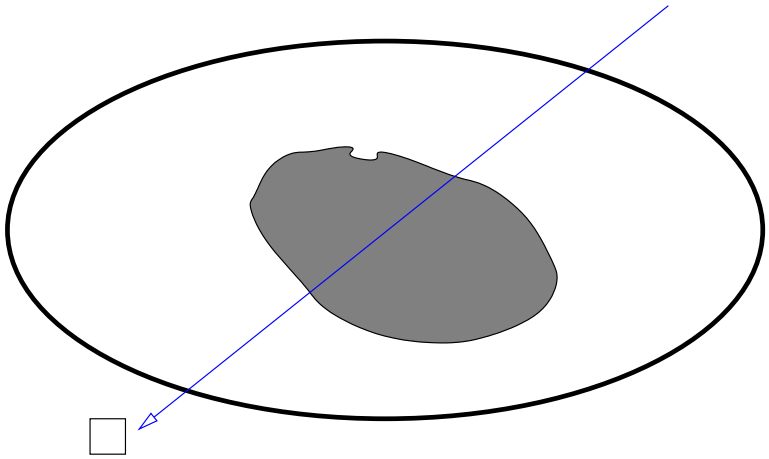


Abbildung 1.5: Ein Strahl wird durch ein Objekt geschickt und auf die Intensität auf der anderen Seite durch einen Detektor gemessen. Die Absorption durch das Material ist ein Integral entlang dieser Linie.

was nun wieder nichts anderes als eine **Quadraturformel** für das **Linienintegral**

$$\int_{[x_S,x_D]} f(x)\,dx := \|x_D - x_S\| \int_0^1 f\left(\lambda x_S + (1-\lambda)x_D\right)\,d\lambda$$

ist, wobei die Normalisierung so gewählt wurde, daß $\int_{[x_S,x_D]} 1\,dx = \|x_D - x_S\|$ die Länge des Streckenzugs reproduziert. Damit sind wir auch schon fast bei der Radontransformation, die 1917 von Radon zu rein mathematischen Zwecken eingeführt wurde und die zu einer Funktion f alle möglichen Linienintegrale berechnet. Dazu müssen wir die Linien in $\mathbb{R}^2$ nur noch parametrisieren, was wir beispielsweise mit einem Richtungsvektor y, $\|y\| = 1$, und dem vorzeichenbehafteten Abstand s der Geraden zum Nullpunkt tun können. Die Menge $\mathscr{L}$ aller Geraden in $\mathbb{R}^2$ kann man somit mit $\mathbb{R}\times\mathbb{S}^2$ identifizieren, wobei $\mathbb{S}^2 = \{y \,:\, \|y\| = 1\}$ die zweidimensionale Einheitskugel, auch als Einheitskreis bekannt, ist.

Übung 1.2 Wann schneiden sich zwei Geraden $L, L' \in \mathscr{L}$? ◊

Dann ist das zugehörige Lininienintegral zu dieser Linie $L = (s, y) \in \mathscr{L}$ definiert als

$$\int_L f(x)\,dx := \int_{\mathbb{R}} f\left(sy^\perp + ty\right)\,dt, \qquad y^\perp = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{1.1}$$

Übung 1.3 Zeigen Sie, daß die Gerade L durch den Punkt $s\,y^\perp$ geht. ◊

Definition 1.2 *Die **Radontransformation** R ordnet einer Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ die Linienintegrale zu:*

$$Rf(L) := \int_L f(x)\,dx.$$

$Rf : \mathscr{L} \to \mathbb{R}$ *ist also eine Funktion von der Menge aller Geraden in $\mathbb{R}^2$ nach $\mathbb{R}$.*

Man sieht also: Das Ergebnis der Radontransformation hat einen recht obskuren Definitionsbereich, nämlich $\mathscr{L}$, was für Mediziner erst einmal unbrauchbar ist, wie man in Abb. 1.6 auch sehr schön sehen kann. Zu praktischen Zwecken

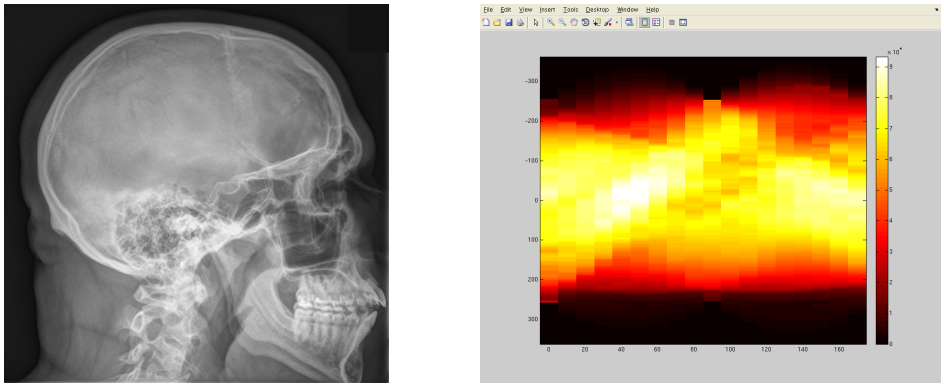


Abbildung 1.6: Via `Matlab` „simulierte“ Radontransformation eines Längsschnitts durch einen Schädel *(links, Quelle: Wikimedia Commons*. In der x–Achse sind die Winkel aufgetragen, in der y–Achse die Verschiebungen.

müssen wir die Linienintegrale erst einmal wieder in ein zweidimensionales Bild umwandeln, das uns vor das folgende mathematische Problem stellt:

Berechne die Inverse der Radontransformation einer Funktion

Dieses Problem mathematisch–theoretisch und dann auch numerisch angehen zu können[14] benötigt aber einiges an Theorie. Nur so viel im Moment: Die gute Nachricht ist, daß die Radontransformation invertierbar ist, die schlechte, daß die normalerweise “gemessenen” Linienintegrale dafür nicht ausreichen. Daß alles dennoch funktioniert, liegt an geeignet gewählten numerischen Verfahren und der Tatsache, daß man in der medizinischen Bildverarbeitung normalerweise nicht die exakten Dichtewerte berechnen muss, sondern, daß es sehr viel wichtiger ist, *Dichteveränderungen* zu bestimmen, denn diese markieren den Übergang zwischen verschiedenen Gewebetypen. Nicht die Farbe zählt, der Kontrast ist wichtig.

Bemerkung 1.3 (CT–Rekonstruktion)

1. *Die Rekonstruktion der Dichteverhältnisse bzw. -veränderungen aus unvollstxändiger Information ist so alt wie die Computertomographie und es gibt hierfür eine Vielzahl von Verfahren, die in der aktuellen Praxis zumeist auf einem Verfahren beruhen, das als* ***gefilterte Rückprojektion*** *bekannt ist. Wir werden das im Kontext der Signalverarbeitung einmal kurz ansprechen.*

2. *Moderne Tomographen arbeiten in vielen Fällen nicht mehr schichtweise, das dauert zu lang, sondern umlaufen das Objekt auf einer Spiralbahn, siehe Abb 1.7.*

[14] Wir werden sehen, daß das alles nicht so einfach ist.

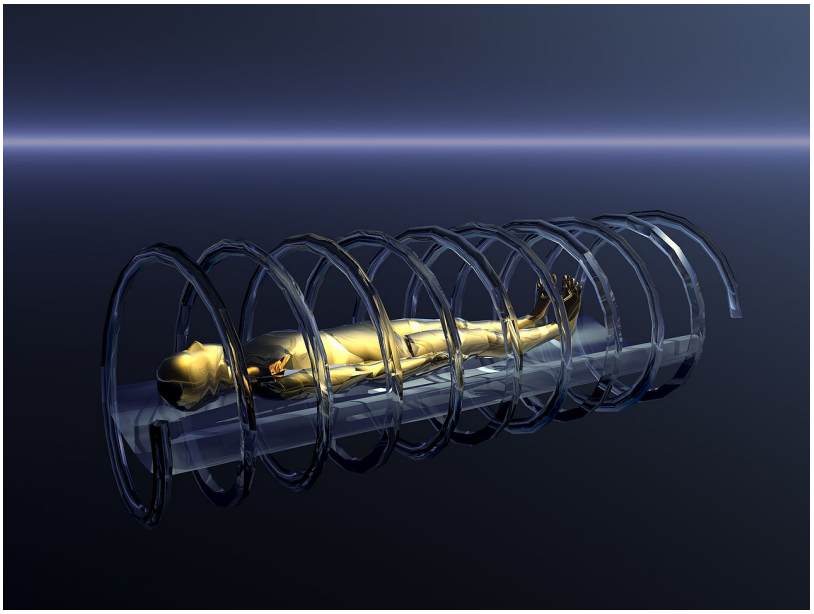


Abbildung 1.7: Lauf der Röhren im Spiral–CT.
Quelle: `Nevit Dilmen @Wikimedia commons`

*Damit hat man allerdings erst einmal auf **keiner** Schicht eine „richtige“ Radon–Transformation zur Verfügung und die Geschichte wird auch mathematisch aufwendiger.*

3. *So einfach wie in Abb 1.5 ist die Physik auch wieder nicht: Röntgenstrahlen gehen nicht einfach so glatt durch, sondern werden gebeugt, gestreut und teilweise auch ganz aufgehalten, beispielsweise von Metall. **Metallartefakte** sind dann auch heute noch ein ganz großes Problem, siehe Abb 1.8.*

1.4 Medical Imaging

Völlig andere „Bilder“ werden oftmals in der medizinischen Bildverarbeitung betrachtet. Ein Beispiel hier ist die **Elektroenzephalographie**, die das **EEG** liefert. Beim EEG werden Hirnaktivitäten über Elektroden abgeleitet, die aussen am Kopf angebracht sind. Genau genommen werden die Potentialdifferenzen zur einer *Referenzelektrode* gemessen. Aktivitäten in bestimmten Hirnbereich zeichnen sich dann durch starke Positivierung oder Negativierung des Signals an den Elektroden aus, die über dem entsprechenden Bereich liegen. Allerdings ist die Sache so klar auch wieder nicht, denn die Schädeldecke[15] funktioniert hier als ein **Tiefpassfilter** und streut die Hirnaktivitäten über den Kopf. Mathematisch gehören EEG–Messungen zu einer klassischen Familie von Signalen.

[15] Auf die die meisten Patienten und Versuchspersonen aus nicht immer ganz nachvollziehbaren und recht egoistischen Gründen bestehen.

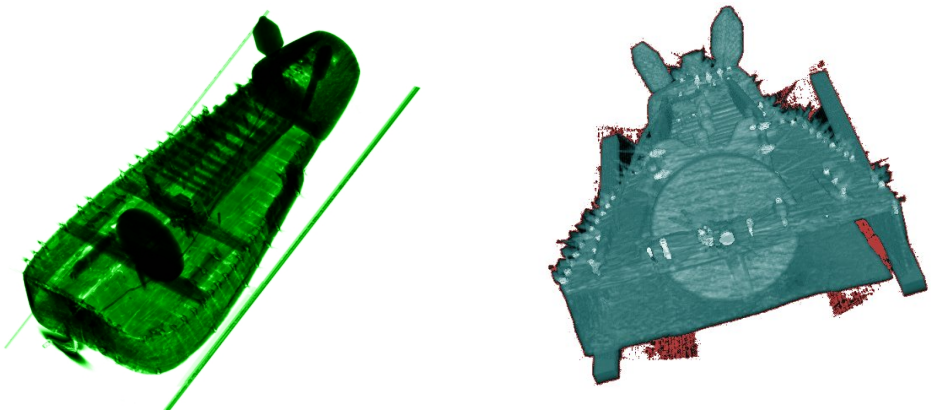


Abbildung 1.8: Scans eines Musikinstruments (Drehleier). Auf beiden Bildern kann man sehr gut die Metallartefakte sehen, links an der Kurbel (unten links) mit sehr massiven Streuungseffekten, rechts an den Nägeln, wo sternförmige Artefakte zu sehen sind. Die Dichtewerte dieser Artefakte entsprechen gemeinerweise recht genau der Dichte des Holzes.

Definition 1.4 *Eine **multivariate Zeitreihe** ist eine Abbildung* $\mathbf{f} : I \to \mathbb{R}^N$*, wobei* $I \subset \mathbb{Z}$ *oder* $I \subset \mathbb{R}$ *ein diskretes oder kontinuierliches Intervall ist.*

Reale EEG–Messungen sind ein zeitdiskretes, 30–50–dimensionales Signal mit einer Abstastfrequenz von oftmals ca. 1000Hz. Das EEG einfach als Vektor zu sehen wird ihm aber nicht gerecht, denn schließlich gibt es zwischen den Elektroden auch Nachbarschaftsbeziehungen. Deswegen kann man EEG–Signale auch als „richtige“ Bilder darstellt, indem man die Potentiale farbcodiert[16] an den entsprechenden Stellen aufträgt, wie es ebenfalls in Abb 1.9 zu sehen ist. Daraus kann man sogar eine Theorie machen, indem man die „wesentlichsten“ Zustände zu sogenannten **Microstates** zusammenfasst und so versucht, die wesentlichen Hirnzustände während eines Experiments zu katalogisieren, siehe Abb 1.10.

Aber es geht noch besser! Im Rahmen von **LORETA** versucht man, ausgehend von den Messungen auf der Schädeloberfläche das Anregungspotential **im** Gehirn zu rekonstruieren, siehe (Pascual-Marqui *et al.*, 1994). Dabei nimmt man eine Potentialverteilung $x : \mathbb{R}^3 \to \mathbb{R}$ im Gehirn an, die dann entsprechend physikalischer Gegebenheiten eine Potentialverteilung $F(x)$ auf der Schädeldecke liefern soll, wobei F eine halbwegs berechenbare Funktion ist. Und dann muss man nur noch diese Werte mit den Messwerten y zur Deckung bringen, also $F(x) = y$ lösen. Klingt zuerst einmal einfach, wird aber interessanter, wenn man bedenkt, daß die Messungen nur auf einer zweidimensionalen Fläche vorliegen, die Aktivität aber in einem dreidimensionalen Volumen vorliegt. Diskretisiert man das in jeder Dimension in N Blöcke, so hat man also N^2

[16] Dabei spielt es natürlich eine Rolle, **wie** man die Farbtabelle auswählt!

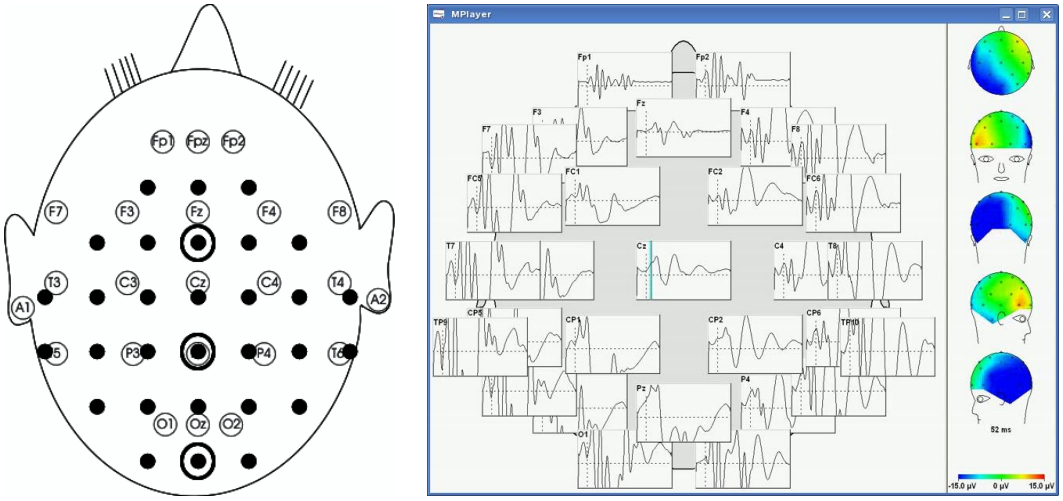


Abbildung 1.9: Lage und Name der Elektroden in einem Standard–EEG–System *(links)* und Darstellung eines Anregungspotentials (in diesem Fall ein cardioballistisches Artefakt) als Farbverlauf auf dem Kopf *(rechts)*.

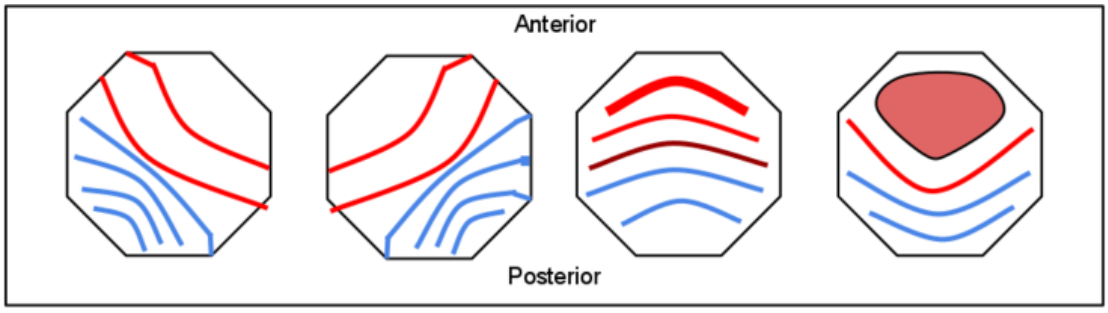


Abbildung 1.10: Verschiendene Microstates, also schematisierte Potentialbilder.

Gleichungen in N^3 Unbekannten und das Ganze wird immer unterbestimmter, je genauer man zu rechnen versucht.

1.5 Indirekte Messung, inverse Probleme

Wir haben es ja beim EEG bereits gesehen: Es gibt Probleme, bei denen man die zu untersuchenden Effekte nicht direkt messen kann, sondern nur indirekt als Ausgabe eines anderen Prozesses. Wir erhalten also nicht x sondern nur $F(x)$, wobei F normalerweise **nicht** surjektiv ist, das heißt, es gibt Punkte $x \neq x'$ mit $F(x) = F(x)$. Der einfachste Fall hiervon sind lineare Probleme, also *unterbestimmte* Systeme der Form

$$Ax = y, \qquad A \in \mathbb{R}^{m \times n}, \quad m < n. \tag{1.2}$$

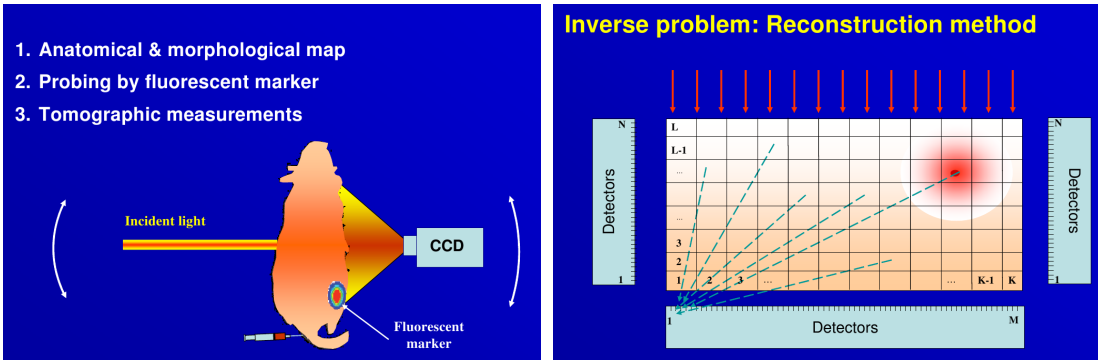


Abbildung 1.11: Optische Tomographie *(links)*. Ein fluoreszierender Marker *innerhalb* einens Lebewesens (hier eine Maus) wird durch Laserbestrahlung aktiviert und das Streulicht am anderen Ende aufgesammelt. Aus diesen Informationen soll die Lage des Markers rekonstruiert werden.

Im physikalischen Modell *(rechts)* wird der Lichtaustausch zwischen benachbarten Zellen durch entsprechende Beugungs-, Reflektions- und Absorptionsgesetze ermittelt, woraus sich der Lichteinfall auf den Detektoren ergibt.

Beispiel 1.5 (Optische Tomographie) *Die **optischen Tomographie** durchleuchtet Objekte, normalerweise Lebewesen*[17] *ausschließlich mit Hilfe von sichtbarem Licht, wobei im Inneren ein fluoreszierender Marker plaziert wird, siehe Abb. 1.11, der durch das einfallende (Laser-) Licht aktiviert wird. Es klingt ein wenig abwegig mit Hilfe von Licht durch eine Maus "hindurchschauen" zu wollen, aber tatsächlich kommt wirklich einiges an Licht auf der anderen Seite des Objekts an, allerdings durch die Moleküle*[18] *gestreut und gebrochen.*

Zur Modellierung dieses Problems teilt man jeden Schnitt durch das Objekt und seine Umgebung in kleine Quadrate ein und modelliert den Lichtaustausch zwischen diesen Quadraten[19] *sowie den Randquadraten und den Detektoren. Bezeichnet* $x \in \mathbb{R}^{MN}$ *die Lichtintesität an den Quadraten des* $M \times N$*–Gitters und* $y \in \mathbb{R}^{K}$ *die gemessenen Werte an den* K *Detektoren, die um das Objekt herum oder hinter dem Objekt angebracht sind, dann erhalten wir ein Gleichunssystem der Form*

$$y = Ax, \qquad A \in \mathbb{R}^{K \times MN}.$$

Ab einer bestimmten Rekonstruktionsgenauigkeit hat man dann aber mit Sicherheit wesentlich mehr Quadrate als Detektoren und das Problem wird massiv unterbestimmt!

Jetzt aber genug der Realität und zurück zur Theorie. Da unser unterbestimmtes Gleichungssytem $Ax = b$ aus (1.2) normalerweise jede Menge Lösungen hat, genauer gesagt, einen $n - m$–dimensionalen Lösungsraum, müssen wir uns aus

[17] Und in diesem Sinne eigentlich eher Subjekte als Objekte.
[18] Bekanntlicherweise bei Lebewesen zumeist Wasser.
[19] Der Einfachheit halber werden hier normalerweise Quadrate und deren Nachbarn betrachtet.

diesen Lösungen eine gute Lösung aussuchen – warum also nicht gleich die Beste? Dazu lösen wir das Optimierungsproblem

$$\min_x \gamma(x), \qquad Ax = b, \tag{1.3}$$

mit einem vorgegeben **Gütefunktional** $\gamma : \mathbb{R}^n \to \mathbb{R}$, das zumindest nach unten beschränkt sein sollte[20]; diese untere Schranke können wir dann auch gleich auf Null setzen und so $\gamma : \mathbb{R}^n \to \mathbb{R}_+$ annehmen. Beliebte Werte für γ sind $\gamma(x) = \|x\|_p$, $1 \le p \le \infty$, oder

$$\gamma(x) = \frac{1}{2} x^T B x, \qquad B \text{ positiv definit.} \tag{1.4}$$

Die Lösung von (1.4) wollen wir uns noch schnell etwas genauer ansehen. Dazu erinnern wir uns, daß ein *notwendiges* Kriterium für die Existenz eines Extremums von $f : \mathbb{R}^N \to \mathbb{R}$ unter der Nebenbedinung $g(x) = 0$, $g : \mathbb{R}^n \to \mathbb{R}^m$, durch die Existenz **Lagrange–Multiplikatoren** $\lambda \in \mathbb{R}^m$ gegeben ist, daß also

$$\nabla f(x) - \nabla g(x)\,\lambda = 0, \qquad \nabla g := \left[\frac{\partial}{\partial x_j} g_k \;:\; \begin{array}{c} j = 1, \ldots, n \\ k = 1, \ldots, m \end{array} \right], \tag{1.5}$$

gelten muss. Dieses Resultat lernt man entweder in der Analysis unter dem Stichwort "Extrema unter Nebenbedingungen", siehe z.B. (Heuser, 1983), oder in der Optimierung, siehe z.B. (Sauer, 2002b; Spellucci, 1993). Mit $f(x) = \frac{1}{2} x^T B x$ und $g(x) = Ax - b$ ist nun ganz einfach $\nabla f(x) = Bx$ sowie $\nabla g(x) = A^T$, so daß unsere gesuchte Optimallösung als Lösung von

$$\begin{array}{rcl} Bx - A^T\lambda &=& 0 \\ Ax &=& b \end{array} \qquad \Leftrightarrow \qquad \left[\begin{array}{cc} B & A^T \\ A & 0 \end{array} \right] \left[\begin{array}{c} x \\ -\lambda \end{array} \right] = \left[\begin{array}{c} 0 \\ b \end{array} \right]$$

gegeben ist. Daß dieses Extremum ein Minimum sein muss, liegt an der Tatsache, daß $x \mapsto x^T B x$ ein nach oben geöffnetes Paraboloid darstellt und deswegen nur ein Minimum haben kann. Anders gesagt: Die Minimallösung bestimmt man durch Lösen des *symmetrischen* quadratischen $m + n \times m + n$–Systems

$$Cx = d, \qquad C = \left[\begin{array}{cc} B & A^T \\ A & 0 \end{array} \right], \qquad d = \left[\begin{array}{c} 0 \\ b \end{array} \right], \tag{1.6}$$

wobei man sich nur um die ersten n Variablen zu kümmern braucht, die Werte von λ interessieren ja herzlich wenig.

Wir können die Tomographie in „pixelisierter" Form auch als ein inverses Problem angehen. Dazu diskretisieren wir zuerst einmal die **Region of Interest** in der Ebene, die wir der Einfachheit halber als das Quadrat $D = [0, 1]^2$ ansetzen wollen. Dieses zerlegen wir dann in die mn Teilquadrate

$$D_{jk} := \left[\frac{j-1}{m}, \frac{j}{m} \right] \times \left[\frac{k-1}{n}, \frac{k}{n} \right], \qquad j = 1, \ldots, m,\ k = 1, \ldots, n,$$

und für jeden Strahl durch D bestimmen wir, welche Teilquadrate D_{jk} dieser Strahl trifft und wie lange der Weg des Strahls durch D_{jk} ist, siehe Abb. 1.12.

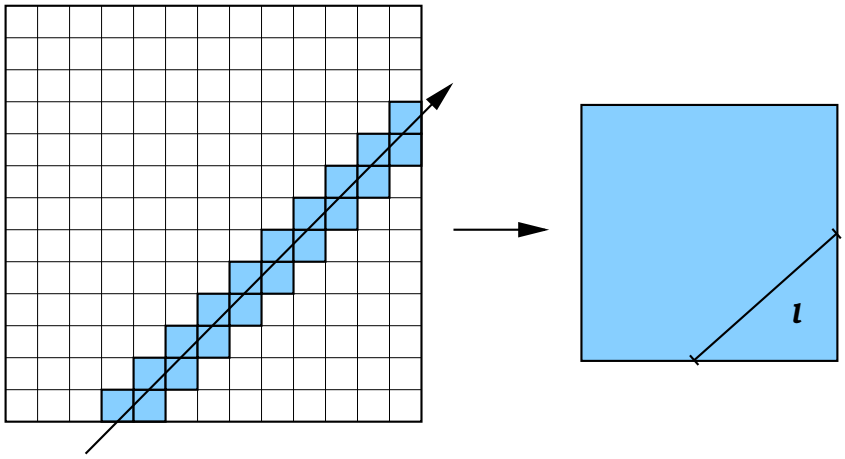


Abbildung 1.12: Verfolgung eines Strahls durch die ROI. Dieser Strahl trifft relativ wenige Pixel und sein Anteil in diesen Pixeln entspricht der Länge des „abgeschnittenen“ Strahl in diesem kleinen Quadrat *(rechts)*.

Das ist lediglich ein einfaches Clipping–Problem. Sei ℓ_{jk} die Länge dieses Strahlstücks, wobei wir natürlich $\ell_{jk} = 0$ setzen, wann immer der Strahl nicht durch das Rechteck geht. Seien ausserdem p_{jk} die **Pixelwerte** bzw. die lokale Dichte in D_{jk}, dann ergibt sich als Näherung für das pixelisierte Linienintegral entlang unseres Strahls der Wert

$$y = \sum_{j,k=1}^{m,n} \ell_{jk}\, p_{jk}. \tag{1.7}$$

Das ist auch schon fast das, was wir wollen, wir müssen nur noch ein wenig die Schreibweise ändern, um die ganze Sache in etwas vertrautere Form zu bringen. Dazu setzen wir

$$\mathbb{R}^{mn} \ni \mathbf{l} := \left[\ell_{n(j-1)+k} \;:\; j,k\right] \qquad \text{und} \qquad \mathbb{R}^{mn} \ni \mathbf{p} := \left[p_{n(j-1)+k} \;:\; j,k\right],$$

womit wir (1.7) kurz als $y = \mathbf{l}^T\mathbf{p}$ schreiben können. Für N Strahlen erhalten wir entsprechend die Gleichungen

$$y_j = \mathbf{l}_j^T\mathbf{p}, \quad j = 1,\ldots,N, \qquad \text{also} \qquad \mathbf{y} = \mathbf{L}\mathbf{p}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{l}_1^T \\ \vdots \\ \mathbf{l}_N^T \end{bmatrix} \in \mathbb{R}^{N\times nm}, \tag{1.8}$$

was genau das gesuchte, zumeist unterbestimmte[21] Gleichungssystem ist.

Übung 1.4 Implementieren Sie dieses Verfahren in `Matlab/Octave` und testen Sie die Rekonstruktionsqualität an selbstgewählten „Phantomen“. ♢

[20] Sonst gibt es normalerweise kein Minimum!

[21] Im „Normalfall“ einer gleichmäßigen Diskretisierung ist $m = n = N$.

1.6 Fazit

Was uns dieser kurze und völlig unvollständige Überblick zeigt, ist, daß es eine Vielzahl von Bilderfassungsverfahren gibt, die alle ihre eigenen mathematischen Bildmodelle haben und damit auch völlig andere Probleme, die zu lösen sind: Bei einem Foto muss man möglicherweise optische Verzerrungen kompensieren, bei einer EEG–Aufnahme die Einstreuungseffekte durch den umgebenden Wechselstrom berücksichtigen. Rauschen kann sich bei einer Bildklasse additiv, bei einer anderen multiplikativ bemerkbar machen und so weiter. Daher ist es vor der Anwendung aller Bildverarbeitungsverfahren immer eine gute Idee, sich zuerst einmal zu überlegen, ob und inwieweit diese auch für den jeweiligen Bildtyp geeignet sind. Das klingt banal, aber erschreckend oft glauben Anwender an die eierlegende Wollmilchsau, die jedes Problem erschlägt.

Mathematik ist Religion. Die Mathematiker sind die einzig Glücklichen. Wer ein mathematisches Buch nicht mit Andacht ergreift und es wie Gottes Wort liest, der versteht es nicht.

Novalis

2 Mathematische Grundlagen der Signalverarbeitung

Es hilft alles nichts: Die Verfahren der Bildverarbeitung sind Mathematik und je leistungsfähiger sie sind, desto substantiellere Mathematik verwenden sie. Deswegen werden wir uns jetzt auch erst einmal ein paar mathematische Grundlagen ansehen, die wir zu Bildverarbeitungszwecken brauchen.

2.1 Signalräume

Für uns soll vorerst ein Signal eine Abbildung $f : D \to \mathbb{R}$ sein, wobei $D \subseteq \mathbb{R}$ ist. Dabei betrachtet man vor allem die folgende Fälle des Definitionsbereichs D:

$D = \mathbb{R}$**:** Ein (prinzipiell) unbeschränktes *kontinuierliches* Signal.

$D = [a, b]$**:** Ein *zeitbeschränktes* kontinuierliches Signal. Mittels einer (affinen) Umskalierung des Intervalls können wir eigentlich immer gewährleisten, daß $a = 0$ und $b = 2\pi$ gilt. Ist außerdem $f(0) = f(2\pi)$, dann können wir das Signal **periodisch** zu einem unbeschränkten kontinuierlichen Signal fortsetzen, indem wir einfach

$$f(x + 2k\pi) := f(x), \qquad x \in [0, 2\pi], \quad k \in \mathbb{Z},$$

setzen. Analog können wir eine 2π–periodische Funktion auch als Funktion auf dem **Torus** $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z} \simeq [0, 2\pi]$ betrachten.

$D = \mathbb{Z}$**:** Ein *zeitdiskretes* Signal, also eine (doppeltunendliche) Folge. Wir werden also solche Gebilde bequemerweise als diskrete Funktionen schreiben.

Natürlich betrachtet man nicht beliebige, völlig unstrukturierte Signale, sondern solche, die gewissen mathematischen Voraussetzungen genügen[22], insbesondere in irgendeiner Form beschränkt sind. Daher erst einmal ein paar Definitionen.

[22]Und in Lehrbüchern für Signalverarbeitung oft einfach gemacht werden, ohne gesondert darauf hinzuweisen.

Definition 2.1 (Signalräume) *Wir bezeichnen mit* $L(\mathbb{R})$ *die Gesamtheit aller reellwertigen Funktionen*[23] *und mit* $\ell(\mathbb{Z})$ *alle reellen Folgen und definieren die folgenden Räume:*

1. Quadratsummierbare *Funktionen*

$$L_2(\mathbb{R}) := \left\{ f \in L(\mathbb{R}) \ : \ \infty > \|f\|_2 := \left(\int_{\mathbb{R}} |f(t)|^2 \ dt \right)^{1/2} \right\}$$

und Folgen

$$\ell_2(\mathbb{Z}) := \left\{ c \in \ell(\mathbb{Z}) \ : \ \infty > \|c\|_2 := \left(\sum_{k\in\mathbb{Z}} |c(k)|^2 \right)^{1/2} \right\};$$

man spricht in diesem Fall oftmals auch von endlicher Energie, *da die Gesamtenergie eines (diskreten oder kontinuierlichen) Signals gerade* $\|f\|_2$ *ist*[24].

2. (absolut)[25]summierbare *Funktionen und Folgen, definiert durch die Normen*

$$\|f\|_1 := \int_{\mathbb{R}} |f(t)| \ dt \qquad \textit{bzw.} \qquad \|c\|_1 := \sum_{k\in\mathbb{Z}} |c(k)| \, .$$

3. Beschränkte *Funktionen und Folgen unter Verwendung der Normen*

$$\|f\|_\infty := \sup_{t\in\mathbb{R}} |f(t)| \qquad \textit{bzw.} \qquad \|c\|_\infty := \sup_{k\in\mathbb{R}} |c(k)| \, .$$

Achtung: *Im kontinuierlichen Fall ist das Supremum ein* ***wesentliches Supremum****, das heißt, Mengen vom Maß* 0 *dürfen von der Supremumsbildung ausgeschlossen werden*[26].

4. *Funktionen und Folgen mit* endlichem Träger, *für die es ein* $N \in \mathbb{N}$ *gibt, so daß*

$$\left.\begin{matrix} \{t \in \mathbb{R} \ : \ f(t) \neq 0\} \\ \{k \in \mathbb{Z} \ : \ c(k) \neq 0\} \end{matrix}\right\} \subseteq [-N, N] \, .$$

Auch hier sind im kontinuierlichen Fall wieder Nullmengen auszunehmen. Solche Funktionen schreiben wir als $L_{00}(\mathbb{R})$ *bzw.* $\ell_{00}(\mathbb{Z})$.

Bemerkung 2.2 *Die Räume* $L_1(\mathbb{R})$, $L_2(\mathbb{R})$, $L_\infty(\mathbb{R})$ *sowie* $\ell_1(\mathbb{Z})$, $\ell_2(\mathbb{Z})$ *und* $\ell_\infty(\mathbb{Z})$ *sind jeweils ein* ***Banachraum****, also vollständig*[27] *und* $L_{00}(\mathbb{R})$ *bzw.* $\ell_{00}(\mathbb{Z})$ *ist* ***dicht*** *in* $L_1(\mathbb{R})$ *und* $L_2(\mathbb{R})$ *bzw.* $\ell_1(\mathbb{Z})$ *und* $\ell_2(\mathbb{Z})$, *aber nicht in* $L_\infty(\mathbb{R})$ *bzw.* $\ell_\infty(\mathbb{Z})$.

[23] Man könnte auch lokale Integrierbarkeit verlangen.

[24] Ich werde notationell **nicht** zwischen diskreten und kontinuierlichen Signalen unterscheiden, der Sinn wird sich ohnehin zumeist aus dem Kontext ergeben.

[25] Das "absolut" kann man sich eigentlich auch schenken, wenn man bedenkt, daß konvergente Reihen eigentlich immer absolut konvergent sein sollten, weil sonst der Grenzwert nicht wirklich vernünftig ist – Stichwort "Umordnungssatz".

[26] Preisfrage: Was ist dann $\sup \chi_{\mathbb{Q}}$?

[27] Zur Erinnerung: Das bedeutet, daß der Grenzwert jeder Cauchy–Folge auch zum Raum gehört.

Definition 2.3 (δ–Puls, Dirac–Puls) *Ein ganz besonderes Signal ist der "Puls"* $\delta \in \ell_{00}(\mathbb{Z})$*, definiert durch*[28]

$$\delta(k) = \delta_{0k} = \begin{cases} 1, & k = 0, \\ 0, & k \neq 0. \end{cases}$$

Man spricht hier oftmals auch von einer "Funktion" namens Dirac[29]*–*δ*, obwohl es sich dabei eigentlich um eine* ***Distribution*** *handelt.*

Nachdem man auf Rechnern eigentlich ja nur diskret arbeiten kann, empfiehlt es sich natürlich, ein eventuell vorhandenes kontinuierliches Signal[30] in ein zeitdiskretes Signal umzuwandeln; daß man eigentlich auch noch die kontinuierlichen Werte $f(t)$ in diskrete Werte umwandeln müsste – man spricht hier von **Quantisierung** – schenken wir uns an dieser Stelle und argumentieren wie in (Kammeyer & Kroschel, 1998), daß Quantisierungseffekte bei modernen doppeltgenauen Gleitkommaarithmetiken, siehe (Higham, 1996), kaum ins Gewicht fallen. Um von einem kontinuierlichen zu einem diskreten Signal überzugehen, definiert man den **Abtastoperator** $S_h : L(\mathbb{R}) \to \ell(\mathbb{Z})$ mit **Schrittweite** h als

$$(S_h f)(k) := f(hk), \qquad k \in \mathbb{Z}. \tag{2.1}$$

Bemerkung 2.4 *Eigentlich ergibt* S_h *für nichtstetige Funktionen gar keinen Sinn, da* f *nur auf der* ***Nullmenge*** $h\mathbb{Z}$ *betrachtet wird. Das kann man beheben, indem man* f *als stückweise stetig annimmt, oder aber die Werte von* f *an der Stelle* hk, $k \in \mathbb{Z}$*, als*

$$f(hk) = \lim_{\varepsilon \to 0} \frac{1}{2\varepsilon} \int_{-\varepsilon}^{\varepsilon} f(hk + t)\, dt$$

festlegt *und so zum* ***Lebesgue–Punkt*** *macht, siehe (Akhieser, 1988).*

Dies führt uns zur ersten interessanten *mathematischen* Frage, nämlich wann dieser Prozess umkehrbar ist, also unter welchen Voraussetzungen die Funktion f aus $S_h f$ rekonstruierbar ist. Diese Aussage, die eine Beziehung zwischen f und h herstellen wird, ist der berühmte *Abtastsatz von Shannon*, siehe Satz 2.16.

2.2 Fourier

Ein wichtiges Hilfsmittel bei der Betrachtung von Wavelets, aber auch in der Signalverarbeitung generell ist, vor allem in L_2 die **Fouriertransformierte** einer Funktion. Wir werden hier im wesentlichen den *Kalkül* der Fourier[31]–Analysis

[28] Die Notation δ_{jk} ist das "Kronecker–Delta" (also eigentlich "Kronecker–δ"), das, man glaubt es kaum, von Leopold Kronecker eingeführt wurde.

[29] Paul Adrien Maurice Dirac, 1902–1984, mathematischer Physiker mit wichtigen Beiträgen zur Quantenmechanik.

[30] Beispielsweise die Schalldruckwerte an einem Mikrophon.

[31] Jean Baptiste Fourier, 1768–1830, französischer Mathematiker und Politiker. Er war nicht nur Mitglied der "Acadèmie des Sciences", sondern (vorher) auch Teilnehmer an der Ägypten–Expedtion von Napoleon (Bonaparte) als wissenschaftlicher Berater und Gouverneur des Departement Isére mit Hauptstadt Grenoble. In den beiden letzteren Eigenschaften trug er nicht unwesentlich (als Förderer von Champollion) zur Entzifferung der Hieroglyphen bei, siehe (Doblhofer, 1990).

bereitstellen und uns weniger um ihre theoretischen Konzepte kümmern; für diese sei beispielsweise auf (Katznelson, 1976) verwiesen. Bei der Definition müssen und werden wir $f \in L_1(\mathbb{R})$ voraussetzen, was bei Funktionen mit *kompaktem* Träger sowieso einfacher ist.

Übung 2.1 Zeigen Sie: Ist $f \in L_2(\mathbb{R}) \cap L_{00}(\mathbb{R})$, dann ist $f \in L_1(\mathbb{R})$. ◊

Definition 2.5 *Für $f \in L_1(\mathbb{R})$ definieren wir die **Fouriertransformierte** $\widehat{f} : \mathbb{R} \to \mathbb{C}$ als*

$$\widehat{f}(\xi) := f^\wedge(\xi) := \int_{\mathbb{R}} f(t)\, e^{-i\xi t}\, dt, \qquad \xi \in \mathbb{R}, \tag{2.2}$$

und die Fouriertransformierte einer Folge $c \in \ell_1(\mathbb{Z})$ als diskretes Gegenstück

$$\widehat{c}(\xi) := c^\wedge(\xi) := \sum_{k\in\mathbb{Z}} c(k)\, e^{-ik\xi}, \qquad \xi \in \mathbb{R}. \tag{2.3}$$

Bemerkung 2.6

1. *In ihrer physikalischen oder technischen Interpretation liefert die Fouriertransformierte eines "Signals" (das man als Aplitudenfunktion der Zeit ansieht), den Anteil der entsprechenden Frequenz an diesem Signal.*

2. *Die Bedingung $f \in L_1(\mathbb{R})$ garantiert, daß die $\widehat{f}(\xi)$ für alle $\xi \in \mathbb{R}$ existiert:*

$$\left|\widehat{f}(\xi)\right| \le \int_{\mathbb{R}} |f(t)| \underbrace{\left|e^{-i\xi t}\right|}_{=1} dt = \|f\|_1 . \tag{2.4}$$

   Allerdings ist das "nur" hinreichend, aber eben nicht notwendig für die Existenz der Fouriertransformierten.

3. *Manchmal wird die Fouriertransformierte auch noch mit dem Vorfaktor $(2\pi)^{1/2}$ versehen, wir werden bald sehen, warum. Man sollte also bei der Verwendung von Literatur immer gut aufpassen, welche Normierung dort gewählt ist, sonst kann so ein konstanter Faktor für üble Fehler sorgen.*

4. *Man kann die Fouriertransformierte auch für allgemeinere "Funktionen"klassen als L_1 definieren, beispielsweise für* temperierte Distributionen, *siehe z.B. (Katznelson, 1976; Yosida, 1965).*

5. *Außerdem gibt es die Fouriertransformation nicht nur auf $\mathbb{R}$ oder $\mathbb{R}^n$ sondern auf lokal kompakten abelschen Gruppen unter Verwendung des Haar–Maßes; dann sieht man, daß die Fouriertransformierte auf der* dualen Gruppe *definiert ist. Das soll uns aber hier nicht stören, in unserem einfachen aber bedeutenden Spezialfall spielt $\mathbb{R}$ beide Rollen.*

6. *Die zwei bedeutendsten Gruppenoperationen[32] auf $\mathbb{R}$ sind die* Translation *und die* Skalierung *die durch die beiden Operatoren τ_y und σ_h, definiert als*

$$\tau_y f = f(\cdot + y) \qquad \textit{und} \qquad \sigma_h f = f(h\cdot)$$

   realisiert werden sollen.

[32] Das heißt, sie sind insbesondere invertierbar.

Definition 2.7 (Faltung) *Zu* $f, g \in L(\mathbb{R})$ *und* $c, d \in \ell(\mathbb{Z})$ *definieren wir*[33] *die* ***Faltung***

$$f * g := \int_{\mathbb{R}} f(\cdot - t)\, g(t)\, dt, \qquad * : L(\mathbb{R}) \times L(\mathbb{R}) \to L(\mathbb{R}), \tag{2.5}$$

sowie

$$c * d := \sum_{k\in\mathbb{Z}} c(\cdot - k)\, d(k), \qquad * : \ell(\mathbb{R}) \times \ell(\mathbb{R}) \to \ell(\mathbb{R}), \tag{2.6}$$

bzw.

$$c * f := f * c := \sum_{k\in\mathbb{Z}} f(\cdot - k)\, d(k), \qquad * : L(\mathbb{R}) \times \ell(\mathbb{R}) \to L(\mathbb{R}). \tag{2.7}$$

Die Faltung (2.5) bezeichnet man als kontinuierlich*, die in (2.6) als* diskret *und die in (2.7) als* semidiskret*.*

Als nächstes stellen wir ein paar einfache Eigenschaften der Fouriertransformierten zusammen – daß die Fouriertransformierte linear in f bzw. c ist, das braucht nicht mehr besonders betont werden.

Satz 2.8 (Eigenschaften der Fouriertransformierten) *Es gelten die folgenden Aussagen:*

1. *Für* $f \in L_1(\mathbb{R})$ *und* $y \in \mathbb{R}$ *ist*

$$(\tau_y f)^\wedge(\xi) = e^{iy\xi}\, \widehat{f}(\xi), \qquad \xi \in \mathbb{R}. \tag{2.8}$$

2. *Für* $f \in L_1(\mathbb{R})$ *und* $h > 0$ *ist*

$$(\sigma_h f)^\wedge(\xi) = \frac{\widehat{f}(h^{-1}\xi)}{h}, \qquad \xi \in \mathbb{R}. \tag{2.9}$$

3. *Für* $f, g \in L_1(\mathbb{R})$ *bzw.* $c, d \in \ell_1(\mathbb{Z})$ *ist* $f * g \in L_1(\mathbb{R})$ *bzw.* $c * d \in \ell_1(\mathbb{Z})$ *und es gilt für* $\xi \in \mathbb{R}$

$$(f * g)^\wedge(\xi) = \widehat{f}(\xi)\, \widehat{g}(\xi), \qquad \text{bzw.} \qquad (c * d)^\wedge(\xi) = \widehat{c}(\xi)\, \widehat{d}(\xi). \tag{2.10}$$

4. *Für* $f \in L_1(\mathbb{R})$ *und* $c \in \ell_1(\mathbb{Z})$ *ist* $f * c \in L_1(\mathbb{R})$ *und*

$$(f * c)^\wedge(\xi) = \widehat{f}(\xi)\, \widehat{c}(\xi), \xi \in \mathbb{R}. \tag{2.11}$$

5. *Sind* $f, f' \in L_1(\mathbb{R})$*, dann gilt*

$$\left(\frac{d}{dx} f\right)^\wedge(\xi) = i\xi\, \widehat{f}(\xi), \qquad \xi \in \mathbb{R}. \tag{2.12}$$

6. *Sind* $f, xf \in L_1(\mathbb{R})$*, dann ist* $\widehat{f}$ *differenzierbar und es gilt*

$$\frac{d}{d\xi} \widehat{f}(\xi) = (-ix f)^\wedge(\xi), \qquad \xi \in \mathbb{R}. \tag{2.13}$$

[33] Vorbehaltlich Existenz der unendlichen Summen.

7. *Sind* $f, \widehat{f} \in L_1(\mathbb{R})$*, dann ist*

$$f(x) = \left(\widehat{f}\right)^{\vee}(x) := \frac{1}{2\pi}\int_{\mathbb{R}} \widehat{f}(\vartheta)\, e^{ix\vartheta}\, d\vartheta \tag{2.14}$$

Die Operation $f \mapsto f^{\vee} := \frac{1}{2\pi} f^{\wedge}(-\cdot)$ *bezeichnet man als* inverse Fouriertransformation[34].

Beweis: Für 1) berechnen wir

$$\begin{aligned}(\tau_y f)^{\wedge}(\xi) &= \int_{\mathbb{R}} f(t+y)\, e^{-i\xi t}\, dt = \int_{\mathbb{R}} f(t)\, e^{-i\xi(t-y)}\, dt = e^{iy\xi}\int_{\mathbb{R}} f(t)\, e^{-i\xi t}\, dt \\ &= e^{iy\xi}\,\widehat{f}(\xi),\end{aligned}$$

während 2) ganz ähnlich mit

$$(\sigma_h f)^{\wedge}(\xi) = \int_{\mathbb{R}} f(ht)\, e^{-i\xi t}\, dt = \frac{1}{h}\int_{\mathbb{R}} f(t)\, e^{-i(\xi/h)t}\, dt = \frac{\widehat{f}\left(\frac{\xi}{h}\right)}{h}$$

bewiesen wird. Die erste Aussage von 3) folgt aus

$$\|f * g\|_1 = \int_{\mathbb{R}} \left|\int_{\mathbb{R}} f(t) g(s-t)\, dt\right| ds \le \int_{\mathbb{R}}\int_{\mathbb{R}} |f(t)g(s)|\, dt\, ds = \|f\|_1\, \|g\|_1$$

bzw.

$$\|c * d\|_1 = \sum_{j\in\mathbb{Z}} \left|\sum_{k\in\mathbb{Z}} c(k)\, d(j-k)\right| \le \sum_{j,k\in\mathbb{Z}} |c(k)\, d(j)| = \|c\|_1\, \|c\|_1,$$

der zweite, etwas interessantere Teil hingegen aus

$$\begin{aligned}(f*g)^{\wedge}(\xi) &= \int_{\mathbb{R}}\left(\int_{\mathbb{R}} f(s) g(t-s)\, ds\right) e^{-i\xi t} dt = \int_{\mathbb{R}}\int_{\mathbb{R}} f(s)\, e^{i\xi s}\, g(t-s)\, e^{i\xi(t-s)}\, ds dt \\ &= \widehat{f}(\xi)\,\widehat{g}(\xi),\end{aligned}$$

bzw.

$$\begin{aligned}(c*d)^{\wedge}(\xi) &= \sum_{j\in\mathbb{Z}}\left(\sum_{k\in\mathbb{Z}} c(k)\, d(j-k)\right) e^{-ij\xi} = \sum_{j,k\in\mathbb{Z}} c(k)\, e^{-ik\xi}\, d(j-k)\, e^{-i(j-k)\xi} \\ &= \widehat{f}(\xi)\,\widehat{g}(\xi).\end{aligned}$$

4) folgt aus Übung 2.2 und

$$\begin{aligned}(f*c)^{\wedge}(\xi) &= \int_{\mathbb{R}} \sum_{k\in\mathbb{Z}} f(t-k)\, c(k) e^{-i\xi t}\, dt = \int_{\mathbb{R}} \sum_{k\in\mathbb{Z}} f(t-k)\, e^{-i\xi(t-k)}\, c(k)\, e^{-ik\xi} \\ &= \widehat{f}(\xi)\,\widehat{c}(\xi).\end{aligned}$$

[34] Die Gründe dafür sind ja wohl offensichtlich.

oder auch direkt unter Verwendung von (2.8). Für 5 verwenden wir partielle Integration[35], um

$$(f')^\wedge(\xi) = \int_{\mathbb{R}} \frac{df}{dt}(t) e^{-i\xi t}\,dt = -\int_{\mathbb{R}} f(t) \frac{d}{dt} e^{-i\xi t}\,dt = i\xi \int_{\mathbb{R}} f(t) e^{-i\xi t}\,dt = i\xi\,\widehat{f}(\xi).$$

6) erhalten wir, indem wir für $h > 0$ den Differenzenquotient

$$\frac{\widehat{f}(\xi+h) - \widehat{f}(\xi)}{h} = \int_{\mathbb{R}} f(t)\,\frac{e^{-i(\xi+h)t} - e^{-i\xi t}}{h}\,dt = \int_{\mathbb{R}} f(t)\,e^{-i\xi t}\,\frac{e^{-iht}-1}{h}\,dt$$

betrachten; das Integral existiert, weil $xf \in L_1(\mathbb{R})$ und da

$$\lim_{h\to 0} \frac{e^{-iht}-1}{h} = \lim_{h\to 0}(-it)\,e^{-iht} = -it$$

ist, folgt (2.13). Der Beweise von 7) ist ein klein wenig aufwendiger und verwendet die *Fejér–Kerne*

$$F_\lambda := \lambda F(\lambda\cdot), \quad \lambda > 0, \qquad F(x) := \frac{1}{2\pi}\int_{-1}^{1} (1-|t|)\;e^{ixt}\,dt, \quad x \in \mathbb{R},$$

auf $\mathbb{R}$, die die Eigenschaft haben, daß für jedes $f \in L_1(\mathbb{R})$

$$\lim_{\lambda\to\infty} \|f - f * F_\lambda\| = 0, \tag{2.15}$$

siehe (Katznelson, 1976, S. 124–126), also auch $f*F_\lambda \to f$ punktweise fast überall[36]. Dann ist aber für $x \in \mathbb{R}$

$$\begin{aligned}
f * F_\lambda(x) &= \frac{1}{2\pi}\int_{\mathbb{R}} f(t)\left(\lambda\int_{-1}^{1}(1-|\vartheta|)\,e^{i(x-t)\lambda\vartheta}\,d\vartheta\right)dt \\
&= \frac{1}{2\pi}\int_{\mathbb{R}} f(t)\int_{-\lambda}^{\lambda}\left(1-\frac{|\vartheta|}{\lambda}\right)e^{i(x-t)\vartheta}\,d\vartheta\,dt \\
&= \frac{1}{2\pi}\int_{-\lambda}^{\lambda}\left(1-\frac{|\vartheta|}{\lambda}\right)\underbrace{\int_{\mathbb{R}} f(t)\;e^{-it\vartheta}\,dt}_{=\widehat{f}(\vartheta)}\;e^{ix\vartheta}\,d\vartheta \\
&= \frac{1}{2\pi}\int_{-\lambda}^{\lambda}\left(1-\frac{|\vartheta|}{\lambda}\right)\widehat{f}(\vartheta)\,e^{ix\vartheta}\,d\vartheta \\
&= \underbrace{\frac{1}{2\pi}\int_{0\le|\vartheta|\le\sqrt{\lambda}}\overbrace{\left(1-\frac{|\vartheta|}{\lambda}\right)}^{1-1/\sqrt{\lambda}\le\cdot\le 1}\widehat{f}(\vartheta)\,e^{ix\vartheta}\,d\vartheta}_{\to\frac{1}{2\pi}\int_{\mathbb{R}}\widehat{f}(\vartheta)\,e^{ix\vartheta}\,d\vartheta} + \underbrace{\frac{1}{2\pi}\int_{\sqrt{\lambda}\le|\vartheta|\le\lambda}\left(1-\frac{|\vartheta|}{\lambda}\right)\widehat{f}(\vartheta)\,e^{ix\vartheta}\,d\vartheta}_{\to 0},
\end{aligned}$$

[35] Daß dies gerechtfertigt ist, liegt an der Tatsache, daß für $f \in L_1(\mathbb{R})$ immer $\lim_{x\to\pm\infty}|f(x)| = 0$ sein muß und daß die stetigen Funktionen mit kompaktem Träger bezüglich der Norm $\|\cdot\|_1$ *dicht* in $L_1(\mathbb{R})$ sind. Deswegen muß man sich um "Randwerte" hier nicht kümmern.

[36] Zumindest für eine Teilfolge, siehe (Forster, 1984, S. 96).

weil $\widehat{f} \in L_1(\mathbb{R})$. □

Übung 2.2 Zeigen Sie, daß für $f \in L_1(\mathbb{R})$ und $c \in \ell_1(\mathbb{Z})$ die Ungleichung

$$\|f * c\|_1 \le \|f\|_1 \, \|c\|_1$$

gilt. ◊

Übung 2.3 Beweisen Sie *ohne* Verwendung von (2.12) die folgende Aussage: Sind $f, f' \in L_1(\mathbb{R})$, dann ist $(f')^\wedge(0) = 0$.
Hinweis: Partielle Integration. ◊

Übung 2.4 Zeigen Sie, daß sich für $f \in L_1(\mathbb{R})$ und $h \neq 0$ die Identität

$$(\sigma_h f)^\wedge = \frac{\widehat{f}(h^{-1}\cdot)}{|h|} \tag{2.9'}$$

ergibt. ◊

Beispiel 2.9 *Berechnen wir doch mal zu Übungszwecken so eine Fouriertransformierte, und zwar die der kardinalen B–Splines* N_j, $j \in \mathbb{N}_0$, *definiert durch* $N_0 = \chi$ *und* $N_j = \chi * N_{j-1}$, $j \in \mathbb{N}$. *Insbesondere ist also*

$$\widehat{N_0}(\xi) = \widehat{\chi}(\xi) = \int_{\mathbb{R}} \chi(t)\, e^{-i\xi t}\, dt = \int_0^1 e^{-i\xi t}\, dt = \frac{e^{-i\xi t}}{-i\xi}\Big|_{t=0}^{1} = \frac{1 - e^{-i\xi}}{i\xi}$$

und somit, nach (2.10),

$$\widehat{N_j}(\xi) = (\widehat{\chi}(\xi))^{j+1} = \left(\frac{1 - e^{-i\xi}}{i\xi}\right)^{j+1}.$$

Übung 2.5 Die *zentrierten B–Splines* M_j, $j \in \mathbb{N}_0$, sind definiert als

$$M_j = \underbrace{\chi_{[-1/2,1/2]} * \cdots * \chi_{[-1/2,1/2]}}_{j+1}.$$

Zeigen Sie:

1. Diese Funktionen sind gerade: $M_j(-x) = M_j(x)$, $x \in \mathbb{R}$.
2. Für $j \in \mathbb{N}_0$ ist
$$\widehat{M_j}(\xi) = \left(\frac{\sin \xi/2}{\xi/2}\right)^{j+1}, \qquad \xi \in \mathbb{R}.$$

◊

Ist $f \in L_1(\mathbb{R})$, dann ist für $\xi, \eta \in \mathbb{R}$

$$\left|\widehat{f}(\xi + \eta) - \widehat{f}(\xi)\right| \le \int_{\mathbb{R}} |f(t)| \underbrace{\left|e^{-i\xi t}\right|}_{=1} \left|e^{-i\eta t} - 1\right| dt,$$

was auf der rechten Seite unabhängig von ξ ist und mit $\eta \to 0$ gegen Null konvergiert, denn für jedes $\varepsilon > 0$ gibt es ein $N > 0$, so daß

$$\int_{|t|>N} |f(t)|\, dt < \varepsilon$$

ist, während wir, durch Wahl eines hinreichend kleinen Wertes von η, die Funktion $|e^{-i\eta t} - 1|$ auf $[-N, N]$ so klein machen können, wie wir wollen. Der langen Rede kurzer Sinn:

> *Ist* $f \in L_1(\mathbb{R})$, *so ist* $\widehat{f} \in C_u(\mathbb{R})$, *dem Vektorraum der gleichmäßig stetigen und gleichmäßig beschränkten*[37] *Funktionen auf* $\mathbb{R}$.

Außerdem kann man sogar sagen, wie sich die Fouriertransformierte für $|\xi| \to \infty$ benimmt, das ist das klassische Riemann[38]–Lebesgue[39]–Lemma.

Proposition 2.10 (Riemann–Lebesgue–Lemma)
Ist $f \in L_1(\mathbb{R})$, *so ist*

$$\lim_{\xi \to \pm\infty} \widehat{f}(\xi) = 0. \tag{2.16}$$

Beweis: Ist auch $f' \in L_1(\mathbb{R})$ so folgt (2.16) sofort mittels (2.12) und (2.4):

$$\|f'\|_1 \geq \left|(f')^{\wedge}(\xi)\right| = |\xi| \left|\widehat{f}(\xi)\right|, \qquad \xi \in \mathbb{R},$$

also $\left|\widehat{f}(\xi)\right| \leq \|f'\|_1 / |\xi| \to 0$ für $|\xi| \to \infty$. Für beliebiges $f \in L_1(\mathbb{R})$ und differenzierbares $g \in L_1(\mathbb{R})$ mit[40] mit $\|f - g\|_1 \leq \varepsilon$ ist

$$\|f - g\|_1 \geq \left|\widehat{f}(\xi) - \widehat{g}(\xi)\right| \geq \left|\widehat{f}(\xi)\right| - |\widehat{g}(\xi)|,$$

also

$$\lim_{|\xi|\to\infty} \left|\widehat{f}(\xi)\right| \leq \lim_{|\xi|\to\infty} |\widehat{g}(\xi)| + \|f - g\|_1 \leq \varepsilon$$

und da man ε beliebig klein wählen kann, folgt die Behauptung. □

Wie sieht es nun auf anderen L_p–Räumen, $p \neq 1$, insbesondere mit $L_2(\mathbb{R})$ aus[41]? Hier nutzt man aus, daß $L_1(\mathbb{R}) \cap L_p(\mathbb{R})$ eine *dichte Teilmenge* von $L_p(\mathbb{R})$ ist. Für L_2 gibt es nun noch eine besonders schöne Eigenschaft, nämlich eine Isometrie, für die Parseval[42] bzw. Plancherel[43] die Namensgeber sind.

---

[37] Siehe (2.4).

[38] Georg Friedrich Bernhard Riemann, 1826–1866, Schüler von Gauss mit Beiträgen zu Analysis, Algebra, Geometrie.

[39] Henri Lebesgue, 1875–1941, sein bedeutendstes Werk war seine Dissertation "Intégrale, longueur, aire" (1902).

[40] Man beachte, daß sogar die *unendlich oft stetig differenzierbaren Funktionen mit kompaktem Träger* eine dichte Teilmenge von $L_1(\mathbb{R})$ bilden.

[41] $L_2(\mathbb{R})$ spielt in der Signalverarbeitung schon deswegen so eine wesentliche Rolle, weil das gerade die Signale (und die sind normalerweise nicht unbedingt stetig) mit *endlicher Energie* sind – eine ziemlich realistische Annahmen, oder nicht?

[42] Marc–Antoine Parseval des Chênes, 1755–1836, Zeitgenosse von Fourier, der ziemlich heftig in die Wirren der französischen Revolution verwickelt wurde, publizierte überhaupt nur 5 (in Worten: "fünf") Arbeiten, die er aber allesamt der *Académie des Sciences* vorlegte.

[43] Leider keine biografischen Daten.

Satz 2.11 (Parseval/Plancherel) *Für* $f, g \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ *ist*

$$\int_{\mathbb{R}} f(t)\, g(t)\, dt = \frac{1}{2\pi} \int_{\mathbb{R}} \widehat{f}(\vartheta)\, \overline{\widehat{g}(\vartheta)}\, d\vartheta, \tag{2.17}$$

also insbesondere, mit $f = g$,

$$\|f\|_2 = \frac{1}{\sqrt{2\pi}} \left\|\widehat{f}\right\|_2 . \tag{2.18}$$

Wie die Mathematiker sagen: Die Fouriertransformation ist bis auf Normierung[44] *eine* ***Isometrie*** *auf* L_2.

Diese Aussage hilft uns nun, die Definition der Fouriertransformierten auf $L_2(\mathbb{R})$ zu übertragen: Zu $f \in L_2(\mathbb{R})$ betrachtet man eine Folge

$$f_n := \chi_{[-n,n]} \cdot f \quad \in L_1(\mathbb{R}) \cap L_2(\mathbb{R}), \qquad n \in \mathbb{N},$$

die für $n \to \infty$ in der Norm $\|\cdot\|_2$ gegen f konvergiert. Da

$$\left\|\widehat{f}_{n+k} - \widehat{f}_n\right\|_2 = \left\|(f_{n+k} - f)^{\wedge}\right\|_2 = \sqrt{2\pi}\, \|f_{n+k} - f_n\|_2 , \qquad k, n \in \mathbb{N},$$

ist $\widehat{f}_n$ eine Cauchyfolge und konvergiert gegen eine Funktion in $L_2(\mathbb{R})$, die wir $\widehat{f}$ nennen wollen.

Beweis von Satz 2.11: Wir definieren

$$h(x) = \int_{\mathbb{R}} f(t)\, g\,(t - x)\; dt = (f * g(-\cdot))\,(x), \qquad x \in \mathbb{R},$$

und erhalten, daß $h(0) = \int fg$. Außerdem ist

$$\widehat{h}(\xi) = \widehat{f}(\xi)\; \underbrace{(g(-\cdot))^{\wedge}\,(\xi)}_{=\overline{\widehat{g}(\xi)}} = \widehat{f}(\xi)\, \overline{\widehat{g}(\xi)}, \qquad \xi \in \mathbb{R}.$$

Sind nun f und g so "brav", daß $\widehat{f}, \widehat{g} \in L_2(\mathbb{R})$ ist[45], dann ist mit (2.14)

$$\frac{1}{2\pi} \int_{\mathbb{R}} \widehat{f}(\vartheta)\, \overline{\widehat{g}(\vartheta)}\, d\vartheta = \frac{1}{2\pi} \int_{\mathbb{R}} \widehat{h}(\vartheta)\, e^{i0\theta}\, d\vartheta = h(0) = \int_{\mathbb{R}} f(t)\, g(t)\, dt,$$

was (2.17) liefert. Und die *Plancherel–Identität* (2.18) ist dann eine unmittelbare Konsequenz aus der *Parseval–Formel* (2.17). □

Jetzt machen wir als nächstes einen kleinen Abstecher in die Welt der Fourieranalysis auf dem Torus $\mathbb{T}$, bei dem wir den Begriff der **Fourierreihe** kennelernen und mit den bisherigen Fourierismen in Beziehung bringen werden. Tatsächlich wird nämlich der Beweis des Shannonschen Abtastsatzes, Satz 2.16 ebenfalls auf der Interaktion zwischen Fourierreihen und der Fouriertransformierten basieren. Doch dazu sollten wir erst einmal die Fourierreihe einer Funktion definieren.

[44] Und wen interessiert bitte Normierung? Aber im Ernst, eine Normierung kann man durch geeignete Multiplikation mit einer unbestritten langweiligen Konstanten immer kompensieren.

[45] Das ist beispielsweise der Fall, wenn f und g differenzierbar sind; dies folgt aus (2.12) und dem Riemann–Lebesgue–Lemma, Proposition 2.10.

Definition 2.12 *Zu* $f \in L_1(\mathbb{T})$[46] *sind die* ***Fourierkoeffizienten*** *definiert als*

$$f_k := \frac{1}{2\pi}\int_{\mathbb{T}} f(t)\, e^{-ikt}\, dt, \qquad k \in \mathbb{Z},$$

und die ***Fourierreihe***[47] *zu* f *als*

$$\mathscr{F} f := \sum_{k\in\mathbb{Z}} f_k\, e^{ik\cdot}. \tag{2.19}$$

Eigentlich sollte uns die trigonometische Reihe in (2.19) bekannt vorkommen: Definieren wir nämlich zu $f \in L_1(\mathbb{T})$ mit Fourierkoeffizienten f_k, $k \in \mathbb{Z}$, die Folge

$$c_f(k) = f_k, \qquad k \in \mathbb{Z},$$

dann ist $c_f \in \ell_1(\mathbb{Z})$ und $\mathscr{F} f = \widehat{c_f}$. Da außerdem für $j, k \in \mathbb{Z}$

$$\int_{\mathbb{T}} e^{-ijt} e^{ikt}\, dt \;=\; \int_{-\pi}^{\pi} e^{i(k-j)t}\, dt = \begin{cases} 2\pi, & j = k, \\ \frac{1}{i(k-j)} e^{i(k-j)t}\Big|_{t=-\pi}^{\pi} = 0, & j \neq k, \end{cases}$$

erhalten wir für $k \in \mathbb{Z}$, daß

$$c(k) = \frac{1}{2\pi}\int_{\mathbb{T}} \sum_{j\in\mathbb{Z}} c(j) e^{ijt}\, e^{-ikt}\, dt = \frac{1}{2\pi}\int_{\mathbb{T}} \widehat{c}(\theta)\, e^{ik\theta}\, d\theta =: (\widehat{c})^{\vee}(k). \tag{2.20}$$

Mit anderen Worten: Wir haben eine *Inverse* zur Fouriertransformierten einer Folge gefunden. Daß diese (formale) Rechnung auch wirklich in Ordnung ist kann man sich übrigens leicht überlegen.

Übung 2.6 Zeigen Sie, daß für $c \in \ell_1(\mathbb{Z})$ auch $\widehat{c} \in L_1(\mathbb{T})$ gilt. ◊

Der folgende Satz stellt eine weitere zentrale Beziehung zwischen Fourierreihen un der Fouriertransformierten her und liefert eine Identität, die sich schon oft genug als hilfreich erwiesen hat, nämlich die Poissonsche[48] Summenformel.

Satz 2.13 (Poisson–Summenformel) *Für* $f \in L_1(\mathbb{R})$ *ist*

$$\sum_{k\in\mathbb{Z}} f(2k\pi) = \frac{1}{2\pi}\sum_{k\in\mathbb{Z}} \widehat{f}(k) \qquad \textit{und} \qquad \sum_{k\in\mathbb{Z}} f(k) = \sum_{k\in\mathbb{Z}} \widehat{f}(2k\pi)\,. \tag{2.21}$$

Beweis: Wir setzen

$$g(x) = \sum_{k\in\mathbb{Z}} f(x + 2k\pi)\,, \qquad x \in \mathbb{R}, \tag{2.22}$$

[46] Man bemerke, daß $L_p(\mathbb{T}) \subset L_1(\mathbb{T})$ für $1 < p \leq \infty$ ist.

[47] Eine sogenannte **trigonometrische Reihe**.

[48] Siméon Denis Poisson, 1781–1840, studierte bei Laplace und Legendre, Beiträge zur Fourier–Analysis und Wahrscheinlichkeitstheorie ("Poisson–Verteilung"), schrieb zwischen 300 und 400 Arbeiten, auch über Elektrizität, Magnetismus und Astronomie.

und bemerken, daß $g(x+2\pi) = g(x)$, also g eine 2π–periodische Funktion ist, und wegen

$$\begin{aligned}\|g\|_{\mathbb{T},1} &= \int_{\mathbb{T}} |g(t)|\,dt = \int_{\mathbb{T}} \left|\sum_{k\in\mathbb{Z}} f(t+2k\pi)\right| dt \le \sum_{k\in\mathbb{Z}} \int_0^{2\pi} |f(t+2k\pi)|\,dt \\ &= \int_{\mathbb{R}} |f(t)|\,dt = \|f\|_{\mathbb{R},1}\end{aligned}$$

ist $g \in L_1(\mathbb{T})$ und insbesondere wohldefiniert – die Summe in (2.22) divergiert nicht allzu unmotiviert. Die Fourierkoeffizienten g_k von g haben dann die Form

$$g_k = \frac{1}{2\pi}\int_{\mathbb{T}} g(t)\,e^{-ikt}\,dt = \frac{1}{2\pi}\int_{\mathbb{T}} \sum_{\ell\in\mathbb{Z}} f(t+2\ell\pi)\;e^{-ikt}\,dt = \frac{1}{2\pi}\int_{\mathbb{R}} f(t)\,e^{-ikt}\,dt = \frac{1}{2\pi}\,\widehat{f}(k)$$

und, angenommen die Partialsummen der Fourierreihe von g würden konvergieren[49], erhalten so, daß

$$\frac{1}{2\pi}\sum_{k\in\mathbb{Z}} \widehat{f}(k) = \sum_{k\in\mathbb{Z}} g_k \underbrace{e^{ik0}}_{=1} = g(0) = \sum_{k\in\mathbb{Z}} f(0+2k\pi) = \sum_{k\in\mathbb{Z}} f(2k\pi)\,,$$

was die erste Identität liefert. Mit deren Hilfe und (2.9) ergibt sich dann, daß

$$\sum_{k\in\mathbb{Z}} f(k) = \sum_{k\in\mathbb{Z}} \left(\sigma_{(2\pi)^{-1}} f\right)(2k\pi) = \frac{1}{2\pi}\sum_{k\in\mathbb{Z}} \left(\sigma_{(2\pi)^{-1}} f\right)^{\wedge}(k) = \sum_{k\in\mathbb{Z}} \widehat{f}\,(2k\pi)\,.$$

□

2.3 Der Abtastsatz

Jetzt können wir auch schon unsere erste "große" Frage beantworten, nämlich welche Funktionen aus ihrer Abtastfolge eindeutig rekonstruiert werden können. Dazu zwei Begriffe.

Definition 2.14

1. *Eine Funktion* $f \in L_1(\mathbb{R})$ *heißt* ***bandbeschränkt*** *mit* ***Bandbreite*** T *oder kurz* T–bandbeschränkt, *wenn*

$$\widehat{f}(\xi) = 0, \qquad \xi \notin [-T, T].$$

2. *Die* ***Sinus Cardinalis*** *oder* sinc*–Funktion*[50] *ist definiert als*

$$\operatorname{sinc}(x) := \frac{\sin \pi x}{\pi x}, \qquad x \in \mathbb{R}.$$

[49] Ansonsten müssten wir ein Summationsverfahren, beispielsweise den Féjer–Kern verwenden.

[50] In Ingenieurskreisen auch als "si"–Funktion bezeichnet.

Bemerkung 2.15 *Wegen*

$$\operatorname{sinc} 0 = \lim_{x\to 0} \frac{\sin \pi x}{\pi x} = \lim_{x\to 0} \frac{\cos x}{1} = \cos 0 = 1$$

ist

$$(S_1 \operatorname{sinc})(k) = \operatorname{sinc}(k) = \delta(k), \qquad k \in \mathbb{Z}. \tag{2.23}$$

Daher stammt dann auch der Name kardinal*: Die Funktion hat an $\mathbb{Z}$ die Werte $0/1$.*

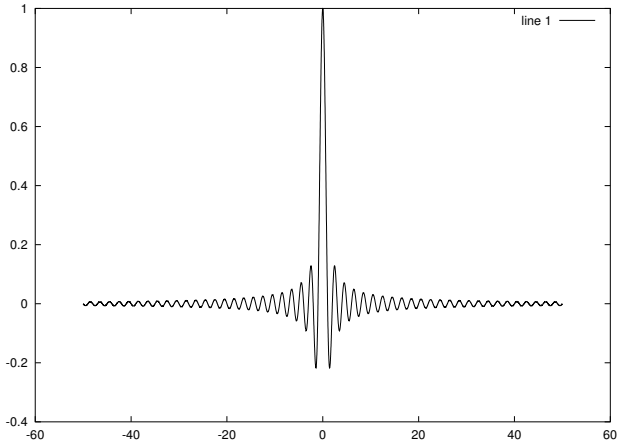


Abbildung 2.1: Die Funktion sinc .

Das nächste Resultat, der Shannonsche[51] Abtastsatz[52], sagt uns nun, daß man bandbeschränkte Funktionen rekonstruierbar abtasten kann.

Satz 2.16 (Abtastsatz von Shannon) *Ist $f \in L_1(\mathbb{R})$ eine T–bandbeschränkte Funktion und ist $h < h^* = \frac{\pi}{T}$, dann ist*

$$f = (S_h f * \operatorname{sinc})\left(h^{-1}\cdot\right) = \sum_{k\in\mathbb{Z}} f(hk)\, \frac{\sin \pi (x/h - k)}{\pi (x/h - k)}. \tag{2.24}$$

Bemerkung 2.17 *Die Kopplung von Bandbreite T der Funktion und Auflösung h ist ein wirklich zentrales Konzept der digitalen Signalverarbeitung. Daher noch ein paar Anmerkungen:*

1. *In vielen Büchern heißt es, daß die **Abtastfrequenz** $1/h$, oftmals auch als **Nyquist–Frequenz** bezeichnte, die* Hälfte *der maximalen Frequenz T sein soll, siehe z.B. (Kammeyer & Kroschel, 1998). Solche Konstanten kommen von einer etwas anderen Normierung von Frequenzen (mit 2π) oder auch der Fouriertransformierten.*

[51] Claude Elwood Shannon, 1916–2001, Elektroingenieur und Mathematiker, Erfinder des Wortes "bit" und Entwickler von Schachprogrammen (und zwar um 1950).

[52] Der eigentlich gar nicht von Shannon ist, siehe Bemerkung 2.17.

2. *Die* sinc*–Funktion ist kein wirklich guter Weg zur Rekonstruktion eines Signals. Sie hat ja keinen endlichen Träger und fällt nur linear*[53] *ab, so daß man mit Abschneiden und den dabei auftretenden Fehlern* sehr *vorsichtig sein muß.*

3. *In vielen Fällen wählt man die Abtastrate* h *einfach als* h^*/k *für ein ganzzahliges* k*, was nichts anderes ist als die Abtastfrequenz* k*–mal so groß wie nötig anzusetzen. In diesem Fall spricht man von "*k–fachem Oversampling"[54].

4. *Die Beweise des Abtastsatzes aus Bücher der elektrotechnischen Literatur, z.B. (Grüningen, 1993) oder (Schüßler, 1992), und die dort verwendeten Argumentationen, sind oftmals (zumindest für Mathematiker) nur schwer nachzuvollziehen. Daß es auch anders geht sieht man in (Mallat, 1999). Der jetzt folgende Beweis ist eine Modifikation des Beweises aus (Kammeyer & Kroschel, 1998).*

5. *Auch die Herkunft des Satzes ist nicht so ganz klar. Laut (Mallat, 1999) wurde er zuerst (theoretisch) 1935 von Whittaker bewiesen (Whittaker, 1935) und 1949 von Shannon im Kontext der Signalverarbeitung wiederentdeckt (Shannon, 1949).*

Beweis von Satz 2.16: Wegen der Bandbeschränktheit ist $\widehat{f} \in C_u(\mathbb{R}) \cap L_{00}(\mathbb{R})$, also $\widehat{f} \in L_1(\mathbb{R})$. Für $h > 0$ und $k \in \mathbb{Z}$ ist gemäß der *diskreten* Beziehung (2.20)

$$S_h f(k) = \Big((S_h f)^\wedge\Big)^\vee(k) = \frac{1}{2\pi}\int_{\mathbb{T}} (S_h f)^\wedge(\theta)\, e^{ik\theta}\, d\theta, \tag{2.25}$$

aber eben auch

$$\begin{aligned} S_h f(k) &= f(hk) = \widehat{f^\vee}(hk) = \frac{1}{2\pi}\int_{\mathbb{R}} \widehat{f}(\theta)\, e^{ihk\theta}\, d\theta = \frac{1}{2\pi}\sum_{j\in\mathbb{Z}} \int\limits_{h^{-1}([-\pi,\pi]+2\pi j)} \widehat{f}(\theta)\, e^{ihk\theta}\, d\theta \\ &= \frac{1}{2\pi}\sum_{j\in\mathbb{Z}} h^{-1} \int\limits_{-\pi+2j\pi}^{\pi+2\pi j} \widehat{f}\Big(h^{-1}\theta\Big)\, e^{ik\theta}\, d\theta = \frac{1}{2\pi h}\sum_{j\in\mathbb{Z}} \int_{-\pi}^{\pi} \widehat{f}\Big(h^{-1}(\theta+2\pi j)\Big)\, e^{ik\theta}\, d\theta \\ &= \frac{1}{2\pi h}\int_{-\pi}^{\pi}\left(\sum_{j\in\mathbb{Z}} \widehat{f}\Big(h^{-1}(\theta+2\pi j)\Big)\right) e^{ik\theta} d\theta, =: \frac{1}{2\pi}\int_{-\pi}^{\pi} F(\theta)\, e^{ik\theta}, \end{aligned}$$

das heißt

$$S_h f(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} F(\theta)\, e^{ik\theta}, \qquad F = \frac{1}{h}\sum_{j\in\mathbb{Z}} \widehat{f}\Big(h^{-1}(\cdot + 2\pi j)\Big) \tag{2.26}$$

wobei $F \in C(\mathbb{T}) \subset L_1(\mathbb{T})$ ist – die Funktion ist offensichtlich 2π–periodisch und wegen der Bandbeschränktheit von f ist für $\theta \in [0, 2\pi]$ die Summen nur endlich. Da die Exponentialfunktionen $e^{ik\cdot}$, $k \in \mathbb{Z}$, ein vollständiges Orthonormalsystem bilden, können wir aus (2.25) und (2.26) folgern, daß

$$h^{-1}\sum_{j\in\mathbb{Z}} \widehat{f}\Big(h^{-1}(\theta+2\pi j)\Big) = F(\theta) = (S_h f)^\wedge(\theta) = \sum_{j\in\mathbb{Z}} f(hj)\, e^{-ij\theta}, \qquad \theta \in \mathbb{T}. \tag{2.27}$$

[53] Also wie $1/x$.

[54] Was ja bespielsweise von CD–Playern bekannt sein sollte.

Ist nun h so klein, daß

$$h^{-1}\,[-\pi,\pi] \supseteq [-T,T] \quad\Longleftrightarrow\quad [-\pi,\pi] \supseteq [-Th,Th] \quad\Longleftrightarrow\quad Th < \pi \quad\Longleftrightarrow\quad h < \frac{\pi}{T},$$

dann erhalten wir für $j > 0$ und $\theta \in [-\pi,\pi]$, daß

$$h^{-1}\,(\theta + 2\pi j) > \frac{T}{\pi}\,(-\pi + 2\pi j) \geq T\,(-1+2j) \geq T,$$

und analog für $j < 0$, daß $h^{-1}\,(\theta + 2\pi j) < -T$. Das heißt aber, daß die Summe auf der linken Seite von (2.27) nur aus dem Term $j = 0$ besteht und wenn wir nun noch θ durch $h\xi$ ersetzen, dann erhalten wir, daß

$$\widehat{f}(\xi) = h \sum_{j\in\mathbb{Z}} f(hj)\, e^{-ijh\xi}$$

und somit ergibt sich, da $T < \pi/h$ und f T–bandbeschränkt ist,

$$\begin{aligned}
f(x) &= \frac{1}{2\pi}\int_{\mathbb{R}} \widehat{f}(\theta)\, e^{ix\theta}\,d\theta = \frac{1}{2\pi}\int_{-T}^{T} \widehat{f}(\theta)\, e^{ix\theta}\,d\theta = \frac{h}{2\pi}\int_{-T}^{T} \sum_{j\in\mathbb{Z}} f(hj)\, e^{i(x-jh)\theta}\,d\theta \\
&= \frac{h}{2\pi}\sum_{j\in\mathbb{Z}} f(hj)\int_{-\pi/h}^{\pi/h} e^{i(x-jh)\theta}\,d\theta = \frac{h}{2\pi}\sum_{j\in\mathbb{Z}} f(hj)\left[\frac{e^{i(x-jh)\theta}}{i(x-jh)}\Big|_{\theta=-\pi/h}^{\pi/h}\right] \\
&= \sum_{j\in\mathbb{Z}} f(hj)\,\underbrace{\frac{e^{i(x-jh)\pi/h} - e^{-i(x-jh)\pi/h}}{2i}}_{=\sin\pi(x/h-j)}\,\underbrace{\frac{h}{\pi}\frac{1}{(x-jh)}}_{=(\pi(x/h-j))^{-1}} \\
&= \sum_{j\in\mathbb{Z}} f(hj)\,\frac{\sin\pi\,(x/h-j)}{\pi\,(x/h-j)} = (S_h f * \mathrm{sinc}\,)\,(\cdot/h),
\end{aligned}$$

was damit (2.24) liefert. □

Der Clou im Beweis von Satz 2.16 ist die Betrachtung der Identität

$$\sum_{k\in\mathbb{Z}} \widehat{f}\Big(h^{-1}(\theta + 2\pi k)\Big) = h\sum_{k\in\mathbb{Z}} f(hk)\, e^{-ik\theta}, \qquad \xi \in \mathbb{T}, \tag{2.28}$$

die die **Periodisierung** der Fouriertransformierten einer Funktion mit der Fouriertransformierten der Abtastfolge verknüpft, und zwar *ohne* jedwede Forderung von Bandbeschränktheit[55] oder hinreichend feine Abtastung. Ist nun h so groß, daß die Funktion $\widehat{f}\,(h^{-1}\cdot)$ über das Intervall $[-\pi,\pi]$ "hinausragt", dann wird die Funktion durch die überlappenden Teile periodisch "verschmiert", siehe Abb. 2.2. Aus dieser Funktion läßt sich $\widehat{f}$ natürlich nicht mehr rekonstruieren. Aber es ist sogar noch schlimmer: Durch die Überlagerung von Frequenzen die eigentlich nichts miteinander zu tun haben und nun modulo 2π betrachtet werden, kommt es bei der Rekonstruktion des Signals im Falle von Unterabtastung zu sehr unerwünschten Effekten, die man als **Aliasing** bezeichnet.

[55] Außer man möchte, daß die Summe auf der linken Seite von (2.28) existiert, dann sollte vielleicht doch zumindest $\widehat{f} \in L_1(\mathbb{R})$ sein.

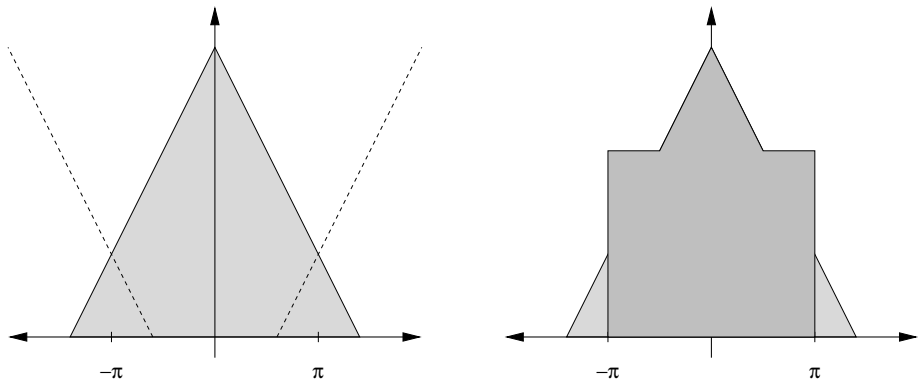


Abbildung 2.2: Periodisierung einer Funktion mit "zu großem" Träger, die schraffiert unterlegte Funktion wird dann periodisch fortgesetzt. Natürlich ist es nicht möglich, die Funktion links *eindeutig* aus der Periodisierung zu rekonstruieren.

Anders wird die Sache, wenn man h so wählt, daß der Träger von $\widehat{f}$ *ganz* in $[-\pi, \pi]$ liegt, denn dann gibt es keine Überlappung und die 2π–periodisch Fourierreihe $(S_h f)^\wedge$ ist gleich der Periodisierung von $\widehat{f}$. Und was dann noch kam war reine Rechnerei ...

2.4 Filter

Jetzt aber endlich zu den Objekten der Begierde der digitalen Signalen, nämlich den Filtern, denn die sind es ja letztendlich, die gerade den Job der Verarbeitung der Signale erledigen sollen. Ein **Filter** ist nun nichts anderes als eine Abbildung von einem Signalraum in einen anderen; manchmal spricht man auch von einem **System**, das ein diskretes oder kontinuierliches Sigal in ein Signal gleicher Bauart (also wieder diskret oder kontinuierlich) umsetzt. Wir wollen uns hier, schon aufgrund der "Praxisnähe" mit *diskreten* Filtern befassen, also Filtern, die diskrete Signale in diskrete Signale überführen.

Definition 2.18 (Filter und Filtertypen) *Ein **Filter** F ist ein Operator, der $c \in \ell(\mathbb{Z})$ in $Fc \in \ell(\mathbb{Z})$ abbildet. Man nennt so einen Filter*

1. ***energieerhaltender Filter**, wenn $F : \ell_2(\mathbb{Z}) \to \ell_2(\mathbb{Z})$ und*

$$1 = \|F\|_2 := \sup_{\|c\|_2 = 1} \|Fc\|_2$$

   ist.

2. ***linearer Filter**, wenn*

$$F[\alpha c + \beta c'] = \alpha Fc + \beta Fc', \qquad \alpha, \beta \in \mathbb{R}, \quad c, c' \in \ell(\mathbb{Z}).$$

3. ***zeitinvarianter Filter**, wenn der Operator **stationär** ist, d.h. seine Handlung nicht vom jeweiligen Zeitpunkt abhängt*[56]*:*

$$F\left[c\left(\cdot+k\right)\right]=\left[Fc\right]\left(\cdot+k\right),\qquad c\in\ell(\mathbb{Z}),\quad k\in\mathbb{Z}.$$

Mit Hilfe des diskreten Translationsoperators τ, *definiert durch* $\tau c=\tau_1 c=c(\cdot+1)$ *kann man das auch recht komfortabel als die Kommutativität*

$$\tau F=F\tau\qquad\textit{bzw.}\qquad\tau_k F=F\tau_k,\quad\tau_k=\tau^k,$$

schreiben.

4. *Ein Filter heißt **kausal**, wenn das Ergebnis zum Zeitpunkt k nur von den Eingaben $c(j)$, $j\le k$, abhängt, der Filter kann also nicht in die Zukunft sehen.*

Lineare und zeitinvariante Filter, nach (Kammeyer & Kroschel, 1998) **LTI–Filter**, in (Hamming, 1989) direkt als **digitaler Filter** eingeführt, sind die richtig schön strukturierten Filter. Sie haben nämlich die Eigenschaft, daß sie nur von der **Impulsantwort**

$$f:=F\delta,\qquad\text{also}\qquad f(k)=\left[F\delta\right](k),\quad k\in\mathbb{Z},$$

abhängen und das auch noch auf ziemlich strukturierte Art und Weise. Da man jedes Signal $c\in\ell(\mathbb{Z})$ formal als

$$c=\sum_{k\in\mathbb{Z}}c(k)\,\delta(\cdot-k)=\sum_{k\in\mathbb{Z}}c(k)\,\tau_{-k}\delta$$

schreiben kann, ist wegen der Linearität und der Zeitinvarianz

$$\begin{aligned}Fc&=F\left[\sum_{k\in\mathbb{Z}}c(k)\,\tau_{-k}\delta\right]=\sum_{k\in\mathbb{Z}}c(k)\,F\left[\tau_{-k}\delta\right]=\sum_{k\in\mathbb{Z}}c(k)\,\tau_{-k}F\delta=\sum_{k\in\mathbb{Z}}c(k)\,\tau_{-k}f\\&=\sum_{k\in\mathbb{Z}}c(k)\,f\left(\cdot-k\right)=c*f,\end{aligned}$$

weswegen lineare, zeitinvariante Filter immer *Faltungen* mit der Impulsantwort sind. Damit wissen wir aber auch, wie so ein Filter oder System im *Frequenzbereich* funktioniert, nämlich ganz einfach als

$$\left(Fc\right)^{\wedge}\left(\xi\right)=\left(f*c\right)^{\wedge}\left(\xi\right)=\widehat{f}(\xi)\,\widehat{c}(\xi).\tag{2.29}$$

Im Frequenzbereich ist also ein LTI–Filter immer linear, also eine einfache Multiplikation zwischen der Fouriertransformierten des Filters, der sogenannten **Transferfunktion** des Filters und der Fouriertransformierten des Signals. Und oftmals werden auch Eigenschaften des Filters, wie beispielsweise Hoch-, Tief- oder Bandpass über deren Fouriertransformierte gefordert.

[56] Einfachstes Beispiel: Wenn man den CD–Player eine Stunde später mit derselben CD anwirft, dann sollte auch dieselbe Musik rauskommen.

Was sind aber nun Filter, die man wirklich in der Praxis realisieren kann? Nun, zuerst sollten diese Filter natürlich einmal kausal sein, das heißt, daß für $k < 0$ der Impuls δ keinen Beitrag liefern darf, daß also

$$0 = [F\delta](k) = f(k), \qquad k < 0,$$

ist. Ein Filter mit der Eigenschaft, daß $f(k) = 0$, $k > 0$, heißt im Übrigen **antikausal**[57]. Auch nichtkausale Filter können Sinn machen, und zwar dann, wenn das Signal zeitbeschränkt ist[58], gespeichert und in beide Richtungen abgearbeitet werden kann.

Von jetzt an, soll die Bezeichnung "digitaler Filter" immer für einen LTI–Filter stehen. Bevor wir uns mit der praktischen Realisierung befassen, noch zwei Begriffe.

Definition 2.19 (Weitere Filtertypen) *Ein digitaler Filter F heißt*

1. ***FIR–Filter***[59] *oder* Filter mit endlicher Impulsantwort, *wenn die Impulsantwort endlichen Träger hat, wenn also $F\delta \in \ell_{00}(\mathbb{Z})$ liegt.*

2. IIR–Filter[60] *oder* Filter mit unendlicher Impulsantwort, *wenn die Impulsantwort* keinen *endlichen Träger hat.*

Jetzt können wir uns überlegen, wie man einen Digitalfilter, genauer einen *kausalen FIR–Filter* in der Praxis realisiert. Dazu braucht man "nur" drei Schaltglieder, nämlich

- einen **Multiplizierer**, der eine Zahl mit einer *festen* Konstanten c multipliziert,
- einen **Addierer**, der zwei Zahlen miteinander addiert,
- einen **Verzögerer**, der einen Wert *eine* Zeiteinheit lang speichern kann.

Die Symbole für diese drei Bausteine sind in Abb. 2.3 dargestellt. Da ein Verzögerer, angewandt auf ein Signal $c \in \ell(\mathbb{Z})$ das Signal $\tau_{-1}c$ liefert und somit eine Kette von k Verzögerern das Signal $\tau_{-k}c$, kann man einen kausalen FIR–Filter F, der ja die Form

$$Fc = f * c = \sum_{k\in\mathbb{Z}} f(k)\, c\,(\cdot - k) = \sum_{k=0}^{N} f(k)\, \tau_{-k}c, \qquad \operatorname{supp} f \subseteq [0, N],$$

hat, mit Hilfe von N unserer Bausteine darstellen: Die Werte $\tau_{-k}c$, $k = 0, \ldots, N$, werden von einer Kette von $N + 1$ Verzögerern abgegriffen, jeweils durch Multiplizierer mit den Werten $f(k)$ gewichtet und dann mit N Summierern aufsummiert. Diese Vorgehensweise ist in Abb. 2.4 dargestellt. Mit Hilfe von M

[57] So ein Filter wird dadurch realisiert, daß man die Zeit rückwärts laufen lässt.

[58] Beispielsweise bei Bildern, die sind zwar ausdehnungsbeschränkt, aber das läuft hier aufs gleiche raus.

[59] **F**inite **I**mpulse **R**esponse.

[60] **I**nfinite **I**mpulse **R**esponse

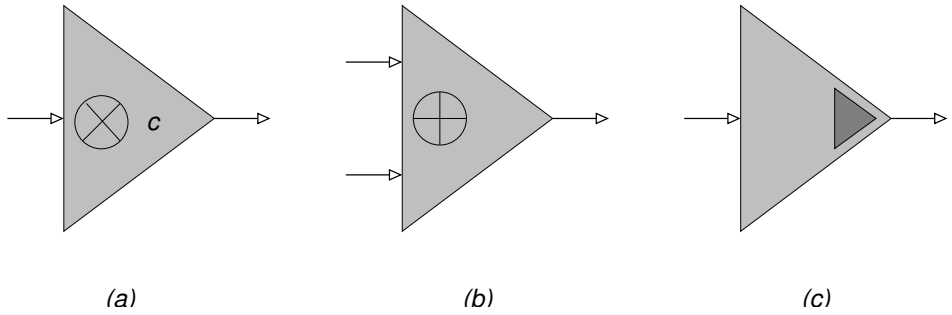


Abbildung 2.3: Symbolische Darstellung der drei Bausteine: Multiplizierer (a), Addierer (b) und Verzögerer (c).

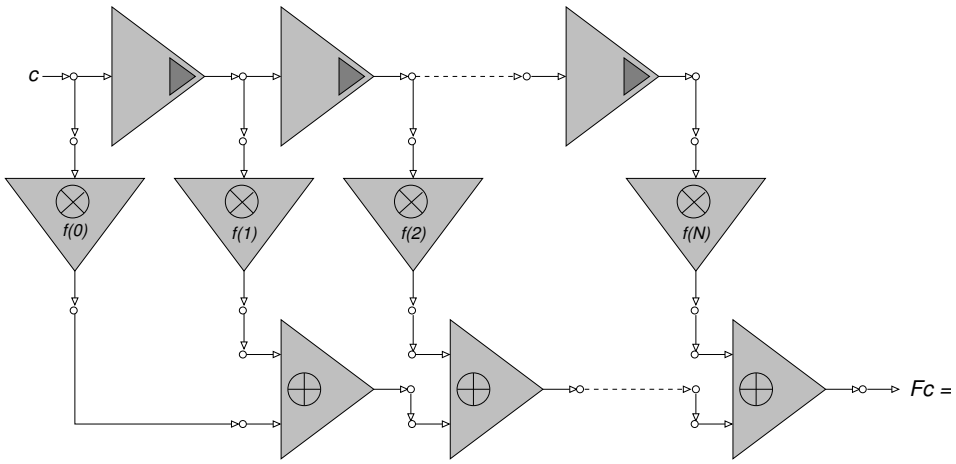


Abbildung 2.4: Ein FIR–Filter als Kaskade der Bausteine aus Abb. 2.3. Die Verzögerer sorgen für die Translationen, die Multiplizierer für die Gewichtung und die Addierer summieren den ganzen Kram auf.

Speichereinheiten könnte man übrigens auch einen Filter realisieren, dessen „Taps“ in $[-M, N]$ liegen, der aber das Ergebnis um M Zeiteinheiten *verzögert* ausgibt – womit sich dann *alle* FIR–Filter praktisch realisieren lassen. Achtung:

> Jeder Filter hat wegen der Verzögerer eine gewisse Laufzeit, weswegen in der Realität alle Filter eine gewisse Trägheit besitzen.

Diesen Effekt kann man heute bei Live–Übertragungen von Fußballspielen beobachten, bei denen der eine oder andere Torschrei schon einmal um zwei bis drei Sekunden versetzt sein kann.

Jetzt wollen wir uns mal ein einfaches Beispiel aus (Hamming, 1989, Sec. 3.8) für Filterdesign ansehen und zwar einen symmetrischen, *nichtkausalen* FIR–Filter mit

$$f(0) = a, \qquad f(\pm 1) = b, \qquad f(\pm 2) = c.$$

Um die Symmetrie $f(k) = f(-k)$ und somit eine reelle Transferfunktion zu erhalten, ist es vorteilhaft, besser auf die Kausalität zu verzichten.

Übung 2.7 Zeigen Sie: Ist $f \in \ell_{00}(\mathbb{Z})$ der reelle Koeffizientenvektor eines Filters F, dann ist die Transferfunktion $\widehat{f}$ genau dann reell, wenn f symmetrisch ist und genau dann rein imaginär, wenn f antisymmetrisch ist, d.h. $f(k) = -f(-k)$. ◊

Das *Design* des Filters wird nun meistens im *Frequenzbereich* festgelegt, das heißt, man stellt Forderungen an die Transferfunktion[61] $\widehat{f}(\xi)$. Die Forderung hier soll sein, daß der Filter im niederfrequenten Bereich voll durchlässig ist und im hochfrequenten Bereich sperrt, also

$$\widehat{f}(0) = 1, \qquad \widehat{f}(\pi) = 0. \tag{2.30}$$

Mit

$$\widehat{f}(\xi) = a + b\left(e^{-i\xi} + e^{i\xi}\right) + c\left(e^{-2i\xi} + e^{2i\xi}\right) = a + 2b\cos\xi + 2c\cos 2\xi$$

heißt (2.30), daß

$$a + 2b + 2c = 1, \qquad \text{und} \qquad a - 2b + 2c = 0,$$

also $b = \frac{1}{4}$ und $a = \frac{1}{2} - 2c$, also

$$\begin{aligned}
\widehat{f}(\xi) &= \frac{1}{2} - 2c + \frac{1}{2}\cos\xi + 2c\,\cos 2\xi = \frac{1}{2} - 2c + \frac{1}{2}\cos\xi + 2c\left(2\cos^2\xi - 1\right) \\
&= \frac{1}{2} - 4c + \frac{1}{2}\,\cos\xi + 4c\,\cos^2\xi = 4c\left[\cos^2\xi + \frac{1}{8c}\cos\xi + \frac{1}{8c} - 1\right] \\
&= 4\,(\cos\xi + 1)\left(c\,\cos\xi + \frac{1}{8} - c\right).
\end{aligned}$$

Der erste Faktor, $\cos\xi + 1$, sorgt dabei für die Nullstelle an $\xi = \pi$, der zweite degeneriert zu einer Konstanten, wenn $c = 0$ ist, also wird mit Sicherheit der Fall $c = 0$ besonders sein[62] Einige Beispiele für die Transferfunktion dieser Filter mit variierendem Parameter c finden sich in Abb. 2.5.

Man kann aber diesen *Designparameter* c nun auch so wählen, daß der resultierende Filter weitere Eigenschaften besitzt, beispielsweise:

- *Erhaltung eine weiteren Frequenz* ω, d.h.

$$1 = \widehat{f}(\omega) = 4\,(\cos\omega + 1)\left(c\,\cos\omega + \frac{1}{8} - c\right)$$

  also

$$c = \left(\frac{1}{4\,(\cos\omega + 1)} - \frac{1}{8}\right) / \,(\cos\omega - 1) = -\frac{1}{8\,(\cos\omega + 1)}$$

  was für $\omega \neq \pi$ immer lösbar ist.

[61] Diese ist eine 2π–periodische Funktion auf $\mathbb{R}$, oder eben auch eine Funktion auf $\mathbb{T}$.

[62] Kein Wunder, denn dann hat ja das trigonometrische Polynom $\widehat{f}$ Grad 1 und nicht Grad 2.

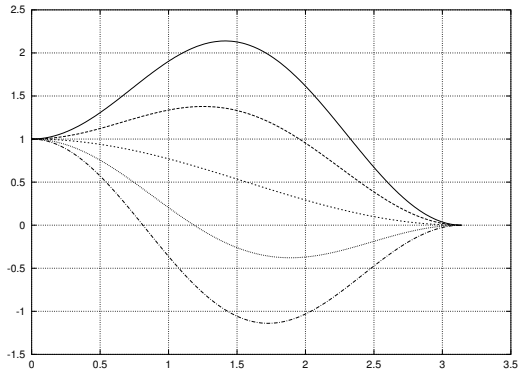


Abbildung 2.5: Beispiele für Transferfunktionen mit $c = -.4, -.2, 0, .2, .4$ (von oben nach unten). Besonders interessant ist der Fall $c = 0$, denn da bewegt sich der Filter nur zwischen 1 und 0 und stellt eine sogenannte **sigmoidale** Funktion dar.

- *So flach wie möglich an* $\xi = 0$, d.h., es verschwinden so viele Ableitungen von $\widehat{f}$ wie möglich an $\xi = 0$, so daß sich der Filter "soweit wie möglich" einer "charakteristischen Funktion" annähert. Da

$$\frac{d}{d\xi}\widehat{f}(\xi) = -2b\,\sin\xi - 4c\sin 2\xi$$

an $\xi = 0$ *immer* verschwindet, können wir also fordern, daß

$$0 = \frac{d^2}{d\xi^2}\widehat{f}(0) = -2b\,\cos 0 - 8c\cos 0 \qquad \Rightarrow \qquad c = -\frac{1}{4}\,b = -\frac{1}{16}.$$

- *"Balanciertheit"* oder *Antisymmetrie um* $\frac{\pi}{2}$, d.h.

$$\widehat{f}\left(\frac{\pi}{2} - \xi\right) - \widehat{f}\left(\frac{\pi}{2}\right) = \widehat{f}\left(\frac{\pi}{2}\right) - \widehat{f}\left(\frac{\pi}{2} + \xi\right),$$

also

$$\frac{1}{2}\left[\widehat{f}\left(\frac{\pi}{2} - \xi\right) + \widehat{f}\left(\frac{\pi}{2} + \xi\right)\right] = \widehat{f}\left(\frac{\pi}{2}\right)$$

und somit insbesondere ($\xi = \frac{\pi}{2}$)

$$\frac{1}{2} = \widehat{f}\left(\frac{\pi}{2}\right) = a + 2b\,\underbrace{\cos\frac{\pi}{2}}_{=0} + 2c\,\underbrace{\cos\pi}_{=-1} \qquad \Rightarrow \qquad a = \frac{1}{2} + 2c$$

was zusammen mit $a = \frac{1}{2} - 2c$ die Bedingungen $a = \frac{1}{2}$ und $c = 0$ liefert.

Diese drei Filter sind in Abb. 2.6 dargestellt.

Auch wenn dieses Beispiel sehr einfach ist, zeigt es bereits, wie beim Filterdesign vorgegangen wird: Man gibt sich normalerweise das Verhalten des Filters im Frequenzbereich vor und bestimmt dann die Koeffizienten des Filters so, daß diese Verfahren realisiert oder zumindest approximiert wird. Und das zu ist einiges zu sagen.

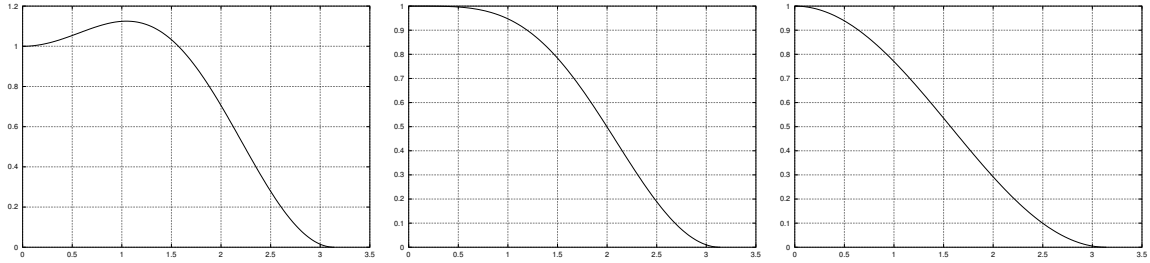


Abbildung 2.6: Die drei Filter zu den zusätzlichen Forderungen: Erhaltung der Frequenz $\frac{\pi}{2}$ (links), "Flachheit" an 0 (mitte) und Balanciertheit (rechts).

Bemerkung 2.20 (Filterdesign/Transferfunktion)

1. *Die Transferfunktion eines Filters, insbesondere eines nichttrivialen kausalen Filters, ist normalerweise eine* komplexwertige *Funktion. Deren* Realteil *liefert den* Gewichtungsfaktor *für die jeweilige Frequenz während ihr* Imaginärteil *die* Phasenverschiebung *angibt. Da letztere eigentlich nicht hörbar ist, gibt man oftmals bei Bandpassfiltern nur den Realteil des Filters vor.*

2. *Es mag zuerst etwas seltsam anmuten, daß der Frequenzgang des Filters immer nur von $-\pi$ bis π läuft, schließlich hätte man doch eigentlich gerne Filter, die beispielsweise hörbare Frequenzen von $10 - 20000$ Hz verarbeiten können. Und hier ist der Punkt, an dem der Abtastsatz 2.16 ins Spiel kommt: Der Filter operiert auf dem* diskreten *Signal, das eben durch hinreichend feines Abtasten gewonnen sein muß, und zwar so fein, daß die Fouriertransformierte des Signals auf das* relevante *Band $[-\pi, \pi]$ beschränkt wird.*

Das viel größere Problem besteht aber darin, daß die Transferfunktionen von FIR–Filtern nur eine sehr "kleine" Familie von Funktionen aus $L_2(\mathbb{T})$, oder was auch immer man als Filtermenge nehmen möchte, darstellen und man so bei weitem nicht jeden Filter, den man gerne hätte, auch wirlich exakt als FIR–Filter bekommt.

Definition 2.21 *Eine Funktion $f \in C^\infty(\mathbb{T})$ der Form*

$$f(x) = \sum_{|k| \le n} f_k \, e^{ikx}, \qquad f_k \in \mathbb{C}, \quad x \in \mathbb{T},$$

heißt ***trigonometrisches Polynom*** *der Ordnung n.*

Übung 2.8 Zeigen Sie: Jedes trigonometrische Polynom der Ordnung n läßt sich auch als

$$f(x) = a_0 + \sum_{k=1}^{n} \Big[a_k \cos^k x + b_k \sin^k x \Big], \qquad a_0, \ldots, a_n, \, b_1, \ldots, b_n \in \mathbb{C}$$

schreiben. ◊

Es ist nun klar, daß ein LTI–Filter F, dargestellt durch $f \in \ell(\mathbb{Z})$, genau dann ein FIR–Filter ist, wenn $f \in \ell_{00}(\mathbb{Z})$, also seine Fouriertransformation ein trigonometrisches Polynom ist. Eigentlich nicht so schlimm, denn die trigonometrischen Polynome sind *dicht* in $L_2(\mathbb{T})$ und wir wissen sogar, wie man die *beste Approximation* zu einer Transferfunktion $g = \widehat{f}$ berechnet: Man nimmt die Fourierreihe

$$\mathscr{F} g = \sum_{k\in\mathbb{Z}} g_k \, e^{ik\cdot}, \qquad g_k = \frac{1}{2\pi} \int_{\mathbb{T}} g(t) \, e^{-ikt}$$

und bildet deren n–te **Partialsumme**

$$\mathscr{F}_n g := \sum_{|k|\le N} g_k \, e^{ik\cdot} =: \widehat{h},$$

was uns auch schon die Koeffizienten unseres Filters h liefert. Und $\widehat{h}$ ist sogar dasjenige trigonometrische Polynom der Ordnung n, das in der Norm von $L_2(\mathbb{T})$ die Transferfunktion am besten annähert, also eine **Bestapproximation**, siehe z.B. (Sauer, 2002a). Mehr zu Fourierreihen findet man beispielsweise in (Hardy & Rogosinsky, 1956; Katznelson, 1976; Tolstov, 1962). Leider ist das aber auch nicht so einfach.

Beispiel 2.22 (Tiefpassfilter) *Wir wollen jetzt einen Filter F designen, der nur die tiefen Frequenzen durchläßt, sagen wir die untere Hälfte. Ideal wäre dieser Filter also durch*

$$\widehat{f} = \chi_{[-\pi/2,\pi/2]}$$

definiert – das ist natürlich kein trigonomentrisches Polynom und somit auch kein FIR–Filter, also auch nicht praktisch realisierbar. Um also näherungsweise Tiefpassfilter zu bekommen, bestimmen wir die Fourierkoeffizienten von $g = \widehat{f}$ und erhalten, daß

$$g_0 = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} dt = \frac{1}{2}$$

und, für $k \neq 0$,

$$\begin{aligned} g_k &= \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} e^{-ikt} dt = \frac{1}{2\pi} \, \frac{e^{-ikt}}{-ik} \Big|_{t=-\pi/2}^{\pi/2} = \frac{1}{2\pi} \frac{e^{-ik\pi/2} - e^{ik\pi/2}}{-ik} = \frac{1}{k\pi} \sin \frac{k}{2}\pi \\ &= \begin{cases} 0, & k \in 2\mathbb{Z}, \\ \frac{(-1)^{(k-1)/2}}{k\pi}, & k \in 2\mathbb{Z}+1, \end{cases} \end{aligned}$$

das heißt,

$$g_{2k+1} = \frac{(-1)^k}{k\pi}, \qquad g_{2k+2} = 0, \qquad k \in \mathbb{N}_0.$$

Die Partialsumme der Ordnung $n = 2m + 1$ *ist also*

$$\begin{aligned} h_n(\xi) &= \frac{1}{2} + \sum_{k=0}^{m} \frac{(-1)^k}{(2k+1)\pi} \underbrace{\left[e^{i(2k+1)\xi} + e^{-i(2k+1)x}\right]}_{2\cos(2k+1)\xi} \\ &= \frac{1}{2} + \sum_{k=0}^{m} (-1)^k \frac{2}{(2k+1)\pi} \cos(2k+1)\xi, \end{aligned}$$

was uns den näherungsweisen FIR–Filter liefert. Nur ist es mit der Approximationsqualität nicht so weit her, wie uns Abb. 2.7 zeigt.

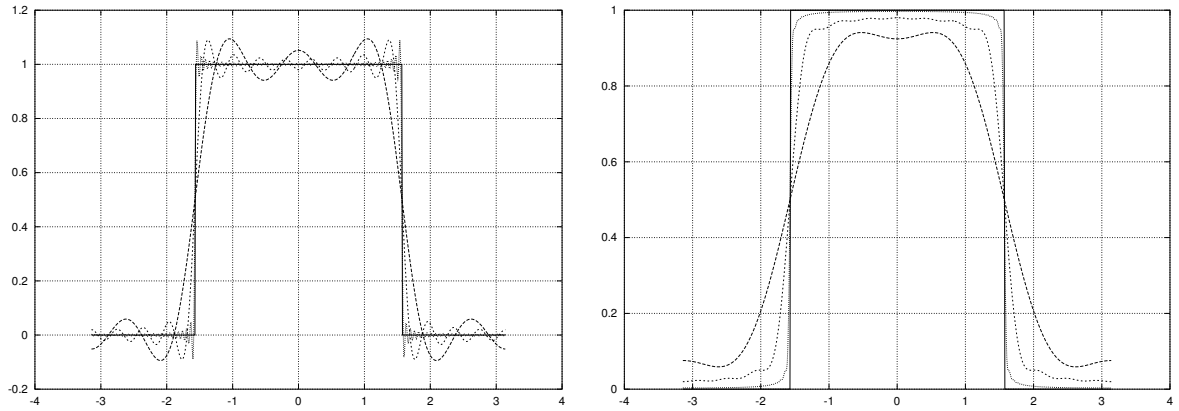


Abbildung 2.7: *Links:* Approximation des Bandpassfilters durch Partialsummen für $n = 5, 15, 100$ (Werte eher zufällig). Man beachte, daß die "Überschiesser" nur schmaler, nicht aber kleiner werden.

Rechts: Approximation durch ein anderes Approximationsverfahren, nämlich die *Fejérschen Mittel*. Diese haben zwar eine größere Abweichung vom Bandpass als die Partialsummen, verzichten dafür aber auf wilde Oszillationen.

Abb. 2.7 zeigt das sogenannte **Gibbs–Phänomen**[63]: Die Partialsummen zu "steilflankigen" Filtern liefern immer Überschwingphänomene, die auch mit steigender Approximationsqualität nicht verschwinden. Dies macht FIR–Filter für die Konstruktion von Bandpassfiltern recht ungeeignet. Außerdem ist die Qualität, mit der man nichtglatte Funktionen durch trigonometrische Polynome approximieren kann, die sogenannte *Approximationsordnung*, ebenfalls stark eingeschränkt, siehe z.B. (DeVore & Lorentz, 1993; Lorentz, 1966; Mhaskar & Pai, 2000; Sauer, 2002a). Man kann, wie das rechte Bild in Abb. 2.7 zeigt, dem Gibbs–Phänomen durch Wahl eines geeigneten Approximationsverfahrens entgehen[64], das "gestalterhaltende[65]" Eigenschaften besitzt, dafür aber mit langsamerer Konvergenz

[63] Josiah Willard Gibbs, 1839–1903, Professor für mathematische Physik in Yale; das Gibbs–Phänomen wurde aber angeblich nicht von ihm entdeckt.

[64] Anstelle der n–ten Partialsumme betrachten man hier das *arithmetische Mittel* der Partialsummen der Ordnungen $0, 1, \ldots, n$.

[65] "Shape preserving".

bezahlt – für Details siehe z.B. nochmals (DeVore & Lorentz, 1993; Lorentz, 1966; Mhaskar & Pai, 2000; Sauer, 2002a).

2.5 Filter für Bilder

Jetzt betrachten wir wirklich einmal Bilder als *zweidimensionale* Signale, also Funktionen von $\mathbb{R}^2 \to \mathbb{R}$ bzw. $\mathbb{Z}^2 \to \mathbb{R}$. Formalisieren wir kurz, womit wir es hier zu tun haben.

Definition 2.23 (Signalklassen) *Mit $\ell(\mathbb{Z}^2)$ bezeichnen wir die Menge aller Signale der Form*[66]

$$c = \left(c(\alpha) \ : \ \alpha \in \mathbb{Z}^2\right) = (c(j,k) \ : \ j,k \in \mathbb{Z})$$

unter Verwendung der Normen

$$\|c\|_p := \left(\sum_{\alpha\in\mathbb{Z}^2} |c(\alpha)|^p\right)^{1/p}, \qquad \|c\|_\infty = \sup_{\alpha\in\mathbb{Z}^2} |c(\alpha)|\,.$$

Analog sind für Funktionen $f : \mathbb{R}^2 \to \mathbb{R}$ die Normen als

$$\|f\|_p = \left(\int_{\mathbb{R}^2} |f(t)|^p \ dt\right)^{1/p}, \qquad \|f\|_\infty = \sup_{x\in\mathbb{R}^2} |f(x)|$$

definiert.

Auch eine Fouriertransformation gibt es in $\mathbb{R}^s$, und zwar, für $\xi \in \mathbb{R}^2$,

$$\widehat{f}(\xi) = \int_{\mathbb{R}^2} f(t) e^{-i\xi^T t}\,dt = \int_{\mathbb{R}^2} f(t) e^{-i(\xi_1 t_1 + \xi_2 t_2)}\,dt = \int_{\mathbb{R}^2} f(t)\, e^{-i\xi_1 t_1}\, e^{-i\xi_2 t_2}\,dt.$$

Und wo es eine Fouriertransformation gibt, da ist auch eine Faltung nicht weit, und die sieht auch noch praktisch ganz genauso aus wie die Faltung in einer Variablen:

$$f * g = \int_{\mathbb{R}^2} f(\cdot - t)\, g(t)\,dt.$$

Übung 2.9 Zeigen Sie, daß auch in zwei Variablen die Identität

$$(f * g)^\wedge(\xi) = \widehat{f}(\xi)\,\widehat{g}(\xi)$$

gilt. ◊

Übung 2.10 Bestimmen Sie für Fouriertransformationen in zwei Variablen

- die inverse Fouriertransformation,
- die Perseval/Plancherel–Formel,

[66] Den Index $\alpha = (\alpha_1, \alpha_2) \in \mathbb{Z}^2$ bezeichnet man auch als *Multiindex*.

- die Poissonsche Summenformel.

◊

Auch LTI–Filter sind damit kein Problem[67], die Impulsantwort ist jetzt halt ein $f \in \ell(\mathbb{Z}^2)$ und die Filterung ergibt sich als

$$Fc = f * c = \sum_{\alpha \in \mathbb{Z}^2} f(\cdot - \alpha)\, c(\alpha), \tag{2.31}$$

was für einen FIR–Filter mit $f \in \ell_{00}(\mathbb{Z}^2)$ auch wieder "nur" eine endliche Summe ist. Besonders einfach wird die Filterung, wenn f ein *Tensorprodukt* ist, das heißt, wenn

$$f = f_1 \otimes f_2, \qquad \text{d.h.} \qquad f(\alpha) = f_1(\alpha_1)\; f_2(\alpha_2), \tag{2.32}$$

ist, denn dann ist

$$\begin{aligned} f * c &= \sum_{\alpha \in \mathbb{Z}^2} f(\cdot - \alpha)\, c(\alpha) = \sum_{\alpha_1, \alpha_2 \in \mathbb{Z}} f_1(\cdot - \alpha_1)\, f_2(\cdot - \alpha_2)\; c(\alpha_1, \alpha_2) \\ &= \sum_{\alpha_1 \in \mathbb{Z}} f_1(\cdot - \alpha_1) \sum_{\alpha_2 \in \mathbb{Z}} f_2(\cdot - \alpha_2)\; c(\alpha_1, \alpha_2) = \sum_{\alpha_1 \in \mathbb{Z}} f_1(\cdot - \alpha_1)\; (f_2 * c(\alpha_1, \cdot)) \end{aligned} \tag{2.33}$$

Das liefert uns bereits ein Schema, wie wir so einen Filter anwenden: Für jedes feste α_1 filtern wir die "Zeile" $c(\alpha_1, \cdot)$ mit f_2 und stecken das Ergebnis dann in den Filter f_1. Schematisch sieht das dann wie folgt aus:

$$\begin{array}{cccccc} \ddots & \vdots & \vdots & \ddots & & \vdots \\ \dots & c(0,0) & c(0,1) & \dots & \to \; f_2 * c(0,\cdot) = & c'(0) \\ \dots & c(1,0) & c(1,1) & \dots & \to \; f_2 * c(1,\cdot) = & c'(1) \\ \ddots & \vdots & \vdots & \ddots & & \vdots \\ & & & & & \downarrow \\ & & & & & f_1 * c' \\ & & & & & \downarrow \\ & & & & & f * c \end{array}$$

Hat nun ein Filter F bzw. seine Impulsantwort f Tensorproduktstruktur, dann ist

$$\begin{aligned} \widehat{f(\xi)} &= \sum_{\alpha \in \mathbb{Z}^2} f(\alpha) e^{-i\alpha^T \xi} = \sum_{\alpha \in \mathbb{Z}^2} f(\alpha) e^{-i(\alpha_1 \xi_1 + \alpha_2 \xi_2)} = \sum_{\alpha \in \mathbb{Z}^2} \underbrace{f(\alpha)}_{= f(\alpha_1)\, f(\alpha_2)} e^{-i\alpha_1 \xi_1}\, e^{-i\alpha_2 \xi_2} \\ &= \left(\sum_{\alpha_1 \in \mathbb{Z}} f(\alpha_1)\; e^{-i\alpha_1 \xi_1} \right) \left(\sum_{\alpha_2 \in \mathbb{Z}} f(\alpha_2)\; e^{-i\alpha_2 \xi_2} \right), \end{aligned}$$

das heisst,

$$\widehat{f(\xi)} = \widehat{f_1}(\xi_1)\, \widehat{f_2}(\xi_2).$$

Anders gesagt: Die Faltung mit Tensorproduktfiltern ist besonders einfach!

[67] Für irgendwas müssen unsere Vorarbeiten ja gut sein.

Es gibt in der (medizinischen) Bildverarbeitung, siehe z.B. (Handels, 2000) eine ganze Menge von "Standardfiltern", die wir uns kurz ansehen wollen. Dabei ist es so, daß diese Filter oftmals *kontinuierlich*, also für Funktionen[68] $\phi : \mathbb{R}^2 \to \mathbb{R}$ konzipiert sind und für diskrete Daten einfach diskretisiert werden.

Der erste Filter ist der *Mittelwertfilter*

$$f = \frac{1}{|\Omega|}\chi_\Omega, \qquad \Omega \subset \mathbb{R}^2,$$

wobei Ω eine *kompakte* Teilmenge sein sollte. Die Filterung

$$\phi \mapsto f{*}\phi(x) = \frac{1}{|\Omega|}\int_{\mathbb{R}^2} \chi_\Omega(t)\phi\left(x-t\right)\,dt = \frac{1}{|\Omega|}\int_{\Omega} \phi\left(x-t\right)\,dt = \frac{1}{|\Omega|}\int_{x+\Omega} \phi(t)\,dt$$

ordnet also an jeder Stelle x der Funktion den Mittelwert über die Menge $x+\Omega$ zu. Die Diskretisierung dieses Filters ist einfach: Man nimmt einfach $f = S_h\chi_\Omega$ mit passender Abtastung h und normalisiert den Filter, indem man durch die Anzahl der in Ω enthaltenen Abtastpunkt teilt.

Ein wesentlicher Vorteil von Mittelungsfiltern ist, daß sie Rauschen unterdrücken. Rauschen ist ein unangenehmer Bestandteil gemessener Daten, der sich normalerweise nur stochastisch beschreiben lässt. Das Standardmodell für verrauschte Daten ist

$$\phi(x) = \psi(x) + \epsilon(x),$$

wobei ψ die eigentlichen Daten sind und ϵ das Rauschen. Eine allgemein übliche[69] Annahme ist, daß das Rauschen *mittelwertfrei* ist, das heißt, daß

$$E(\epsilon) = \int_{\mathbb{R}^2} \epsilon(x)\,dx = 0.$$

Bei der Filterung mit einem Mittelwerfilter erhält man dann also

$$F\phi(x) = \frac{1}{|\Omega|}\int_{x+\Omega} \phi(t)\,dt + \frac{1}{|\Omega|}\int_{x+\Omega} \epsilon(t)\,dt$$

und das mittelwertfreie Rauschen sollte zu einer geringeren Störung führen. Hier sieht man auch schon das Problem mit solchen Filtern: Der Träger sollte klein und "symmetrisch" genug sein, daß $F\psi \sim \psi$, was wegen

$$\phi(x) = \lim_{h\to 0^+} \frac{1}{h|\Omega|}\int_{x+h\Omega} \phi(t)\,dt, \qquad \phi \in L_1\left(\mathbb{R}^2\right)$$

zu schaffen ist, aber auch groß genug, damit das Rauschen auch wirklich ausgemittelt wird.

[68] Wir drücken uns hier vor der Festlegung von Eigenschaften wie Stetigkeit, Differenzierbarkeit oder Integrierbarkeit.

[69] Und in vielen Fällen genauso plausible wie unrealistische.

Will man es etwas vornehmer haben, dann kann man die charakteristische Funktion beispielsweise durch den *Gaußkern*

$$f(x) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{\|x\|_2^2}{2\sigma^2}}, \qquad \sigma > 0$$

mit Standardabweichung σ ersetzen – so hat man sogar noch einen Parameter zur Verfügung. Für die Diskretisierung tastet man nun wieder f, genauer gesagt $f\chi_{[-N,N]^2}$ ab, wobei wir die charakteristische Funktion benötigen, um einen FIR–Filter zu erhalten, denn die Exponentialfunktion klingt zwar schnell ab, hat aber trotzdem unendlichen Träger. Das diskrete Gegenstück dazu sind die Binomialfilter

$$f(j,k) = 2^{-m-n}\binom{m}{j}\binom{n}{k}, \qquad \begin{matrix} j = 0,\dots,m, \\ k = 0,\dots,n, \end{matrix}$$

den man natürlich für gerade m, n auch zentrieren kann. Zum Beispiel hat der zentrierte $2,2$–Binomialfilter die Form

$$\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & \boxed{4} & 2 \\ 1 & 2 & 1 \end{pmatrix},$$

wobei der eingerahmte Wert die Stelle $(0,0)$ bezeichnet.

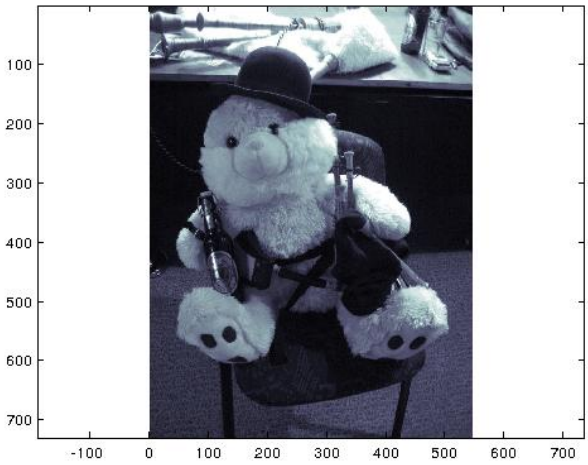


Abbildung 2.8: Ein Testbild mit Kanten und feinen Texturen und natürlich auch nur deswegen ausgewählt.

Gradientenfilter werden bei Bildern verwendet, um Konturen hervorzuheben, denn da, wo der Unterschied zwischen zwei benachbarten Punkten groß ist, ist auch die Steigung groß und bei einem richtigen Sprung sogar unendlich. Der

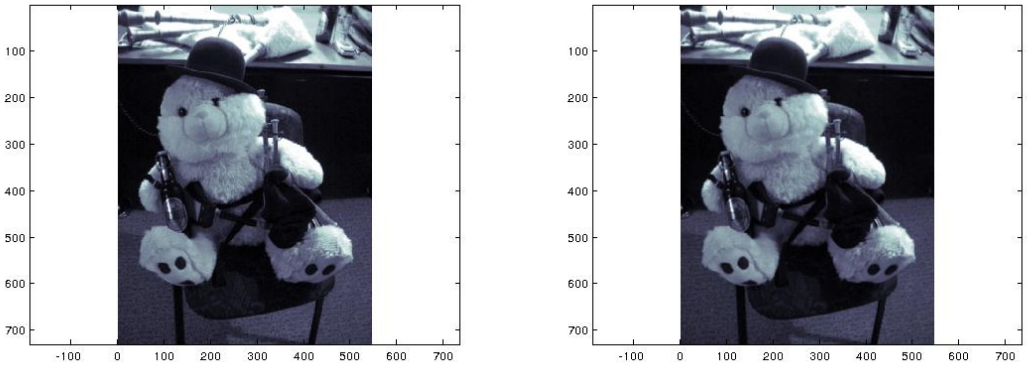


Abbildung 2.9: Anwendung der $2,2$– und $4,4$–Binomialfilter auf des Testbild aus Abb 2.8. Man sieht deutlich, daß die feinen Haarstrukturen verschwinden.

Gradient einer Funktion ist als

$$\nabla\phi = \begin{bmatrix} \dfrac{\partial}{\partial x}\phi(x,y) \\ \dfrac{\partial}{\partial y}\phi(x,y) \end{bmatrix}$$

definiert und läßt sich durch

$$\nabla c = \begin{bmatrix} \nabla_1 c \\ \nabla_2 c \end{bmatrix} = \begin{bmatrix} c(\cdot+1,\cdot) - c(\cdot,\cdot) \\ c(\cdot,\cdot+1) - c(\cdot,\cdot) \end{bmatrix}$$

diskretisieren, und die beiden partiellen Differenzen sind nun wieder durch Faltungen mit den Impulsantworten

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & \boxed{-1} & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} 0 & 1 & 0 \\ 0 & \boxed{-1} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

realisierbar.

Pure Gradientenverfahren sind allerdings sehr empfindlich gegen Rauschen, schließlich verbirgt sich ja hinter dem Gradienten ein Differenzenquotient mit Schrittweite h, wobei h die Abtastgenauigkeit ist, was Rauschen um einen Faktor h^{-1} verstärkt. Deswegen kombiniert man Gradienten gerne mit Mittelungsverfahren, z.B. dem Mittelungsoperator

$$f := \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & \boxed{1} & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Die Faltung $f * \nabla_j$ berechnet man am einfachsten über die z–Transformation als

$$\begin{aligned}(f * \nabla_1)^*(z) &= f^*(z)\,\nabla_1^*(z)\frac{1}{9}\sum_{\|\alpha\|_\infty\le 1} z^{-\alpha}\,(z_1-1)\\ &= \frac{1}{9}\sum_{\|\alpha\|_\infty\le 1} z^{-\alpha+\epsilon_1} - \sum_{\|\alpha\|_\infty\le 1} z^{-\alpha} = \frac{1}{9}\left(z_1^2 - z_1^{-1}\right)\left(z_2 + 1 + z_2^{-1}\right),\end{aligned}$$

und die gemittelten Gradientenfilter haben die Gestalt

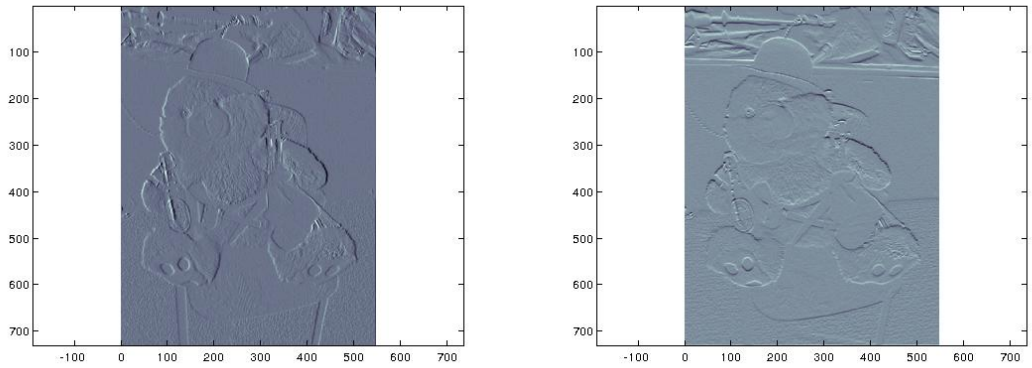


Abbildung 2.10: Gradientenfilter in x–Richtung *(links)* und y–Richtung *(rechts)* von Abb. 2.8. Man sieht sehr schön, daß teilweise horizontale, teilweise vertikale Kanten erfasst werden.

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

siehe Abb 2.10. Aus diesen Daten kann man nun sehr leicht die Kanten extrahieren, indem man beispielsweise die 1–Norm des geglätteten Gradienten bildet:

$$g = \|f * \nabla c\|_1 = |f * \nabla_1 c| + |f * \nabla_2 c|\,,$$

siehe Abb 2.11.

Und wenn wir schon bei Ableitungen sind, dann können wir auch den Laplaceoperator

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

verwenden; die zweiten partiellen Ableitungen kann durch symmetrische zweite Differenzen $c(\cdot+1) - 2c + c(\cdot-1)$ annähern und man erhält so den Filter

$$\begin{pmatrix} 0 & 1 & 0\\ 1 & \boxed{-4} & 1\\ 0 & 1 & 0\end{pmatrix}$$

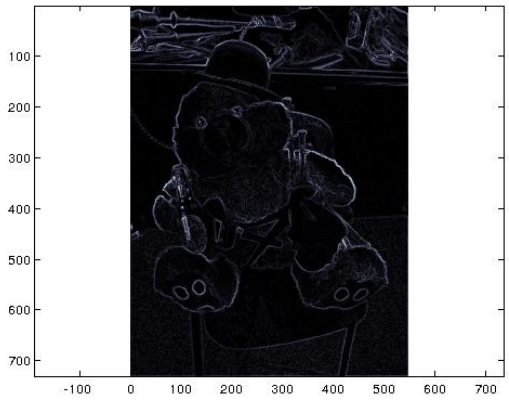


Abbildung 2.11: Kantenextraktion über die Norm des geglätteten Gradientenoperators.

als einfachste Diskretisierung des Laplaceoperators. Eine weitere Variante des Filters, der auch die Diagonalrichtungen berücksichtigt, ist

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

Wie in (Handels, 2000) ausgeführt ist, ist der Laplaceoperator besonders empfindlich gegen Rauscheffekte, weswegen man ihn meistens mit Glättungsfiltern, also Mittelung oder Gauß kombiniert.

Übung 2.11 Implementieren Sie geglättete Varianten des Laplacefilter und versuchen Sie, eine bessere Kantenerkennung zu erhalten. ◊

Bleibt noch ein Klassiker, nämlich der *Medianfilter*

$$Mc(j) = \text{Median}\{c(k) \ : \ k \in j + \Omega\}, \qquad \Omega \subset \mathbb{Z}^2,$$

der alle Werte $c\,(j + \Omega)$ der Größe nach sortiert und dann den Wert in mittlerer Position wählt – eben einen Median berechnet. Dieser Filter hat allerdings einige Eigenheiten: Er ist nicht linear und fordert einen recht hohen Rechenaufwand, auch wenn Sortieren zu den "einfacheren" Operationen gehört. Der Medianfilter hat den Vorteil, daß er völlig unbeeindruckt von Ausreißern arbeitet und im wesentlichen Kanten erhält.

Übung 2.12 Implementieren Sie einen lokalen Medianfilter `MedFilt (M,X)` für Bilder, dem man eine Maske `M` mit Werten $0, 1$ übergibt, die dann zur Bestimmung von Ω genutzt wird. ◊

2.6 Die FFT

In diesem Kapitel geht es um **den** fundamentalen Algorithmus der Signal- und auch der Bildverarbeitung, nämlich die **schnelle Fouriertransformation** oder

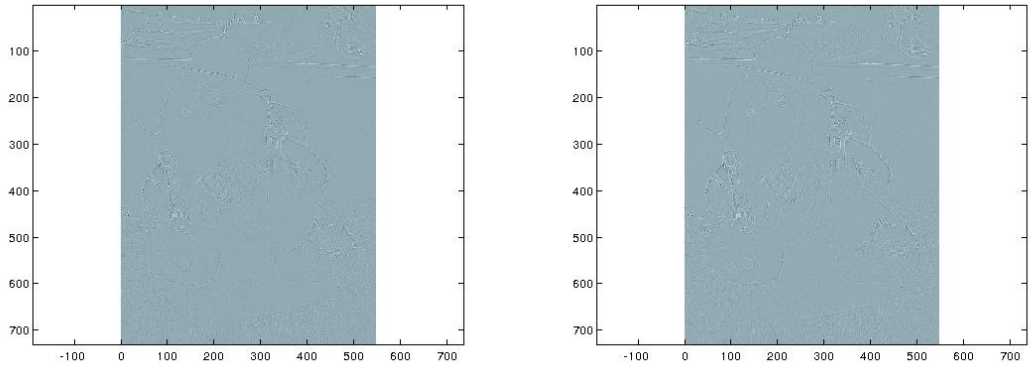


Abbildung 2.12: Anwendung der beiden Laplacefilter. Wie Sie sehen sehen Sie nichts oder zumindest nicht viel - das ist die schlechte Auflösung des Laplacefilter. Wenn man rechts genau hinsieht, erkennt man auch noch ziemlich viel Rauschen vom Boden. Warum das ist, erkennt man, wenn sich Abb 2.13 ansieht, aus der das starke Rauschen des Bildes sichtbar wird und in dem man die Konturen nur an wenigen Stellen und auch dort nur mit großer Mühe erkennen kann.

FFT ("*F*ast *F*ourier *T*ransform"). Als im allgemeinen Milleniumsfieber 2000 die Liste der 10 bedeutendsten und einflußreichsten Verfahren aufgestellt wurde, war die FFT unangefochtener und eindeutiger Sieger. Und dabei handelt es sich eigentlich bei der FFT um eine unglaublich einfache Idee. Doch da die FFT eigentlich "nur" eine schnelle Berechnungsmethode der *diskreten Fouriertranformation* oder *DFT*[70] darstellen, ist es vernünftig, sich diese zuerst einmal anzusehen.

2.6.1 Die diskrete Fouriertransformation

Eigentlich ist die Fouriertransformation einer Folge eine seltsame Operation, bildet sie doch, im Gegensatz zur "normalen" Fouriertransformation, eine Folge $c \in \ell(\mathbb{Z})$ auf die 2π–periodische Funktion $\widehat{c} \in C(\mathbb{T})$ ab, so daß die inverse Fouriertransformation (2.20) einer Folge eine völlig andere Struktur[71] hat als die Fouriertransformation selbst. Ganz abgesehen davon ist es sowieso nicht möglich, *kontinuierliche* Frequenzinformation praktisch zu verarbeiten, so daß man auch im Frequenzbereich abtasten müsste. Wegen der 2π–Periodizität der Fouriertransformierten $\widehat{c}$ empfiehlt es sich natürlich, diese Abtastgenauigkeit

[70] Glücklicherweise haben *"diskret"* und *"discrete"* denselben Anfangsbuchstaben.

[71] Natürlich sind diese Strukturen weder abwegig noch unnatürlich, aber um das wirklich zu verstehen muss man sich mit abstrakter harmonischer Analysis auseinandersetzen, insbesondere mit Haar–Maßen auf lokalkompakten abelschen Gruppen. Dann stellt sich heraus, daß $\mathbb{Z}$ und $\mathbb{T}$ **duale Gruppen** sind und daß die „normale" Fouiertransformierte eigentlich der Spezialfall ist, bei dem Gruppe und duale Gruppe übereinstimmen.

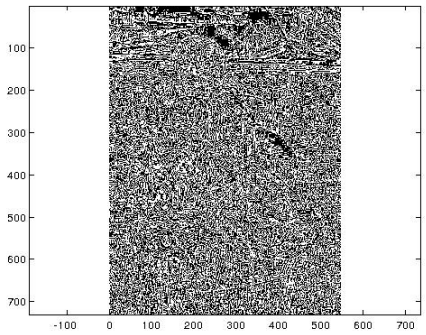


Abbildung 2.13: Das Rauschen im rechten Bild von Abb 2.12. Der Wertebereich ging von -1166 bis 605 und hier sind „nur“ die positiven Werte *binarisiert*, also als schwarze Punkte geplottet.

von der Form $h = \frac{2\pi}{n}$ für ein $n \in \mathbb{N}$ zu wählen. Dann ergibt sich die Folge

$$\widehat{c_n} = \mathrm{DFT}_n := S_{2\pi/n}\widehat{c} = \sum_{k\in\mathbb{Z}} c(k)\, e^{-2\pi i k\,\cdot/n}. \tag{2.34}$$

Definition 2.24 *Die Folge $\widehat{c_n}$ aus (2.34) bezeichnet man als **diskrete Fouriertransformierte** oder **DFT** der Folge $c \in \ell(\mathbb{Z})$ von der Ordnung n.*

Wegen

$$\widehat{c_n}\,(\cdot + n) = \sum_{k\in\mathbb{Z}} c(k)\, e^{-2\pi i\, k\,(\cdot/n+1)} = \widehat{c_n}$$

ist die DFT periodisch und es genügt, lediglich einen Block von n Einträgen zu speichern, das heißt, die DFT ist durch die Werte

$$\widehat{c_n}(k), \qquad k \in \mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z} \simeq \{0,\dots,n-1\},$$

festgelegt.

Bemerkung 2.25 *Unter $\mathbb{Z}_n$ ist normalerweise* mehr *zu verstehen, als "nur" die Menge $\{0,\dots,n-1\}$, zum Beispiel sind alle Operationen auf $\mathbb{Z}_n$ sind immer modulo n aufzufassen. Wir werden aber hier nicht so pingelig auf diesen Details herumreiten und $\mathbb{Z}_n$ auch manchmal nur für die Menge verwenden – zumindest solange die exakte Bedeutung des Symbols aus dem Kontext ohne allzuviel Aufwand ersichtlich wird.*

Auf periodischen oder *periodisierten* Folgen[72] $c \in \ell(\mathbb{Z}_m)$, also mit Periodenlänge m kann man die DFT sogar als Matrix darstellen, nämlich als

$$\widehat{c_n} = V_{n,m}\, c, \qquad V_{n,m} := \left[e^{-2i\pi jk/n} \;:\; j \in \mathbb{Z}_n,\, k \in \mathbb{Z}_m\right].$$

[72] Was nicht passt wird passend gemacht.

Ist $n = m$, dann schreiben wir einfach V_n. Das ist auch die "Standardversion" der DFT, bei der Signale in Signale gleicher Länge oder gleichen Informationsgehalts transformiert werden. Noch ein Wort zur Periodisierung: Ist $c \in \ell_{00}(\mathbb{Z})$, dann kann man c ja so schieben, daß der Träger der Folgen in $\mathbb{Z}_n$ liegt, und dann kann man c mit V_n diskret Fourier–transformieren.

Beispiel 2.26 *Wir bestimmen die diskrete Fouriertransformierte der Folge* $S_{2\pi/512}\cos$ *auf* $\mathbb{Z}_{512}$*. Realteil und Imagnärteil sind in Abb 2.14 dargestellt.*

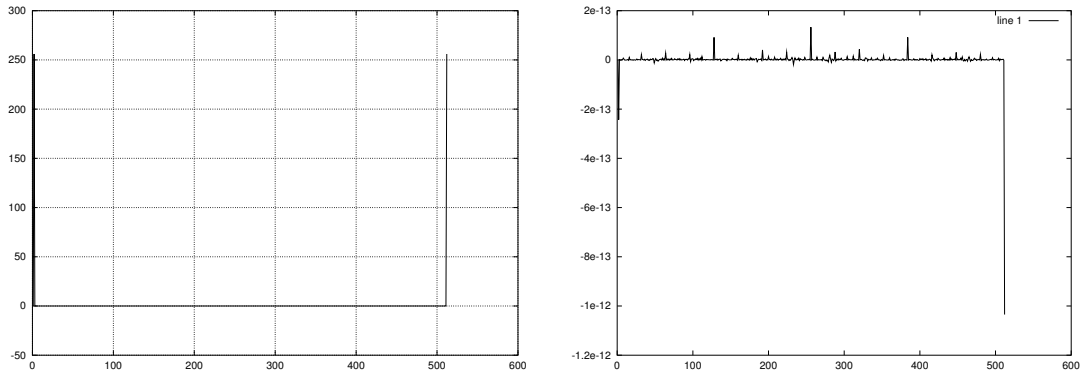


Abbildung 2.14: Die diskrete Fouriertransformierte

$$\mathrm{DFT}_{512}\, S_{2\pi/512}\cos .$$

Im linken Bild der Realteil, der starke Ausschläge an 1 und $511 \simeq -1$ hat – was ja auch passt, denn da

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}$$

ist, tauchen in ihm gerade die beiden Frequenzen ± 1 auf. Und natürlich ist der Imagiärteil rechts praktisch Null und besteht eigentlich nur aus numerischem Müll, auch wenn dieser mit 10^{-13} durchaus in einer nicht so begeisternded Größenordnung liegt.

Generell liefern Sinus- und Kosinusschwingungen mit Frequenzen, die Teiler der Abtastrate sind, scharfe Spitzen in Real- bzw. Imaginärteil der DFT während entsprechende Funktionen mit "unpassenden" Frequenzen "verwaschen" werden.

Beispiel 2.27 *Wir betrachten die Funktion*

$$f(x) = \cos x - \cos 80x + \cos 130.7x + \sin 16x - \sin 277.8x$$

und bestimmen $\mathrm{DFT}_{512} S_{2\pi/512} f$*. Das Ergebnis ist in Abb. 2.15 zu sehen.*

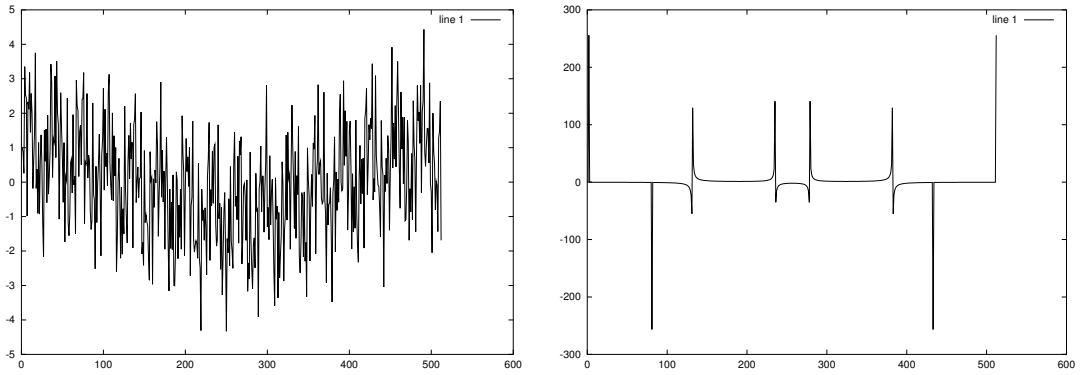


Abbildung 2.15: Die Funktion aus Beispiel 2.27 (links – sieht mit etwas Phantasie fast wie ein Sprachsignal aus) und Real- und Imaginärteil ihrer DFT (rechts). Die ganzzahligen Frequenzen liefern scharfe Zacken und zwar *entweder* im Real- *oder* im Imaginärteil, wohingegen die nichtganzzahligen Frequenzen zu "verschmierten" Ausschlägen in beiden Teilen des Spektrums führen.

Im weiteren gehen wir nun davon aus, daß sowohl c also auch seine diskrete Fouriertransformierte zu $\ell(\mathbb{Z}_n)$ gehören, daß wir es also mit der Matrix V_n zu tun haben. Sehen wir uns die Matrix mal genauer an; dazu ist es vernünftig $\omega = e^{-2\pi i/n}$ zu definieren und uns daran zu erinnern, daß ω eine (primitive) n–te **Einheitswurzel** ist, daß also

$$\omega^n = e^{-2\pi i} = 1 \tag{2.35}$$

gilt. Dann ist

$$V_n = \left[\omega^{jk} \;:\; j,k \in \mathbb{Z}_n\right] = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ 1 & \omega^1 & \dots & \omega^{n-2} & \omega^{n-1} \\ 1 & \omega^2 & \dots & \omega^{2(n-2)} & \omega^{2(n-1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \omega^{n-2} & \dots & \omega^4 & \omega^2 \\ 1 & \omega^{n-1} & \dots & \omega^2 & \omega^1 \end{bmatrix}$$

Diese Matrix hat eine sehr einfache Inverse und wir können daher den Prozess der DFT sehr einfach umkehren.

Lemma 2.28 (Inverse DFT) *Für $n \in \mathbb{N}$ ist die **inverse DFT** gegeben durch*

$$V_n^{-1} = \frac{1}{n}\left[e^{2\pi ijk/n} \;:\; j,k \in \mathbb{Z}_n\right] = \frac{1}{n}\left[\omega^{-jk} \;:\; j,k \in \mathbb{Z}_n\right]. \tag{2.36}$$

Beweis: Wir bezeichnen die Matrix auf der rechten Seite von (2.36) mit W_n, dann ist

$$(V_n W_n)_{jk} = \frac{1}{n}\sum_{\ell\in\mathbb{Z}_n} \omega^{j\ell}\,\omega^{-\ell k} = \frac{1}{n}\sum_{\ell\in\mathbb{Z}_n} \left(\omega^{j-k}\right)^{\ell}.$$

Damit ist

$$(V_n W_n)_{jj} = \frac{1}{n} \sum_{\ell \in \mathbb{Z}_n} \left(\omega^0\right)^\ell = \frac{n}{n} = 1$$

und ansonsten

$$(V_n W_n)_{jk} = \frac{1}{n} \sum_{\ell=0}^{n-1} \left(\omega^{j-k}\right)^\ell = \frac{1}{n} \frac{\omega^0 - (\omega^{j-k})^n}{1-\omega^{j-k}} = \frac{1}{n} \frac{1-(\omega^n)^{j-k}}{1-\omega^{j-k}}$$

und da $\omega^n = 1$ und $-n < j-k < n$, also auch $\omega^{j-k} \neq 1$ ist, erhalten wir, daß

$$(V_n W_n)_{jk} = \frac{1}{n} \frac{1-1^{j-k}}{1-\omega^{j-k}} = 0$$

ist. □

Nun ist aber $\omega^{-1} = e^{2\pi i/n}$ bei genauem Hinsehen nichts anderes als $\overline{\omega}$, das heißt, wir erhalten wegen der Symmetrie von V_n, daß

$$V_n^{-1} = \frac{1}{n}\overline{V_n} = \frac{1}{n}\overline{V_n^T} = \frac{1}{n}V_n^H,$$

was wir auch wie folgt formulieren können.

Korollar 2.29 *Die Matrix $n^{-1/2}V_n$ ist **unitär**.*

Bemerkung 2.30 *Die Verteilung des Faktors $\frac{1}{n}$ zwischen V_n und V_n^{-1} erscheint willkürlich und unsymmetrisch und tatsächlich gibt es auch Leute, die die DFT mit einem Vorfaktor $\sqrt{n}^{-1}$ einführen. Das macht zwar die Konstante "schöner", liefert, wie Korollar 2.29 zeigt, auch eine unitäre Matrix, zerstört aber die Interpretation als Abtastung des trigonometrischen Polynoms und führt eine zusätzliche* irrationale[73] *Größe ein. Außerdem spielt die Konstante $1/\sqrt{n}$ eine Rolle ganz analog zur Konstante $1/2\pi$ bei der Fouriertransformierten und deren Inverser. Und je mehr Analogien, desto besser – allerdings hängt bei der DFT die Normierungsgröße von der Länge des Vektors ab. Und ob die Matrix nun unitär ist oder nur $V_n V_n^H = V_n^H V_n = nI$ gilt, das ist genauso wesentlich wie der Unterschied zwischen orthogonal und orthonormal.*

Wichtig ist dieser Aspekt wieder einmal bei der Verwendung von Software, denn da sollte man sehr genau hinsehen, an welcher Stelle der DFT die Normalisierung erfolgt.

Offenbar ist für $c \in \ell(\mathbb{Z}_n)$ die DFT $\widehat{c_n} \in \ell(\mathbb{Z}_n)$ eine, wenn nicht sogar *die* "natürliche" Operation. Wir stellen nun ein paar Eigenschaften von $\mathrm{DFT}_n : \ell(\mathbb{Z}_n) \to \ell(\mathbb{Z}_n)$ zusammen, und zwar in Analogie zu Satz 2.8. Dazu brauchen wir auch den zur DFT gehörigen Faltungsbegriff und das ist die *zyklische Faltung*, die die "periodische" Struktur von $\mathbb{Z}_n$ ausnutzt[74].

[73] Naja, ganz so schlimm ist es auch wieder nicht, es ist ja "nur" eine Wurzel und die kann man auch algorithmisch effektiv zum Grundkörper $\mathbb{Q}$ adjungieren, siehe z.B. (Sauer, 2001).

[74] Hier rechnen wir jetzt *wirklich* modulo n.

Definition 2.31 (Zyklische Faltung) *Zu* $c, d \in \ell(\mathbb{Z}_n)$ *ist die* zyklische Faltung $c * d = c *_n d \in \ell(\mathbb{Z}_n)$ *definiert als*

$$(c * d)(j) = \sum_{k \in \mathbb{Z}_n} c(k)\, d(j-k), \qquad j \in \mathbb{Z}_n,$$

wobei $j - k$ *enstprechend den Rechenregeln in* $\mathbb{Z}_n$*, also modulo* n *zu verstehen ist.*

Satz 2.32 (Eigenschaften der DFT) *Für* $n \in \mathbb{N}$ *gilt:*

1. DFT_n *ist eine invertierbare lineare Abbildung von* $\ell(\mathbb{Z}_n)$ *in sich mit*[75]

$$\left\|\widehat{c_n}\right\|_2 = \sqrt{n}\|c\|_2, \qquad c \in \ell(\mathbb{Z}_n). \tag{2.37}$$

2. *Für* $c, d \in \ell(\mathbb{Z}_n)$ *ist*

$$(c *_n d)^\wedge_n = \widehat{c}_n\, \widehat{d}_n. \tag{2.38}$$

Beweis: *1.)* Linearität ist klar und Invertierbarkeit folgt aus Lemma 2.28 – da ist die inverse DFT ja explizit angegeben. Für (2.37) wenden wir die unitäre Invarianz der 2–Norm[76] an und erhalten

$$\left\|\widehat{c_n}\right\|_2 = \|V_n c\|_2 = \sqrt{n}\left\|\left(n^{-1/2}V_n\right)c\right\|_2 = \sqrt{n}\|c\|_2.$$

2.) Da ω eine n–te Einheitswurzel ist, also $\omega^n = 1$ und somit auch $\omega^{k+n} = \omega^k$ gilt, ist $k \mapsto \omega^k$ eine Folge in $\ell(\mathbb{Z}_n)$. Daher erhalten wir für $j \in \mathbb{Z}_n$, daß

$$\begin{aligned}(c *_n d)^\wedge_n(j) &= \sum_{\ell \in \mathbb{Z}_n}\left(\sum_{k \in \mathbb{Z}_n} c(k)\, d(\ell-k)\right)\omega^{j\ell} = \sum_{k,\ell \in \mathbb{Z}_n} c(k)\, d(\ell-k)\, \omega^{jk}\omega^{j(\ell-k)} \\ &= \widehat{c}_n(j)\, \widehat{d}_n(j).\end{aligned}$$

Die andere Hälfte von (2.38) funktioniert analog. □

Weil wir es ja mit Bildern, also zweidimensionalen Objekten, zu tun haben, sollten wir uns auch noch die zweidimensionale DFT zu $c \in \ell(\mathbb{Z}_n^2)$ ansehen, die dann als

$$\begin{aligned}\widehat{c}(\beta) &= \sum_{\alpha \in \mathbb{Z}_n^2} c(\alpha)\, e^{-2\pi i \alpha^T \beta/n} = \sum_{\alpha_1 \in \mathbb{Z}_n}\sum_{\alpha_2 \in \mathbb{Z}_n} c(\alpha_1, \alpha_2)\, e^{-2\pi i \alpha_1\beta_1/n} e^{-2\pi i \alpha_2\beta_2/n} \\ &= \sum_{\alpha_1 \in \mathbb{Z}_n} e^{-2\pi i \alpha_1\beta_1/n} \sum_{\alpha_2 \in \mathbb{Z}_n} c(\alpha_1, \alpha_2)\, e^{-2\pi i \alpha_2\beta_2/n} \\ &= \sum_{\alpha_1 \in \mathbb{Z}_n} \left(c(\alpha_1, \cdot)\right)^\wedge(\beta_2)\; e^{-2\pi i \alpha_1\beta_1/n}\end{aligned}$$

schreiben lässt. Diese Formel gibt uns eine „praktische" Bildungsregel für die zweidimensionale DFT:

---

[75] Natürlich sind die p–Normen zu $c \in \ell(\mathbb{Z}_n)$ als $\left(\sum_{k \in \mathbb{Z}_n} |c(k)|^p\right)^{1/p}$ bzw. $\max_{k \in \mathbb{Z}_n} |c(k)|$ definiert.

[76] Zur Erinnerung: Für unitäres U, d.h. $U^H U = I$ und beliebiges x ist $\|x\|_2^2 = x^H x = x^H U^H U x = \|Ux\|_2^2$.

Für jede *Zeile*[77] `c(j,:)` der Matrix `c` bildet man die eindimensionale DFT $(c(j,\cdot))^\wedge$ und erhält so einen eindimensionalen Vektor, den man ebenfalls noch einmal transformieren muss.

Schematisch kann man das folgendermaßen darstellen:

$$\begin{array}{|c|c|c|c|}\hline c(0,0) & \cdots & c(0,n-1) & \to \quad (c(0,\cdot))^\wedge \\ \hline c(1,0) & \cdots & c(1,n-1) & \to \quad (c(1,\cdot))^\wedge \\ \hline \vdots & \ddots & \vdots & \quad \vdots \\ \hline c(n-1,0) & \cdots & c(n-1,n-1) & \to \quad (c(n-1,\cdot))^\wedge \\ \hline \end{array} \qquad \downarrow \quad \widehat{c} \tag{2.39}$$

Damit hängt aber auch die Perfomance der zweidimensionalen DFT ganz klar an der der eindimensionalen DFT und einen $n \times n$–DFT kostet offenbar so viel wie $n+1$ eindimensionale n–DFTs, allerdings kann man die „horizontalen" Operationen in (2.39) nahezu beliebig parallelisieren, beispielsweise auf der Grafikkarte.

2.6.2 Diskret versus diskretisiert

In den allermeiste Praxisfällen, beispielsweise bei der Verarbeitung von Tönen oder auch Hirnstrommessungen, resultieren die zu verarbeitenden, diskreten Daten, aus der *endlichen* Abtastung eines kontinuierlichen Signals, also[78]

$$c(k) = (S_h f)(k), \qquad k \in \mathbb{Z}_N.$$

Wenn wir nun die DFT $\widehat{c}_N$ zu diesem Signal c berechnen, dann berechnen wir eine Diskretisierung des zugehörigen trigonometrischen Polynoms

$$\widehat{c}(\xi) = \sum_{k\in\mathbb{Z}} c(k)\, e^{-ik\xi} = \sum_{k\in\mathbb{Z}_N} f(hk)\, e^{-ik\xi},$$

auf dem Gitten $2\pi\mathbb{Z}_N/N$, also

$$\widehat{c}_n(j) = \sum_{k\in\mathbb{Z}_N} f(hk)\, e^{-2i\pi jk/N}, \qquad j \in \mathbb{Z}_N.$$

Dieser Vektor hat aber erst einmal keine direkte Verbindung zu dem, was wir eigentlich berechnen wollen, nämlich eine Diskretisierung der Fouriertransformation von f. Und das kann eben auch wieder zu Artefakten führen.

Beispiel 2.33 *Wir betrachten die DFT einer Abtastung der wohlbekannten* sinc*–Funktion, deren Fouriertransformierte ja eine charakteristische Funktion ist, und tasten sie, in* `Octave`*–Notation mittels*[79]

[77] Wir verwenden hier die Notation von `Matlab/octave`.

[78] Wir verwenden hier N für die Anzahl der Abtastungen, um zum Ausdruck zu bringen, daß es sich dabei um eine sehr große Anzahl handelt.

[79] Man sollte die sinc–Funktion nicht auf dem ganzzahligen Gitter abtasten, da bekäme man eine δ–Folge,

```
ocatve> N = 512; c = sinc( 100*pi*(0:N-1)/N );
```

ab. Das Ergebnis einer Fouriertransformation mit anschliessendem `fftshift`[80] *ist in Abb. 2.16 gezeigt – die Frequenzen am Rand des Bandpassfilters sind, wie man sieht, deutlich überhöht.*

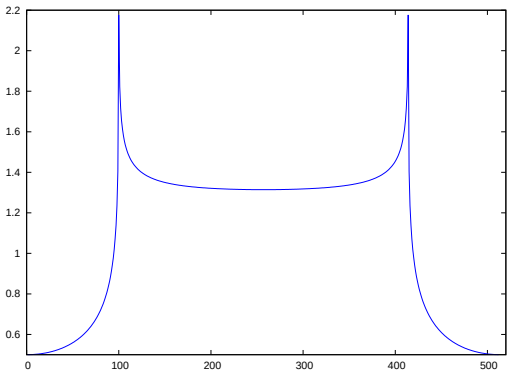


Abbildung 2.16: Die DFT der sinc–Funktion aus Beispiel 2.33. Man sieht, daß am Rand des Frequenzintervalls recht böse Artefakte auftreten, die mit Diskretisierung allein nicht erklärbar sind.

Um eine bessere Annäherung an die eigentliche Funktion zu erhalten, verwenden wir einen sogenannten *Quasiinterpolanten* als Approximation an f auf der Basis der Abtastungen. Zu einer Funktion $\phi \in L_{00}(\mathbb{R})$ ist der Quasiinterpolant recht einfach als skalierte Faltung

$$Q_{h,\phi} c := \phi * c\left(h^{-1}\cdot\right) = \sum_{k\in\mathbb{Z}_N} c(k)\phi\left(h^{-1}\cdot - k\right) = \sum_{k\in\mathbb{Z}_N} f(hk)\ \phi\left(h^{-1}\cdot - k\right)$$

definiert. Ist ϕ sogar eine *kardinale Funktion*, das heißt, gilt $\phi|_{\mathbb{Z}} = \delta$, dann ist $Q_{h,\phi}(hk) = S_h f(k) = f(hk)$, $k \in \mathbb{Z}_N$, es werden also die abgetasteten Daten interpoliert[81]. Andernfalls hofft man, so zumindest eine Approximation zu erhalten.

Beispiel 2.34 *Die gebräuchlichsten Funktionen für derartige Quasiinterpolanten sind die* kardinalen Splines*, also die Splines, deren unendliche Knotenmenge gerade $\mathbb{Z}$ ist. Darunter befinden sich* interpolatorische*, nämlich die Splines zu den Ordnungen 0 und 1, und* nichtinterpolatorische*, nämlich der ganze Rest. Die so resultierenden Approximationsoperatoren, die sogenannten* Schoenbergoperatoren *, sind beispielsweise in (Sauer, 2002a; Sauer, 2007; Schoenberg, 1973) beschrieben.*

Wenn wir einmal davon ausgehen, daß $Q_{h,\phi}c$ die Funktion f halbwegs approximiert, daß also $\left\|f - Q_{h,\phi}c\right\|_1$ klein ist, dann ist die Fouriertransformierte von

[80] Diese `Octave`–Funktion sorgt dafür, daß die Nullfrquenz in die Mitte des Vektors geschoben wird.

[81] Was in gewissem Sinne das "Quasi" erklärt.

$Q_{h,\phi}c$ auch eine gute Approximation der Fouriertransformierten von f und wir können letztere aus den abgetasteten Daten als

$$(Q_{h,\phi}c)^\wedge(\xi) \quad = \quad (\sigma_{h^{-1}}(\phi * c))^\wedge(\xi) = h(\phi * c)^\wedge(h\xi) = h\,\widehat{\phi}(h\xi)\,\widehat{c}(h\xi).$$

berechnen. Ersetzen wir in dieser Gleichung noch ξ durch $h^{-1}\xi$, und diskretieren ξ auf dem *diskreten Torus* $\mathbb{T}_N := 2\pi\mathbb{Z}_N/N$, so erhalten wir, daß

$$\widehat{f}\left(\frac{2k\pi}{Nh}\right) \simeq (Q_{h,\phi}c)^\wedge\left(\frac{2k\pi}{Nh}\right) = h\,\widehat{\phi}\left(\frac{2k\pi}{N}\right)\widehat{c}_N(k), \qquad k \in \mathbb{Z}_N. \tag{2.40}$$

Diese einfache Formel verknüpft nun die diskrete Fouriertransformation $\widehat{c}_N(k) = (S_h f)^\wedge_N$ der Abtastung mit einer *näherungsweisen* Diskretisierung der Fourietransformierten von f und erklärt auch sehr schön die Zusammenhänge:

1. Die *Abtastrate* bzw. *Samplingrate*[82] h bestimmt, welche Frequenzen von f wirklich in der DFT $\widehat{c}_N$ codiert sind und je kleiner h ist, desto größer wird dieser Frequenzbereich[83].

2. Die *Frequenzauflösung*, also die Anzahl der wirklich berechneten Spektrumseinträge, hängt hingegen von der gewählten Diskretisierungszahl N ab - je größer N ist, desto genauer wird das Spektrum dargestellt, je kleiner N ist, desto mehr Frquenzen werden zu einem Block zusammengefasst. Natürlich steigt mit wachsendem N auch der Rechenaufwand, doch dazu gleich mehr.

3. So einfach entkoppeln kann man diese beiden Größen nicht! Normalerweise resultieren die abgetasteten Daten ja aus Messungen über einen gewissen "nicht zu kurzen" Bereich bzw. Zeitraum[84], so daß Nh normalerweise von signifikanter Größe sein wird, was dazu führt, daß eine hohe Abtastrate in der Praxis auch mit einer hohen Frquenzauflösung verbunden sein dürfte.

4. Der Normierungsfaktor h in (2.40) ist nur dann wichtig, wenn wir uns wirklich für die konkreten Werte im Spektrum interessieren, geht es uns nur um die *Spektralverteilung*, dann können wir ihn berücksichtigen oder nicht.

5. Der Abstand zwischen $\widehat{f}$ und dem wirklich berechneten $(Q_{j,\phi}f)^\wedge$ beeinflusst natürlich ganz entscheidend, wie genau die berechneten Werte wirklich die diskretisierte Fouriertransfer beschreibt. Da $\left\|\widehat{g}\right\|_\infty \leq \|f\|_1$ gilt, und da wir nur auf dem Intervall $[0, Nh]$ abtasten, ist die L_1–Approximationsgüte

$$\left\|f - Q_{h,\phi}f\right\|_1 := \int_0^{Nh} |f(t) - Q_{h,\phi}f(t)|\,dt \leq Nh \max_{0\leq t\leq Nh} |f(t) - Q_{h,\phi}f(t)|$$

[82] Der internationale Terminus Technicus.

[83] Wer hätte das gedacht?

[84] Beispielsweise eine Aufnahme eines Musikstücks oder die Erfassung der Hirnstromdaten während eines psychologischen Experiments.

entscheidend für die Qualität unserer Näherung. Bei kardinalen Splinefunktionen gibt es Abschätzungen hierfür[85], siehe (Sauer, 2007), die normalerweise von der Größenordnung Ch^2 sind.

6. Verwendet man Splinefunktionen als ϕ, dann ist der Korrekturfilter $\widehat{\phi}$ besonders einfach zu berechnen, nämlich eine Potenz der sinc–Funktion, aber, wegen der Abtastung nur der ersten "Berg" dieser Funktione. Für hohe Frquenzen fällt so eine Potenz für steigende Ordnung der Spline natürlich auch immer schneller ab, sorgt also für eine stärkere Dämpfung unerwünschter hoher Frequenzen.

7. Lässt man ϕ in (2.40) weg, dann wählt man $\widehat{\phi}$ als $\chi_{[0,1]}$, also ϕ als sinc–Funktion und anstelle von $\widehat{f}$ diskretisiert man die Fouriertransformierte der interpolatorischen Rekonstruktion aus dem Abtastsatz. Die ist aber eben keine sonderlich gute Approximation, es sei denn, sie rekonstruiert exakt, das heisst, f müsste bandbeschränkt und die Abtastrate passend gewählt sein. Nur gerade dafür gibt es in den wenigsten Fällen wirklich eine Garantie.

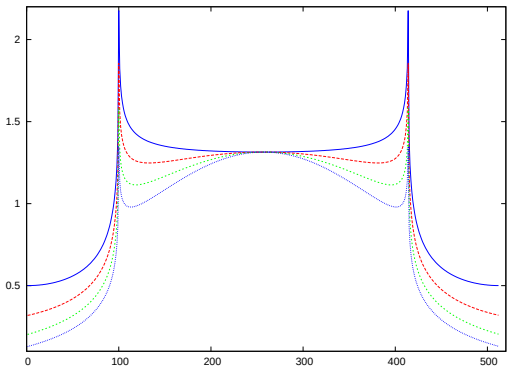


Abbildung 2.17: Filterung der Funktion aus Abb. 2.16 mit Splines der Ordnungen $0, 1, 2, 3$. Man sieht sehr schön, daß der Ausreisser deutlich kleiner wird, das allerdings um den Preis einer "Delle" am Rand. So ganz gut approximieren die Splines die sinc–Funktion also leider nicht.

2.6.3 Die schnelle Fouriertransformation

Nun liegt der besondere Charme der DFT aber nicht nur in der Tatsache, daß sie eine konsistente Erweiterung der Fouriertransformierten für periodische oder periodisierte Folgen ist, sondern vor allem daran, daß man sie besonders schnell durchführen kann. Das führt zur *schnellen Fouriertransformation* oder *Fast Fourier Transform*, kurz als *FFT* bezeichnet. Genaugenommen ist die FFT also

[85] Das Zauberwort hierfür heisst *Schoenbergoperator*.

eine FDFT, eine schnelle *diskrete* Fouriertransformation. Dieses Verfahren wurde 1965 in (Cooley & Tukey, 1965) von Cooley und Tukey (wieder)entdeckt, siehe dazu (Cooley, 1987; Cooley, 1990), und funktioniert nicht nur für die diskrete Fouriertransformation, sondern auf recht beliebigen Ringen – alles, was man wirklich braucht ist eine *primitive* n*–te Einheitswurzel* wie unser ω. Dabei ist die Idee hinter der FFT auch noch sehr einfach: Nehmen wir einmal an, daß $n = 2m$ eine gerade Zahl wäre und bemerken wir, daß

$$\omega^2 = e^{-2\pi i 2/n} = e^{-2\pi i m} =: \omega_m, \qquad \omega_n := \omega,$$

dann ist für $c \in \ell(\mathbb{Z}_n)$ und $j \in \mathbb{Z}_n$

$$\begin{aligned}
\widehat{c_n}(j) &= \sum_{k\in\mathbb{Z}_n} c(k)\,\omega^{jk} = \sum_{k\in\mathbb{Z}_m} c(2k)\,\omega^{2jk} + \sum_{k\in\mathbb{Z}_m} c(2k+1)\,\omega^{j(2k+1)} \\
&= \sum_{k\in\mathbb{Z}_m} c(2k)\,\omega_m^{jk} + \omega^j \sum_{k\in\mathbb{Z}_m} c(2k+1)\,\omega_m^{jk} \\
&= (c(2\cdot))^\wedge_m(j) + \omega^j\ (c(2\cdot+1))^\wedge_m(j),
\end{aligned}$$

also

$$\widehat{c_n} = (c(2\cdot))^\wedge_m + \omega^{\cdot}\,(c(2\cdot+1))^\wedge_m \tag{2.41}$$

Worin liegt nun der Wert dieser Darstellung? Nun, wenn wir einmal annehmen, daß die Werte $\omega, \ldots, \omega^{n-1}$ in tabellierter Form vorliegen und vorberechnet sind[86], dann benötigt die "naive" Realisierung der DFT als Multiplikation einer $n \times n$–Matrix mit einem n–Vektor $O(n^2)$ Rechenoperationen. Nehmen wir an, die Berechnung über (2.41) würde $F(n)$ Rechenoperationen benötigen. Dann sagt uns (2.41), daß wir zur Berechnung von $\widehat{c_n}$ die beiden DFTs der Länge $m = n/2$ berechnen müssen (Aufwand $2F(n/2)$), den zweiten komonentenweise mit dem Vektor $[\omega^j : j \in \mathbb{Z}_n]$ multiplizieren (Aufwand n) und die beiden komponentenweise addieren[87] (Aufwand n). Insgesamt müssen wir also einen Aufwand von $2(F(n/2) + n)$ betreiben und erhalten so die Beziehung

$$F(n) = 2\,(F(n/2) + n)\,, \tag{2.42}$$

für den unbekannten Aufwand n. Nehmen wir mal an, daß $n = 2^\ell$ für $\ell \in \mathbb{N}$ ist, dann gilt die Beziehung

$$F(n) = 2^k F\left(2^{\ell-k}\right) + k\,2^{\ell+1}, \qquad k = 1, \ldots, \ell, \tag{2.43}$$

was für $k = 1$ gerade (2.42) ist und sich sonst induktiv aus

$$\begin{aligned}
F(n) &= 2^k F\left(2^{\ell-k}\right) + k\,2^{\ell+1} = 2^k\,2\left[F\left(2^{\ell-k-1}\right) + 2^{\ell-k}\right] + k\,2^{\ell+1} \\
&= 2^{k+1} F\left(2^{\ell-k-1}\right) + 2^{\ell+1} + k\,2^{\ell+1} = 2^{k+1} F\left(2^{\ell-k-1}\right) + (k+1)\,2^{\ell+1}
\end{aligned}$$

[86] Diese Werte sind ja für alle Vektoren $c \in \ell(\mathbb{Z}_n)$ dieselben und können daher beispielsweise in einem Cachespeicher vorgehalten werden. Und selbst wenn sie nicht vorberechnet sind, dann kann man sie immer noch mit einem Aufwand von "nur" $O(n)$ bestimmen.

[87] Diese beiden Vektoren sind m–periodisch, werden also einfach fortgesetzt

ergibt. Betrachten wir nun speziell (2.43) für $k = \ell = \log_2 n$, dann ist

$$F(n) = \underbrace{2^\ell}_{=n} F(1) + \underbrace{\ell\, 2^{\ell+1}}_{=2n\,\log_2 n} = n\left(2\,\log_2 n + F(1)\right) = O\left(n\,\log_2 n\right),$$

was *deutlich* besser ist als $O(n^2)$. Tatsächlich ist $O\left(n\,\log_2 n\right)$ eine typische asymptotische Komplexität für derartige Methoden, die auf dem Prinzip "Halbieren und Rekursion" basieren, diese Tatsache wird bei diskreten Komplexitätsbetrachtungen auch gerne als *"Master Theorem"* bezeichnet, siehe z.B. (Steger, 2001).
Und diese Komplexitätsaussage gilt nicht nur für Zweierpotenzen! Ist nämlich $2^{\ell-1} < n \le 2^\ell$, dann ersetzen wir einfach n durch 2^ℓ indem wir die Vektoren beispielsweise durch Nullen ergänzen[88] und so eine Komplexität von

$$\begin{aligned} 2^\ell F(1) + 2\ell\, 2^\ell &\le 2n\,F(1) + 2\log_2(2n)\,2n = 2n\,F(1) + 4n\left(\log_2 n + 1\right) \\ &= 2n\left(2\log_2 n + F(1) + 2\right), \end{aligned}$$

also im wesentlichen nur einen Faktor 2 erhalten – asymptotisch ist das immer noch $O\left(n\,\log_2 n\right)$.

Beispiel 2.35 *Natürlich ist das mit dem "Auffüllen" nur ein reines Komplexitätsargument und nicht das, was man in der Realität machen sollte. Bildet man beispielsweise* $S_{2\pi/500}\cos(50\cdot)$ *also eine Folge in* $\ell(\mathbb{Z}_{500})$ *und füllt diese Folge mit* 12 *Nullen zu einem Element von* $\ell(\mathbb{Z}_{512})$ *auf, dann verschmieren sich die Frequenzen ganz gewaltig und zwar reell ebenso wie imaginär*[89]*, siehe Abb. 2.18*

Nun könnte man sagen, das Problem mit Abb. 2.18 läge daran, daß jedwede Form von Periodizität kaputtgemacht wird und man das Signal besser periodisch fortsetzen sollte – aber dann passen halt die beiden Perioden 500 und 512 auch wieder nicht zusammen, außer in dem glücklichen Fall, daß das Signal so hochfrequent ist, daß die Periodisierung gerade eine volle Signalperiode erwischt.

Die Form der FFT, die wir hier betrachtet haben, ist die sogenannte *Radix–2* FFT, da die Zerlegungen auf der Basis 2, also auf Halbierung des Datenbestands beruhen. Man kann das aber auch mit jeder anderen Zahl, beispielsweise mit der Basis $p \in \mathbb{N}$ und erhält dann für $m = n/p$ die analoge Zerlegung "modulo p"

$$\begin{aligned} \widehat{c_n}(j) &= \sum_{k\in\mathbb{Z}_m}\sum_{\ell\in\mathbb{Z}_p} c\,(pk+\ell)\;\omega^{j(pk+\ell)} = \sum_{\ell\in\mathbb{Z}_p}\omega^{j\ell}\left[\sum_{k\in\mathbb{Z}_m} c\,(pk+\ell)\;\omega_p^{jk}\right] \\ &= \sum_{\ell\in\mathbb{Z}_p}\omega^{j\ell}\;\left(c\,(p\cdot+\ell)\right)^\wedge_m(j), \end{aligned}$$

[88] Was zwar, wie wir sofort sehen werden, die Komplexität nicht signifikant verschlechtert, aber unsere schöne Periodizität zerstören wird!

[89] Zur Erinnerung: imaginäre Frequenzen dürften in der DFT gar nicht auftreten!

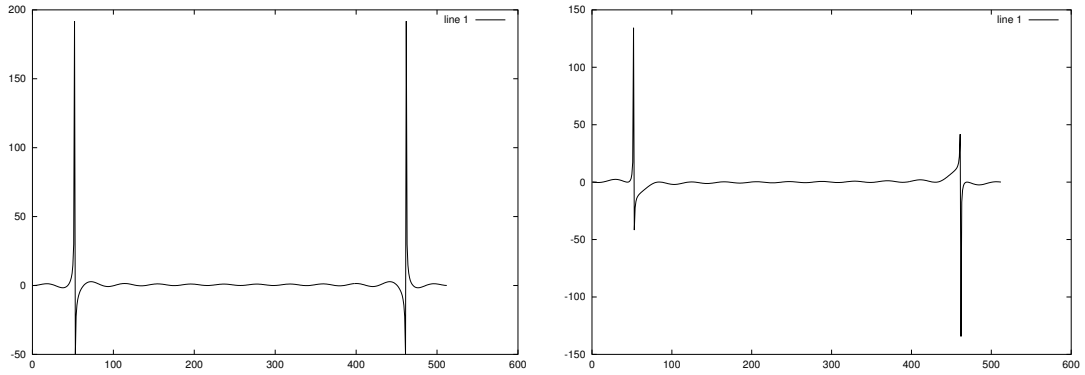


Abbildung 2.18: Real- (links) und Imaginärteil (rechts) der FFT von $S_{2\pi/500}\cos(50\cdot)$, aufgefüllt auf 512 Einträge. Zwar sind die beiden "Frequenzen" 50 und 450 immer noch deutlich sichtbar, aber die sehr großen Imaginärteile sind schon irreführend.

wir mußsen als *mehr* DFTs aber für *kürzere* Segmente berechnen. Der Aufwand hierbei ist dann $O\left(n \log_p n\right)$, bleibt also bis auf eine Konstante unverändert.

Übung 2.13 Zeigen Sie, daß der Rechenaufwand bei der Radix–p–FFT von der Größenordnung $O\left(n \log_p n\right)$ ist. ♢

Übung 2.14 Geben Sie auf der Basis von (2.39) ein Verfahren für die zweidimensionale FFT `fft2` an und bestimmen Sie die theoretische Komplexität. ♢

2.6.4 Fourier und Bilder

Ein wesentlicher Grund für die Nutzung der FFT ist die Tatsache, daß nun die zyklische Filterung (2.38) in $O\left(n \log_2 n\right)$ anstelle von $O(n^2)$ durchgeführt werden kann, indem man erst beide Signale transformiert, dann komponentenweise multipliziert und das Ergebnis zurücktransformiert. Die Transformationen kosten jeweils $O\left(n \log_2 n\right)$, die Multiplikation $O(n)$, so daß wir insgesamt[90] bei $O\left(n \log_2 n\right)$ landen.

Man kann die DFT aber auch direkt auf Bilder anwenden, aber sollte dabei schon ein bisschen aufpassen, wie Abb. 2.19 zeigt. Aber auch die logarithmische Skala[91] gibt uns nicht wirklich Information über das Bild. Das liegt im wesentlichen daran, daß die Fouriertransformation ihre Stärken natürlich bei *periodischen* Ereignissen hat. Das sieht man, wenn man ein Bild aus vertikalen Streifen generiert,

[90] Konstanten interessieren ja in diesem Kontext nicht.

[91] Der helle Fleck in der Mitte zeigt übrigens an, daß die FFT links in Abb. 2.19 eigentlich einen weissen Fleck in der Mitte, also bei der Frequenz 0 haben sollte, der alles andere gnadenlos dominiert.

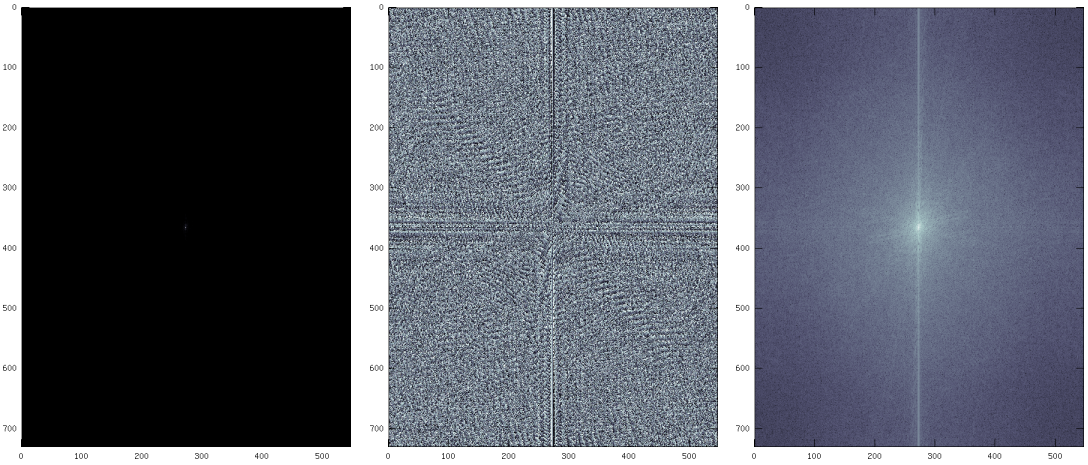


Abbildung 2.19: FFT unseres Testbilds Abb. 2.8, zuerst Absolutbetrag *(links)* und Phase *(mitte)*. Die Absolutbeträge variieren so stark, daß man erst einmal gar nichts sieht, was besser wird, wenn man eine **logarithmische Darstellung** *(rechts)* verwendet. Und noch eine Warnung: Das Phasenbild ist *zyklisch* zu sehen, weiß ist also dasselbe wie schwarz.

```
octave> X = kron( ones(64), ones(8,1)*[ 1 1 1 1 0 0 0 0 ] );
```

und dieses dann transformiert, siehe Abb. 4.3. Die Transformation zeigt dann ganz scharf und sauber den konstanten Teil (in der Mitte) und die Frequenz der Streifen an. Ein ähnliches Spiel liesse sich mit einem Schachbrettmuster machen.

Übung 2.15 Erzeugen Sie verschiedene Schachbrettmuster und stellen Sie deren DFT dar. ♢

Nun machen wir die Annahme, daß Bilder „lokal“ entweder konstant oder periodisch sind, also lokal entweder einen konstanten Farbwert oder eben eine periodische Textur haben, und versuchen, dies auszunutzen. Dazu zerlegen wir das Bild $C = [c(j,k) \,:\, j,k \in \mathbb{Z}_n]$ in m Blöcke der Größe n/m,

$$C = \begin{bmatrix} C_{00} & \dots & C_{0,m-1} \\ \vdots & \ddots & \vdots \\ C_{m-1,0} & \dots & C_{m-1,m-1} \end{bmatrix}, \qquad C_{jk} \in \mathbb{R}^{\frac{n}{m}\times\frac{n}{m}},$$

und transformieren jeden dieser Blöcke *separat*:

$$\widehat{C} \neq \widetilde{C} := \begin{bmatrix} \widehat{C}_{00} & \dots & \widehat{C}_{0,m-1} \\ \vdots & \ddots & \vdots \\ \widehat{C}_{m-1,0} & \dots & \widehat{C}_{m-1,m-1} \end{bmatrix}.$$

Die Ergebnisse sind für verschiendene Werte von m in Abb. 2.21 und Abb. 2.22 zu sehen. Die Berechnung jedes Blocks ist mit $C\,(n/m)^2 \log_2 n/m$ Operationen zu schaffen und da wir insgesamt m^2 Blöcke haben beträgt der gesamte

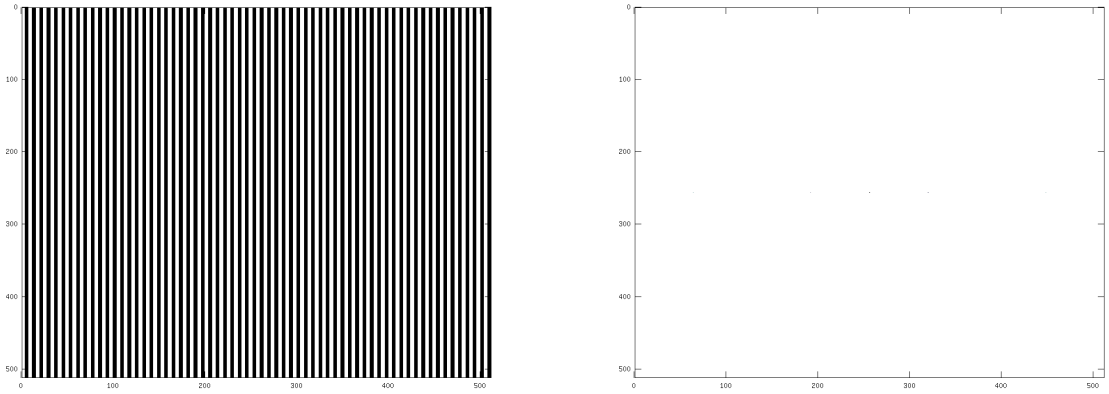


Abbildung 2.20: Ein „Streifenbild“ *(links)* und dessen (invertierte) DFT *(rechts)*, die wirklich nur diese drei kleinen, ganz scharf lokalisierten Punkte hat, wo sie von Null verschieden ist. Achtung: Die Punkte in der DFT sieht man nur bei passender Auflösung.

Rechenaufwand maximal $O(n^2 \log n)$, ist also derselbe wie für eine DFT des Bildes.

Auf diese *lokalen* DFTs können wir nun einfach und effizient einen **Tiefpassfilter** anwenden, indem wir jede der DFTs *komponentenweise* mit einer Maske $M \in \mathbb{R}^{\frac{n}{m} \times \frac{n}{m}}$ multiplizieren, die sich um die Mitte konzentriert, entweder mit einem Block aus lauter Einsen oder auch mit einem passend normalisierten Binomialfilter. Damit erhalten wir also

$$C^* := \begin{bmatrix} \widehat{C}_{00} \odot M & \dots & \widehat{C}_{0,m-1} \odot M \\ \vdots & \ddots & \vdots \\ \widehat{C}_{00} \odot M & \dots & \widehat{C}_{0,m-1} \odot M \end{bmatrix} = \widetilde{C} \odot (\mathbf{1}_{m \times m} \otimes M)\,.$$

Bei dieser Gelegenheit lernen wir gleich zwei neue Matrixprodukte kennen, und zwar das **Hadamard–Produkt**

$$A \odot B := \left[a_{jk} b_{jk} \;:\; \begin{matrix} j = 1, \dots, p \\ k = 1, \dots, q \end{matrix} \right] \in \mathbb{R}^{p \times q}, \qquad A, B \in \mathbb{R}^{p \times q},$$

und das **Kronecker–Produkt**[92]

$$A \otimes B = \begin{bmatrix} a_{11} B & \dots & a_{1q} B \\ \vdots & \ddots & \vdots \\ a_{p1} B & \dots & a_{pq} B \end{bmatrix} \in \mathbb{R}^{pr \times qs}, \qquad A \in \mathbb{R}^{p \times q}, \quad B \in \mathbb{R}^{r \times s}.$$

Beide sind in `Matlab/Octave` ausgesprochen effizient implementiert. Wenn wir C^* einmal ausgerechnet haben, was sich mit einem Aufwand von n^2/m^2 komponentenweisen Multiplikationen a m^2 Operationen, also mit $O(n^2)$ machen

[92] Das wir vorher in der `kron`–Funktion von `Matlab/Octave` schon kennengelernt haben.

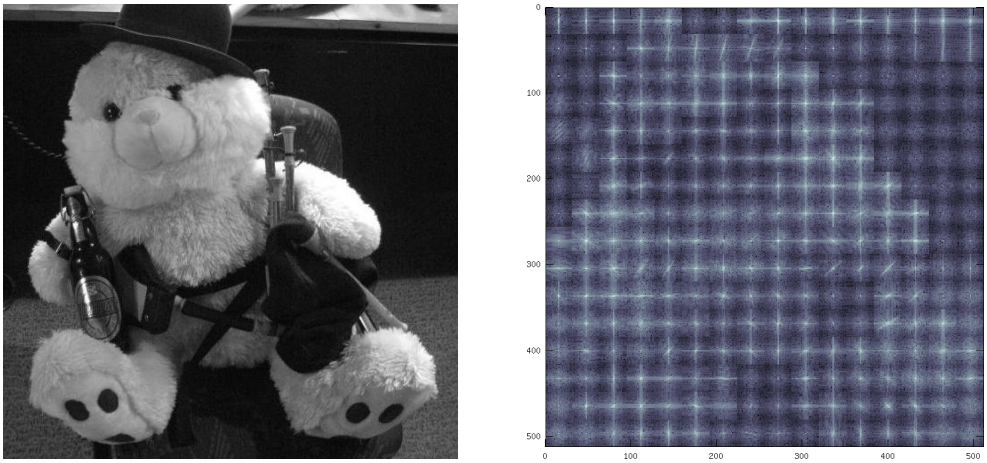


Abbildung 2.21: Bildausschnitt der Größe 512×512 von Abb. 2.8 *(links)* und die (logarithmierte) Block–DFT mit Blöcken der Größe 16 *(rechts)*. Dort, wo Konturen sind, erkennt man sehr ausgeprägte „Kreuze“ in der DFT, teilweise sogar in Richtung der Kanten, dort, wo nichts zu sehen ist (etwa am Fußboden oder rechts), ist die DFT eher diffus.

lässt, müssen wir diese Matrix nur noch zurücktransformieren und erhalten eine gefilterte, möglicherweise komprimierte Version von C.

Beispiel 2.36 *Wir verwenden den* 4×4*–Binomialfilter*

$$B_4 = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix} [1\,2\,2\,1] = \frac{1}{4}\begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 4 & 4 & 2 \\ 2 & 4 & 4 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix}$$

und betten diesen in eine $\frac{n}{m} \times \frac{n}{m}$*–Matrix ein. Den Rest der Matrix setzen wir einfach auf Null. In* `Octave` *geht das mit dem folgenden Stückchen Code, wobei das Bild der Größe* 512×512 *in* `A` *gespeichert sein soll:*

```
octave> bin4 = [ 1 2 2 1 ]'*[ 1 2 2 1 ] / 4;
octave> M = zeros( 16 ); M( 7:10,7:10 ) = bin4;
octave> B16 = blockDFTshift( A ) .* kron( ones( 32 ), M );
octave> A16 = blockIDFTshift( A );
```

Das resultierende Bild und die abgeschnittene Block–DFT sind in Abb. 2.23 zu sehen. Geringere Blockartefakte erhält man (natürlich) durch die Verwendung kleinerer Blöcke, siehe Abb. 2.24.

Bemerkung 2.37 *Eine ganz wichtige Bemerkung: Die Sache mit der DFT vorwärts und rückwärts und dem Verschieben der Nullfrequenz in die Mitte funktioniert nur dann, wenn die Blockgröße eine* ***Zweierpotenz*** *ist, also wenn* $n = 2^k$ *für ein passendes* k *gilt. Die Bildgröße selbst muss dann aber nur noch ein Vielfaches von* n *sein, das macht die Sache deutlich einfacher.*

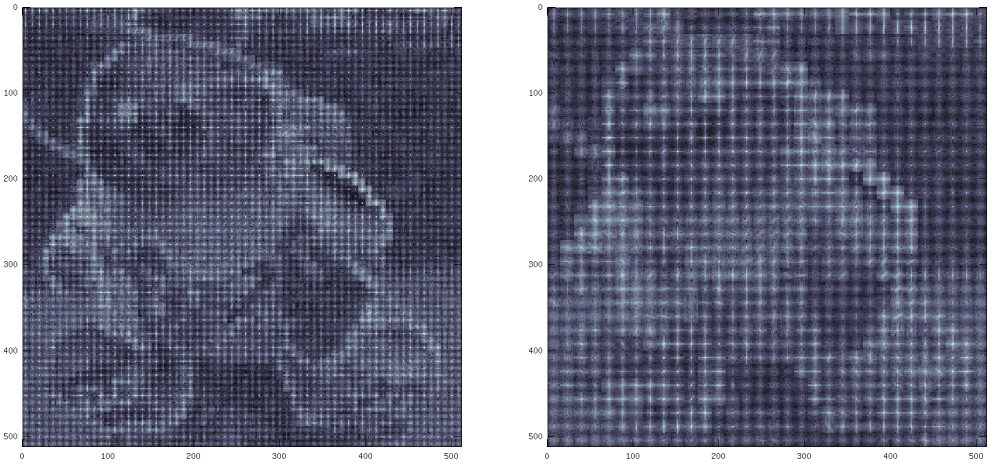


Abbildung 2.22: Block–DFT mit Blöcken der Größe 8 *(links)* und 16 *(rechts)*.

Gut, aber was haben wir nun mit dem ganzen Zauber gewonnen, außer daß wir eine Operation haben, die wir, FFT sei Dank, schnell durchführen können? Nun, werfen wir doch einfach nochmal einen Blick auf Abb. 2.23 und sehen uns die Block–DFT dort an. In dieser haben wir in jedem der 16×16–Blöcke alle Werte bis auf die vom 4×4–Filter „erwischten" auf Null gesetzt, wir haben also überhaupt nur noch $\frac{4\times4}{16\times16} = \frac{1}{16}$ der Information behalten. Das Bild wurde komprimiert!

Diese Idee, die Kompression einer Transformation, ist genau das Konzept hinter dem Bildkompressionsstandard **JPEG**[93]. Allerdings:

1. JPEG verwendet einiges an Tricks bereits bei der Bildvorbereitung! So werden unterschiedliche Farbkanäle unterschiedlich quantisiert, was der Farbwahrnehmung durch das menschliche Auge[94] entspricht.

2. JPEG filtert die DFT–Koeffizienten nicht so mit Holzhammer wie wir hier. Kleine Koeffizienten in der DFT kann man leicht unter den Tisch fallen lassen, ohne eine allzu großen Informationsverlust im Bild zu erleiden. Man sieht das schön in Abb 2.21, wo man ganze Bereiche fast auf Null setzen kann, während bei anderen der Informationsverlust durch blindwütiges Tiefpassfiltern ziemlich untragbar werden würde.

3. JPEG hat keine Lust, mit komplexen Zahlen zu rechnen und verwendet daher nicht die DFT, sondern die sogenannte DCT, die nur mit rellen Zahlen arbeitet, aber ebenfalls schnelle Berechnung zulässt.

4. JPEG schaltet hinter die Bildkompression noch einen Entropie–Encoder, der die Daten mit sogenannten **Huffman–Encoding** nochmals komprim-

[93] Die volle Dokumentation des JPEG–Standards findet sich unter `www.w3.org/Graphics/JPEG/itu-t81.pdf`.

[94] Hier orientiert man sich natürlich am „Normalwert", Menschen mit Farbfehlsichtigkeiten haben dann unter Umständen Pech gehabt.

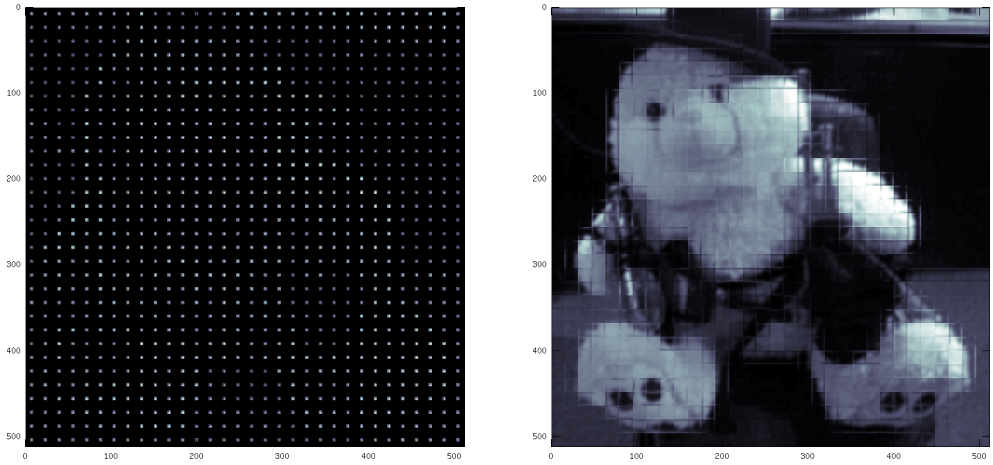


Abbildung 2.23: „Abgeschnittene“ Block–DFT des Bildes *(links)* und Rekonstruktion durch die inverse DFT *(rechts)*. Das Bild ist immer noch gut erkennbar, auch wenn man natürlich nun deutliche Blockartefakte erhält.

iert. Dies ist ein *verlustfreies* Verfahren[95], im Gegensatz zu den DCT–Modifikationen, die normalerweise immer auch Informationsverlust bedeuten. Aber gute Kompression gibt's halt nun einmal nicht umsonst.

Definition 2.38 (DCT) *Die **diskrete Kosinustransformation** oder **DCT**[96] bildet einen Vektor $f \in \ell(\mathbb{Z}_n)$ auf eine Linearkombination von Cosinustermen ab, beispielsweise als*

$$\mathrm{DCT}_{\mathrm{II}} f(j) = \sum_{k \in \mathbb{Z}_n} f(k) \cos \frac{\pi\left(k + \frac{1}{2}\right) j}{n}. \tag{2.44}$$

*Diese gebräuchlichste Transformation nennt man auch **DCT–II**, insgesamt gibt es vier Typen der DCT.*

Natürlich könnte man die DCT nun direkt und naiv über die Matrix

$$D_n^{\mathrm{II}} := \left[\cos \frac{\pi\left(k + \frac{1}{2}\right) j}{n} \ : \ j, k \in \mathbb{Z}_n \right], \qquad \text{d.h.} \qquad \mathrm{DCT}_{\mathrm{II}} f = D_n^{\mathrm{II}} f,$$

realisieren, aber das wre natürlich nicht besonders geschickt oder effizient. Da aber

$$\cos \frac{\pi\left(k + \frac{1}{2}\right) j}{n} \ = \ \cos \frac{2\pi\,(2k+1)\, j}{4n} = 2\left(e^{\frac{2\pi i (2k+1) j}{4n}} + e^{\frac{-2\pi i (2k+1) j}{4n}} \right)$$

[95] Das unter gewissen Umständen sogar zu einer leichten Vergrößerung der Ausgangsdatei führen kann.

[96] **D**iscrete **C**osine **T**ransform.

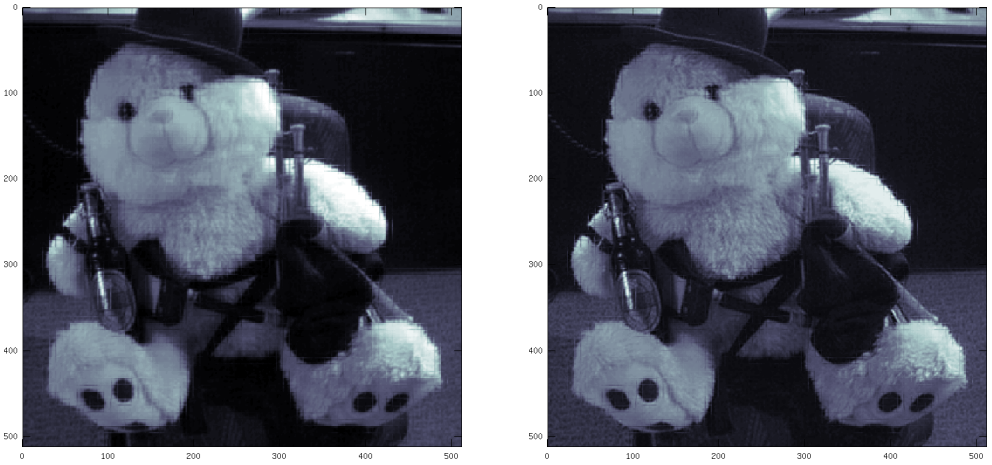


Abbildung 2.24: Rekonstruktion aus 8×8–Blöcken, einmal mit 4×4–Binomialfilter *(links)*, einmal mit $\mathbf{1}_{4\times4}$ *(rechts)*. Gibt es Unterschiede?

ist, brauchen wir unser Signal c nur mit Nullen aufzufüllen, das heisst, wir definieren $d \in \ell(\mathbb{Z}_{4n})$ als[97]

$$d(2k+1) = d(-2k-1) = c(k), \quad k \in \mathbb{Z}_n, \qquad d(2k) = 0, \quad k \in \mathbb{Z}_{2n},$$

und erhalten für $j \in \mathbb{Z}_{4n}$, daß

$$\begin{aligned}
\widehat{d}(j) &= \sum_{k\in\mathbb{Z}_{4n}} d(k)\, e^{-2\pi ijk/(4n)} = \sum_{k\in\mathbb{Z}_{2n}} d(k)\, e^{2\pi ijk/(4n)} + d(-k)\, e^{-2\pi ijk/(4n)} \\
&= \sum_{k\in\mathbb{Z}_n} c(k)\, 2\left(e^{\frac{2\pi i(2k+1)j}{4n}} + e^{\frac{-2\pi i(2k+1)j}{4n}}\right) = \sum_{k\in\mathbb{Z}_n} c(k)\, \cos\frac{\pi\left(k+\frac{1}{2}\right)j}{n} \\
&= \mathrm{DCT}_{II} c(j),
\end{aligned}$$

womit wir auch die DCT wieder mit einem Aufwand von $O\left(n \log_2 n\right)$ berechnen können. Mit unseren Standardmethoden lässt sich nun auch wieder eine zweidimensionale DCT definieren und via `fft2` implementieren.

Übung 2.16 Formulieren Sie die zweidimensionale DCT und implementieren Sie sie auf Basis der FFT. ◊

Bemerkung 2.39 (DCT) *Auch wenn man die DCT prinzipiell über die FFT realisieren kann, ist der Faktor 4 bzw. 16 im zweidimensionalen Fall nicht so wirklich angenehm. Daher gibt es auch spezifische Methoden, die DFT direkt schnell zu errechnen, die im wesentlichen auf einer Matrixformulierung der FFT–Idee basieren: Man kann die Aufspaltung in der FFT nämlich auch als eine* ***Matrixfaktorisierung*** *schreiben.*

[97] Zur Erinnerung: In $\mathbb{Z}_{4n}$ ist $-j = 4n - j$.

Er exzerpierte beständig, und alles, was er las, ging aus einem Buch neben dem Kopfe vorbei in ein anderes.

Lichtenberg

3 Transformationen

Als nächstes wollen wir uns ein wenig mit *Transformationen* beschäftigen. Wie der Name schon sagt, *transformiert* so eine Transformation ein Objekt (Signal oder Bild) in eine andere Form oder sogar Struktur. Das kann zwei Gründe haben:

1. Wir können nicht das Bild selbst, sondern eben nur eine Transformation davon messen. Die Radon–Transformation aus (1.1) war ein schönes Beispiel dafür, auch jede Fotographie ist immer nur eine Transformation des eigentlichen Sachverhalts. Praktisch besteht der Jo dann immer darin, diese Transformationen umzukehren und das Originalbild aus der Transformation zu rekonstruieren.

2. Die Transformation ermöglicht es uns, Bild- oder Signalinformation zu erkennen, die wir in den Ausgangsdaten nicht gesehen haben. Das passiert beispielsweise bei der Umformung in das Spektrum durch die Fouriertransformation.

Ein paar in der Signal- und Bildverarbeitung besonders wichtige und gebräuchliche Transformationen wollen wir uns nun näher ansehen.

3.1 Die Hough–Transformation

Die **Hough–Transformation** wurde 1962 in einer Patentanmeldung (Hough, 1962) veröffentlicht und ist heutzutage in der Bildverarbeitung ausgesprochen populär. Verwendet wird sie zur Erkennung von Linien in Bildern und sie basiert auf einer Dualität zwischen Punkten und Geraden. Das kennen wir schon aus der Radon–Transformation, wo wir eine Linie in der Ebene als

$$L = \left\{ x \in \mathbb{R}^2 \,:\, v^T x = c \right\}, \qquad \|v\|_2 = 1, \quad c \in \mathbb{R}, \tag{3.1}$$

beschrieben und mit den Werten v und c „codiert“ wird, was wir als $L = L(v, c)$ schreiben können. Dabei gibt es noch weitere Freiheitsgrade, denn offensichtlich ist

$$v^T x = c \quad \Leftrightarrow \quad (-v)^T x = (-c), \qquad \text{d.h.} \qquad L(v, c) = L(-v, -c),$$

die Codierung der Linien ist eine **gerade Funktion**. Die Vektoren v auf dem Einheitskreis kann man nun wieder als

$$v = v_\theta = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \qquad \theta \in [-\pi, \pi]$$

schreiben und unter Ausnutzung der „Freiheitsgrade“ beim Vorzeichen Geraden als

$$L = L(\theta, c) := L(v_\theta, c), \qquad \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right), \quad c \in \mathbb{R},$$

darstellen. Wichtig ist dabei das reduzierte Intervall, durch das diese Darstellung der Geraden *eindeutig* wird. Jetzt drehen wir den Spieß um und betrachten mit

$$H(x) = \left\{(\theta, c) \ : \ v_\theta^T x = c\right\} \subset \left[-\frac{\pi}{2}, \frac{\pi}{2}\right) \times \mathbb{R} =: \mathbb{H}, \tag{3.2}$$

die Parametrisierungen aller Geraden durch x als Kurve in $\mathbb{H}$. Bilden wir nun $H(x)$ zu einer Menge $X \subset \mathbb{R}^2$ von Punkten und liegen dieser viele Punkte auf einer Geraden $L(\theta, c)$, dann wird das Paar (θ, c) auch oft in der Menge $H(X)$ auftauchen, und zwar umso öfter, je mehr Punkte auf dieser Geraden liegen. Anders gesagt: Zu $(\theta, c) \in \mathbb{H}$ zählt

$$n_X(\theta, c) := \#\{x \in X \ : \ (\theta, c) \in H(x)\},$$

wieviele Punkte aus X auf der Geraden $L(\theta, c)$ liegen, wobei natürlich nur Werte $n_X(\theta, c) \gg 2$ von Interesse sind[98]. Da man in (3.2) ja $H(x)$ für jeden Bildpunkt als Indikatorkurve aller Geraden durch x bestimmt, hängt der Wert der Transformation also nicht vom Farb- oder Helligkeitswert des Punktes ab, sondern nur davon, ob er „da“ ist oder nicht.

Definition 3.1

1. *Ein **binarisiertes Bild** ist ein Bild, das nur die Werte 0 und 1 annimmt*[99].
2. *Die **Hough–Transformation** eines endlichen*[100] *binarisierten Bildes mit Pixeln $X = \{(x_j, y_j) \in \mathbb{R}^2 \ : \ j = 1, \ldots, N\}$ ist*

$$(H(X))(\theta, c) = n_X(\theta, c), \qquad (\theta, c) \in \mathbb{H}.$$

Natürlich ist $H(X)$ fast überall 0 – die meisten Geraden treffen schlicht und ergreifend keinen Punkt – und das würde zu einer ziemlich instabilen Definition unserer Transformation führen. Deswegen zerlegt man $\mathbb{H}$ in Bereiche $\mathbb{H}_{jk} = \Theta_j \times C_k$, $j, k = 1, \ldots, M, M'$, wobei die Θ_j eine Partition[101] $[-\pi/2, \pi/2]$ ist und C_k eine Zerlegung eines hinreichen großen Teilbereichs von $\mathbb{R}$, und zählt dann, wie viele der Werte $H(x_\ell)$ einen Bereich $\mathbb{H}_{jk}$ treffen. Das geht noch relativ schnell und einfach in digitalisierter Form, wenn man folgendermaßen vorgeht:

[98] Durch zwei Punkte passt bekanntlich immer eine Gerade.

[99] Also „Pixel da“ oder „Pixel nicht da“.

[100] Wer's mathematisch abgehoben will, kann das ganze gerne auch mit Maßen und so weiter auf's Kontinuierliche erweitern.

[101] Eine **Partition** X_j von X ist eine Zerlegung mit der Eigenschaft $X_j \subset X$, $\bigcup_j X_j = X$ und $X_j^\circ \cap X_k^\circ = \emptyset$, also eine Zerlegung in Teilmengen mit disjunktem Inneren.

Algorithmus 3.2 (Diskrete Hough–Transform)

1. ***Gegeben:*** *Punkt* $x = (x, y)$
2. *Setze* $\mathbf{H} = \mathbf{0}_{M \times M'}$
3. *Für* $j = 1, \ldots, M$*:*
   (a) $\theta_j =$ *Mittelpunkt von* Θ_j.
   (b) *Bestimme Index* $1 \leq k \leq M'$ *so, daß*
   $$x \cos \theta_j + y \sin \theta_j \in C_k.$$
   (c) $H_{jk} \leftarrow H_{jk} + 1$.
4. ***Ergebnis:*** $M =$ *diskretisierte Hough–Transformation.*

Übung 3.1 Implementieren Sie diese Version der Hough–Transformation. ◊

Erfreulicherweise ist die Hough–Transformation bereits in `Matlab`[102] und `Octave` als `houghtf` realisiert[103], so daß wir mal kurz damit spielen können. Man verwendet die Hough–Transformation normalerweise zur Bestimmung von Linienkanten in Bildern, wozu das Bild erst einmal mit einem Kantenfilter bearbeitet und dies dann gegen einen Schwellenwert binarisiert werden sollte. Das hat noch einen weiteren wichtigen Grund: Die Komplexität der Hough–Transformation hängt linear von der Anzahl der zu untersuchenden Pixel ab[104], und die Kanten sind normalerweise eindimensional, weswegen man davon ausgehen kann, daß ein Bild mit N Pixeln nur in etwa $O\left(\sqrt{N}\right)$ Kantenpixel haben sollte.

Fangen wir also an und verwenden wir wieder einmal unser Beispielbild aus Abb 2.8. Zuerst bestimmen wir über Gradientenfilter und 1–Norm das Kantenbild aus Abb 2.11:

```
octave> Gx = [ -ones( 3,1) zeros(3) ones( 3,1 ) ]/9; Gy = -Gx';
octave> K = abs( filter2( Gx,H ) ) + abs( filter2( Gy,H ) );
octave> Kb = K > max( max( K ) ) / 7;
```

Der letzte Schritt, die Bestimmung des binarisierten Kantenbildes `Kb` ist reine Willkür. Der Schwellenwert $\frac{1}{7}$ mal Maximum wurde so gewählt, daß möglichst viele Linien und möglichst wenige isolierte Punkte im Kantenbild enthalten sind, siehe Abb. 3.1. Nun verwenden wir die „eingebaute“ Funktion `houghtf`, um die Transformation zu berechnen:

```
octave> [Ht,R] = houghtf( Kb );
```

[102] Dort aber nur in der zusätzlich kostenpflichtigen „Image–Processing Toolbox“.

[103] Das sollte aber niemanden davon abhalten, auch einmal selbst eine Implementierung zu versuchen, nur so kann man wirklich verstehen, was in so einem Algorithmus passiert und was da schiefgehen kann.

[104] Und wegen des gemeinsamen Zugriffs aller Transformationen auf den „Akkumulator“ $\mathbf{H}$ ist auch das Parallelisierungspotential nicht so wirklich riesig.

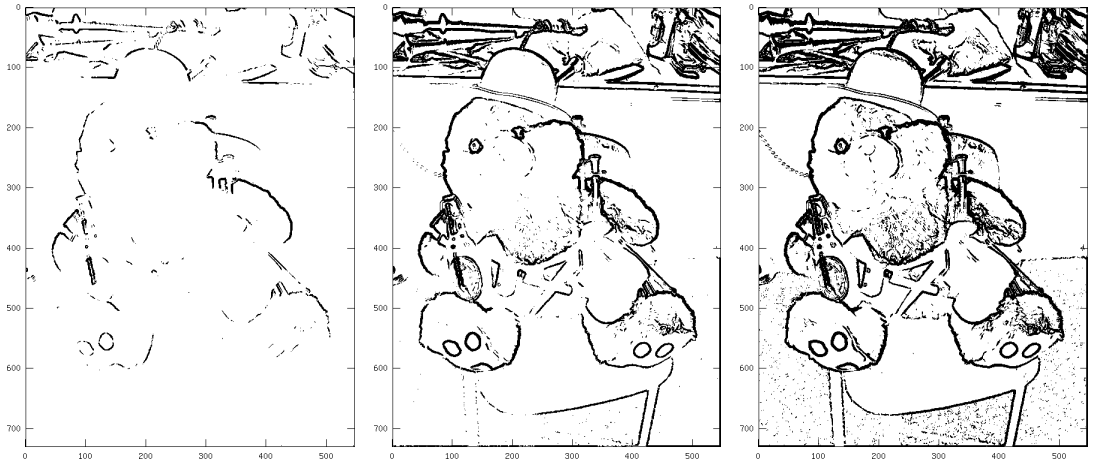


Abbildung 3.1: Binarisierte Kantenbilder mit den Schwellenwerten $\frac{1}{3}$ max *(links)* und $\frac{1}{10}$ max *(rechts)* in Inversdarstellung. Während links noch viele Kanten fehlen, kommt rechts bereits das Bildrauschen durch die Texturen mit ins Bild. In der Mitte die Wahl $\frac{1}{7}$ max.

Dabei werden als Voreinstellung die Winkel mit Abstand 1° abgetastet, also von `1:180`, und für die c–Werte wird der Bereich der doppelten Länge der Diagonalen genommen, mit `0` in der Mitte, die wirklichen c–Werte werden in der Variablen `R` zurückgegeben, das macht es einfacher. Das Ergebnis der Transformation ist in Abb. 3.2 zu sehen. Jetzt holen wir uns die dominantesten Linien aus der Hough–Transformation und lassen diese zusammen mit dem Bild plotten:

```
octave> [Hr,Hc] = find( Ht > .6*max(max(Ht) ) );
octave> imagesc( 1.-Kb ); colormap( bone ); axis equal
octave> houghLines( [Hr,Hc],R,H );
```

Das Ergebnis in Abb 3.3 zeigt sehr deutlich die Vor- und Nachteile der Methode:

1. Es werden Kanten erkannt, die keine sind! Die am stärksten ausgeprägten Maxima der Hough–Transformation entsprechen Bildrändern. Dies ist andererseits aber korrekt, denn auf dem Rand liegen ja relativ viele Pixel und diesen „Fehler“ kann man auch ausgesprochen leicht korrigieren.

2. Die Kanten werden nicht lokalisiert! Die erste detektierte „wirkliche“ Kante ist die schräg von links oben nach rechts unten verlaufende, die durch die Spielpfeife des Instruments definiert wird. Sie ist zwar sehr dominant, aber dennoch nur sehr kurz. Wo genau im Bild die zugehörige Kante liegt, erfordert weitere Arbeit: Man müsste die Kante im Bild diskretisieren[105] und dann nach allen Pixeln suchen, die nahe genug bei dieser Geraden liegen. Das ist machbar, aber nicht umsonst.

[105] Wofür es aus der Computergrafik eine Vielzahl von schnellen Verfahren gibt, siehe (Foley *et al.*, 1990).

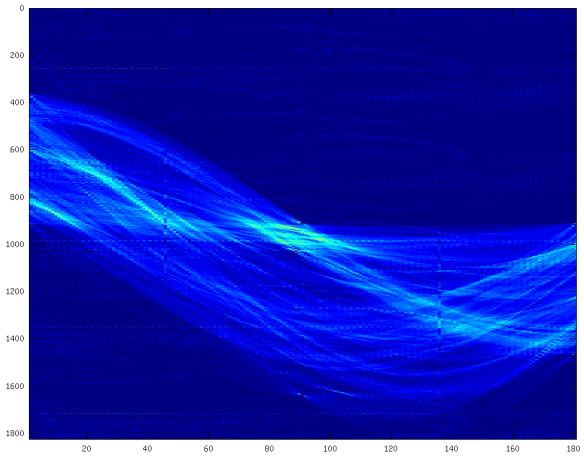


Abbildung 3.2: Hough–Transformation des Kantenbilds. Ein paar Maxima sind gut zu erkennen.

3. Dafür können Kanten durch die Hough–Transformation auch dann erkannt bzw. rekonstruiert werden, wenn sie durch andere Bildelemente teilweise verdeckt werden, wie beispielsweise der Teil der oberen Tischkante, der hinter dem Hut verschwindet.

4. Wir stehen nach wie vor vor dem Problem des passend gewählten Schwellenwerts. Die 60% Maximalwert aus unserem Beispiel waren wieder einmal total willkürlich und unterschlagen ja auch ein paar durchaus wichtige Kanten, beispielsweise die untere Tischkante oder die Stuhlbeine. Das heißt, man sollte den Schwellenwert wohl niedriger wählen. Wählt man ihn hingegen zu gering, so sieht man das Bild vor lauter Linien nicht mehr.

Die Hough–Transformation ist vor allem dann nützlich, wenn Linien durch das ganze Bild hindurchlaufe und eventuell irgendwo unterbrochen werden oder wenn das Bild durch wenige Geraden dominiert wird.

Beispiel 3.3 (Ausrichtung rechteckiger Objekte) *Hat man es mit rechteckigen Objekten, idealerweise bekannter Ausdehnung zu tun, so kann man deren Ausrichtung (Verdrehung) mit Hilfe der Hough–Transformation aus Bildern recht gut erkennen, beispielsweise bei der Überwachung von Fließbändern. Hat man einmal den Threshold kalibriert, so kann man eine automtische Detektion der Richtung eigentlich sehr einfach und leidlich schnell durchführen.*

Übung 3.2 Entwickeln Sie einen Algorithmus zur Erkennung der Ausrichtung eines rechteckigen Objekts Ihrer Wahl. Dabei dürfen Sie Kantenerkenner und Hough–Transformation aus `Matlab/Octave` verwenden. ♢

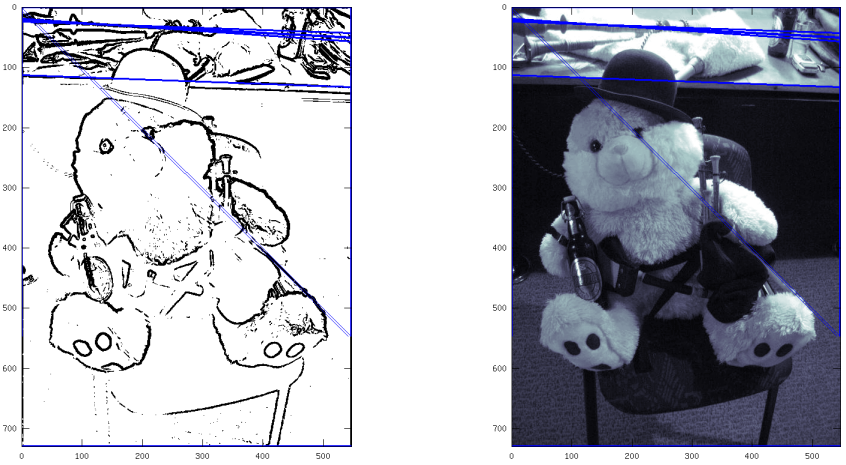


Abbildung 3.3: Kantendetektion über die Hough–Transformation, sowohl im Konturbild *(links)* als auch im Bild selbst *(rechts)*. Bei genauem Hinsehen erkennt man, daß auch der Bildrand von der Transformation detektiert wurde und das sogar sehr massiv.

Mit ein klein wenig mathematischem Abstand kann man die Hough–Transformation auch abstrakter sehen und auf beliebige paramterisierte implizite Kurven verallgemeinern.

Definition 3.4 *Eine implizite parametrische Kurve zu einer Funktion* $F : \mathbb{R}^2 \times \mathbb{P}$ *ist die Menge*

$$\left\{x \in \mathbb{R}^2 \ : \ 0 = f_p(x) = F(x,p)\right\} \tag{3.3}$$

zu einem vorgegebenen ***Parametervektor*** $p \in \mathbb{H}$.

Die ***Hough–Transformation*** *bezüglich* F *ordnet nun jedem Punkt* $x \in \mathbb{R}^s$ *die Menge aller „passenden" Parametervektoren zu:*

$$H(x) := \{p \in \mathbb{H} \ : \ F(x,p) = 0\}. \tag{3.4}$$

Beispiel 3.5 *Im Fall der „normalen" Hough–Transformation war*[106]

$$\begin{aligned} \mathbb{H} = \mathbb{S}^2 \times \mathbb{R}_+ \quad & \textit{und} \quad F(x,p) = \nu^T x + c, \qquad p = (\nu, c), \\ \mathbb{H} = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times \mathbb{R} \quad & \textit{und} \quad F(x,p) = [\cos\theta, \sin\theta]\, x + c, \qquad p = (\theta, c). \end{aligned}$$

In der Literatur findet sich im wesentlichen noch ein weiteres Beispiel für die Hough–Transformation, nämlich

$$F(x,p) = (x_1 - p_1)^2 + (x_2 - p_2)^2 - p_3^2,$$

das heisst, p codiert Mittelpunkt und Radius eines Kreises. Man sammelt jetzt also alle Kreise durch den Punkt x, was ja noch recht einfach geht, da zu einem

[106] $\mathbb{S}^d = \left\{x \in \mathbb{R}^d \ : \ \|x\|_2 = 1\right\}$ steht für die d–dimensionale **Einheitssphäre**.

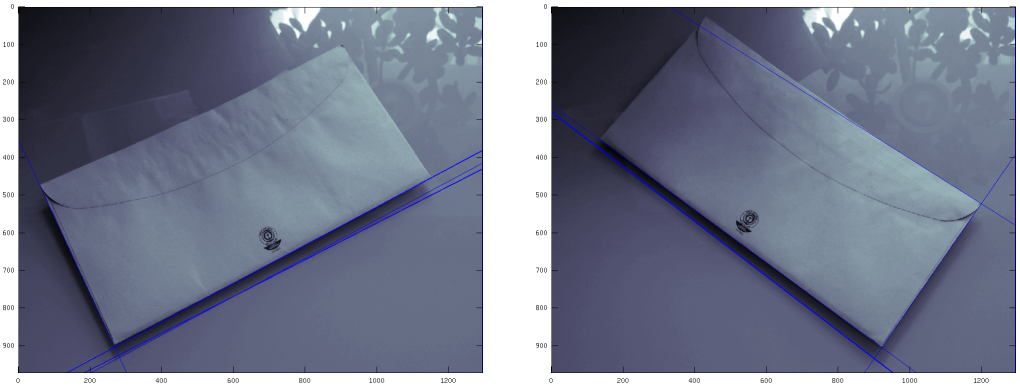


Abbildung 3.4: Zwei Fotos eines Briefumschlags mit verschiedenen Ausrichtungen. Man sieht, daß die Hough–Transformation prinzipiell die Ausrichtung erkennt, daß die länglichen Schattenartefakte zu Problemen führen können, das Muster im Hintergrund oder das auf dem Briefumschlag aber nicht.

vorgegebenen Radius $r = p_3 \in \mathbb{R}_+$ die Mittelpunkte (p_1, p_2) der Kreise durch x sich ihrerseits wieder auf einem Kreis mit Radius r um x befinden. Allerdings muss man nun suchen, wo sich in einem dreidimensionalen Raum die Kreise häufen, wenn denn viele Pixel auf Kreislinien liegen. Das ist deutlich teuerer, denn eine Zerlegung eines Würfels mit Kantenlänge h hat $O(h^{-3})$ Teilwürfel, während ein Rechteck nur $O(h^{-2})$ Teile hat. Damit braucht man aber auch mehr Pixel, bis sich einer der Kreise wirklich signifikant aus den anderen heraushebt, was die kreisbasierte Hough–Transform in vielen Anwendungen schlichtweg zu langsam macht. Dennoch kann man damit – im Prinzip – recht gut Kreise und Bildmittelpunkte erkennen. Eine kreisbasierte Hough–Transformation ist in `Matlab/Octave` verfügbar, es spricht also nichts dagegen, auch mal damit zu experimentieren.

Weitere Anwendungen der verallgemeinerten Hough–Transformation im Sinne von Definition 3.4 wären zwar möglich, aber nachdem mit jedem neuen Parameter die Komplexität **exponentiell** wächst, ist das alles mit Vorsicht zu geniessen.

3.2 Zeit-/Frequenz – Fenster & Gabor

Es ist natürlich schwer, ein Signal "unendlicher Dauer" oder auch nur eine sehr lange Folge $c \in \ell(\mathbb{Z}_n)$ für ein sehr großes n der Fouriertransformation zu unterziehen[107], denn selbst wenn $n \log_2 n$ als ein eher langsames Wachstum gilt überfordert die Transformation eines Musikstücks[108] auf "einen Durchgang" im-

[107] Wie bitte schreibt man das: "Fourierzutransformieren" oder "zu Fouriertransformieren" – "Fourier zu transformieren" ist jedenfalls etwas anderes.

[108] Die durchschnittliche Länge einer `wav`–Datei beträgt etwa 40 MB und selbst wenn genug Speicher zur Verfügung stehen sollte ist das Einlesen und Wegschreiben derartiger Datenmengen immer noch mit ganz immensem Aufwand verbunden!

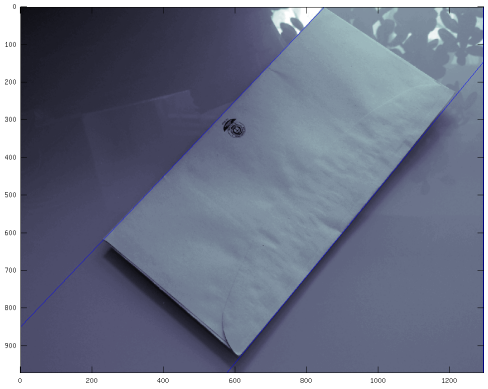


Abbildung 3.5: Noch ein Foto des Briefumschlags, diesmal wird keine der kurzen Seiten erkannt.

mer noch die Kapazitäten verfügbarer Rechner, ganz zu schweigen von kleinen Signalprozessoren, die beispielsweise in einen portablen CD- oder MP3–Player eingebaut werden sollen. Deswegen verarbeitet man nicht das gesamte Signal, sondern eben nur Stücke des Signals, die man durch einen Filterungsprozess erhält. Um eine FFT der Länge $n = 2^\ell$ zu berechnen betrachtet man dazu die *"Fenster"*

$$\ell(\mathbb{Z}_n) \ni c_k = (Fc)(\cdot + kn) = (f * c)(\cdot + kn), \qquad k \in \mathbb{Z},$$

wobei der einfachste Filter natürlich $f = \delta$ ist, was lediglich das Signal in Stücke der Länge n hackt. Aber das kann jetzt zu richtigen Schwierigkeiten führen.

Beispiel 3.6 *Wir betrachten die Funktion* $\cos(64x)$*, abgetastet an den* $n = 768 = 3 * 256$ *Punkten aus* $2\pi/768 * \mathbb{Z}_n$ *und transformiert auf den* 6 *Intervallen der Länge* 128*. In Abb. 3.6 sieht man, daß es mit der Frequenzauflösung nun dahingeht.*

Der in Abb. 3.6 dargestellte Effekt ist ein typisches Phänomen für die gefensterte Fouriertransformation: Die eigentlich scharf lokalisierten Frequenzen "laufen aus", wenn Abtastfrequenz und Fensterbreite nicht nicht mit der Periodizität der zugrundeliegenden Funktion kompatibel sind. Dieses "undichten" Frequenzen bezeichnet man auch als "Leck–Effekt", was vom englischen *"leakage phenenomenon"* stammt. Aber sehen wir uns doch erst einmal an, wo dieser Effekt herstammt. Bei den "naiven" Fenstern berechnen wir ja die Blöcke

$$\begin{aligned}
[c(\cdot + 2kn)]_n^\wedge &= \left[\tau_{2kn}\left(c \times \sum_{j\in\mathbb{Z}_n} \tau_j\delta\right)\right]_n^\wedge = \underbrace{\omega^{2kn\cdot}}_{=1}\left(c \times \sum_{j\in\mathbb{Z}_n} \tau_j\delta\right)_n^\wedge \\
&= S_{2\pi/n}\left(c \times \sum_{j\in\mathbb{Z}_n} \tau_j\delta\right)^\wedge = S_{2\pi/n}\left[\widehat{c} * \left(\sum_{j\in\mathbb{Z}_n} e^{ij\cdot}\right)\right] \\
&= \left[S_{2\pi/n}\widehat{c}\right] * \left[\sum_{j\in\mathbb{Z}_n} e^{2\pi ij\cdot/n}\right],
\end{aligned}$$

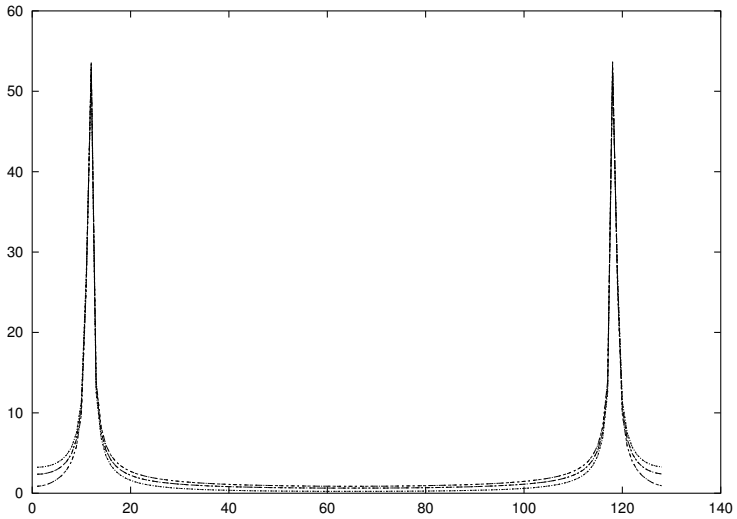


Abbildung 3.6: Absolutbeträge der Fouriertransformierten der Fenster der Länge 128. Man sieht deutlich, daß die Frequenzen "verschmiert" sind, obwohl ja eigentlich nur eine einzige Frequenz (und deren gerade Fortsetzung) auftauchen dürften. Außerdem variiert dieser Effekt mit dem jeweiligen Intervall – die Variationen sind nicht gerade dramatisch aber doch deutlich sichtbar.

und selbst wenn $\widehat{c}$ perfekt lokalisiert wäre, sorgt die Faltung mit den Exponentialfolgen für ein Verschmieren oder "Auslaufen" der Frequenzen.

Trotzdem hat die Idee einen gewissen Charme: Wenn man die Daten fenstert und nur auf diesem Bereicht betrachtet, dann sieht man im wesentlichen[109] auch nur die Frequenzen, die in diesem Bereich auftreten und kann diese lokal modifizieren. Im Zusammenhang mit Bildern haben wir das bei unserer lokalisierten DFT in Abb 2.22 ja auch nicht anders gemacht. Der einzige Unterschied besteht darin, daß wir jetzt das Signal nicht in Stücke hacken, sondern das Fenster über die Daten wandern lassen, so daß sich die betrachteten Stücke überlappen.

Die Gabor[110]–Transformation wurde 1946 von D. Gabor in[111] (Gabor, 1946) als ein Verfahren zur Analyse von Sound-Daten eingeführt und basiert auf einer reellen, um den Ursprung symmetrischen und normierten **Fensterfunktion** g, d.h. $g(t) = g(-t)$ sowie $\|g\|_2 = 1$, und den Funktionen

$$\phi_{u,\xi}(t) = e^{i\xi t}\, g(t-u), \qquad t \in \mathbb{R}, \qquad (u,\xi) \in \mathbb{R}^2, \tag{3.5}$$

aus denen sich die **Gabor–Transformation** als

$$Gf(u,\xi) = T_\Phi f(u,\xi) = \int_{\mathbb{R}} f(t)\overline{\phi_{u,\xi}}(t)\, dt = \int_{\mathbb{R}} f(t)\, e^{-i\xi t}\, g(t-u)\, dt, \qquad (u,\xi) \in \mathbb{R}^2, \tag{3.6}$$

[109] Bis auf Randartefakte, die sich vor allem im hochfrequenten Bereic zeigen.

[110] Dennis Gábor, 1900–1979, ungarischstämmiger Ingenieur, später Professor fur angewandte Elektronenphysik am Imperial College in London.

[111] Die Information stammt aus (Mallat, 1999).

ergibt. Sie ist eigentlich nichts anderes als eine **gefensterte Fouriertransformation** mit verschobenem Fenster g, und wird auch gerne als **Kurzzeit–Fouriertransformation** bezeichnet.
So weit erscheint die Gabor–Transformation noch ziemlich unauffällig; sie liefert einem ein *Spektogramm* $|Gf| : \mathbb{R}^2 \to \mathbb{R}$, das uns sagt, wieviel Energie sich zu einem bestimmten Zeitpunkt in einer bestimmten Frequenz konzentriert, aber sie kann mehr: Aus der Gabor–Transformation lässt sich die Ausgangsfunktion wieder rekonstruieren!

Satz 3.7 *Für* $f \in L_2(\mathbb{R})$ *ist*

$$f(t) = \frac{1}{2\pi} \int_{\mathbb{R}} \int_{\mathbb{R}} Gf(u,\xi)\ e^{i\xi t} g(t-u)\, d\xi du \tag{3.7}$$

und

$$\int_{\mathbb{R}} |f(t)|^2\ dt = \frac{1}{2\pi} \int_{\mathbb{R}} \int_{\mathbb{R}} |Gf(u,\xi)|^2\ d\xi du. \tag{3.8}$$

Bemerkung 3.8 *Gleichung (3.7) definiert eine inverse Gabor–Transformation, und das auch noch auf die denkbar einfachste Art und Weise, während (3.8) eine Plancherel–Variation ist, also eine Analogie zu (2.18). Die Gabor–Transformation ist auf* $L_2(\mathbb{R})$ *im wesentlichen eine Isometrie.*
Allerdings gibt es auch ein Warnung: Die Formeln gelten "nur" im L_2*–Sinn, müssen also* ***nicht*** *punktweise für alle* $t \in \mathbb{R}$ *erfüllt sein!*

Beweis: Wir fixieren einmal $\xi \in \mathbb{R}$ und betrachten die Funktionen $g_\xi = e^{i\xi\cdot} g$ sowie $f_\xi = Gf(\cdot,\xi)$. Nach Definition der Gabor–Transformation und wegen der Symmetrie von g ist

$$\begin{aligned} f_\xi(u) &= Gf(u,\xi) = \int_{\mathbb{R}} f(t) e^{-i\xi t} \underbrace{g(t-u)}_{=g(u-t)}\, dt \\ &= \int_{\mathbb{R}} f(t) e^{i\xi u} e^{i\xi(u-t)} g(u-t)\ dt = e^{i\xi u} \int_{\mathbb{R}} f(t) g_\xi(u-t)\ dt, \end{aligned}$$

also

$$f_\xi(u) = e^{i\xi u} (f * g_\xi)(u), \qquad u \in \mathbb{R}, \tag{3.9}$$

und damit auch

$$(Gf(\cdot,\xi))^\wedge(\omega) = \widehat{f_\xi}(\omega) = \widehat{f}(\omega+\xi)\widehat{g_\xi}(\omega+\xi) = \widehat{f}(\omega+\xi)\widehat{g}(\omega), \qquad \omega \in \mathbb{R}. \tag{3.10}$$

Nun wenden wir die Plancherel–Identität (2.17) auf (3.7) an und erhalten durch Einsetzen von (3.10), daß[112]

$$\frac{1}{2\pi} \int_{\mathbb{R}} \int_{\mathbb{R}} Gf(u,\xi)\ e^{i\xi t} g(t-u)\, d\xi du$$

[112] Für die Vertauschung der Integration bräuchten wir eigentlich $f \in L_1(\mathbb{R})$, aber da $L_1 \cap L_2$ ja dicht in L_1 liegt, können wir dieses "Problem" mit gutem Gewissen vernachlässigen.

$$
\begin{aligned}
&= \frac{1}{4\pi^2}\int_{\mathbb{R}} e^{i\xi t}\int_{\mathbb{R}} (Gf(\cdot,\xi))^{\wedge}(\omega)\overline{(\underbrace{g(t-\cdot)}_{=g(\cdot-t)})^{\wedge}(\omega)}\,d\omega\,d\xi\\
&= \frac{1}{4\pi^2}\int_{\mathbb{R}} e^{i\xi t}\int_{\mathbb{R}} \widehat{f}(\omega+\xi)\widehat{g}(\omega)\,\overline{e^{-it\omega}\widehat{g}(\omega)}\,d\omega\,d\xi\\
&= \frac{1}{4\pi^2}\int_{\mathbb{R}}\int_{\mathbb{R}} \widehat{f}(\omega+\xi)\,|\widehat{g}(\omega)|^2\,e^{it(\xi+\omega)}\,d\omega\,d\xi\\
&= \frac{1}{2\pi}\int_{\mathbb{R}} |\widehat{g}(\omega)|^2\,\underbrace{\frac{1}{2\pi}\int_{\mathbb{R}} \widehat{f}(\omega+\xi)e^{it(\omega+\xi)}\,d\xi}_{(\widehat{f})^{\vee}(t)=f(t)}\,d\omega\\
&= f(t)\,\frac{1}{2\pi}\int_{\mathbb{R}} |\widehat{g}(\omega)|^2\,d\omega = f(t)\int_{\mathbb{R}} |g(t)|^2\,dt = f(t)\,\|g\|_2^2 = f(t),
\end{aligned}
$$

was dann auch schon (3.7) beweist. Für (3.8) verwenden wir ebenfalls wieder (3.10), um

$$
\begin{aligned}
&\frac{1}{2\pi}\int_{\mathbb{R}}\int_{\mathbb{R}} |Gf(u,\xi)|^2\,d\xi du\\
&= \frac{1}{4\pi^2}\int_{\mathbb{R}}\int_{\mathbb{R}} \left|(Gf(\cdot,\xi))^{\wedge}(\omega)\right|^2\,d\xi d\omega\\
&= \frac{1}{4\pi^2}\int_{\mathbb{R}} |\widehat{g}(\omega)|^2\int_{\mathbb{R}} \left|\widehat{f}(\omega+\xi)\right|^2\,d\xi\,d\omega = \frac{1}{4\pi^2}\int_{\mathbb{R}} |\widehat{g}(\omega)|^2\,d\omega\int_{\mathbb{R}} \left|\widehat{f}(\xi)\right|^2\,d\xi\\
&= \int_{\mathbb{R}} |g(t)|^2\,dt\int_{\mathbb{R}} |f(t)|^2\,dt = \|f\|_2^2\,\|g\|_2^2 = \|f\|_2^2
\end{aligned}
$$

zu erhalten – nichts anderes als (3.8). □

Bemerkung 3.9 *Die Formel (3.10) für die Fouriertransformierte*[113] *von* Gf *bezüglich* u *ist nicht nur ein nützliches Hilfsmittel zum Beweis von Satz 3.7, sondern vor allem auch der Schlüssel zu einer* effizienten Implementierung. *Nehmen wir nämlich die inverse Fouriertransformierte von (3.10), dann erhalten wir, daß*

$$Gf(u,\xi) = \left(\widehat{f}(\cdot+\xi)\,\widehat{g}\right)^{\vee}(u) \tag{3.11}$$

und bei diskreter Abtastung[114] *von* f *und* g *kann man die Fouriertransformierten schnell und effizient mit der FFT bestimmen und dann das komponentenweise Produkt ebenso schnell rücktransformieren, siehe z.B. (FFTW, 2003; Sauer, 2003).*

Die Gabor–Transformation ist ein erstes Beispiel für **Zeit–Frequenz–Analyse**, bei der man versucht, beiden Aspekten *simultan* gerecht zu werden, siehe Abb. 3.7. Die dort gezeigte Notenschrift legt ja bekanntlich gerade fest, welche Ton, also welche Frequenz, zu einem bestimmten Zeitpunkt zu erklingen hat. Ein perfekt gestimmtes und perfekt gespieltes Instrument würde also ein in Zeit und Frequenz perfekt lokalisiertes Signal produzieren. Zumindest mehr oder weniger. Um das genauer sagen zu können, brauchen wir zuerst einmal ein wenig Terminologie.

[113] Erinnert ein wenig an das, was wir auch bei der Radontransformation gemacht haben.

[114] Letztere kann man dann sogar in einer Tabelle speichern, denn das Fenster wird ja normalerweise irgendwann mal fest gewählt.

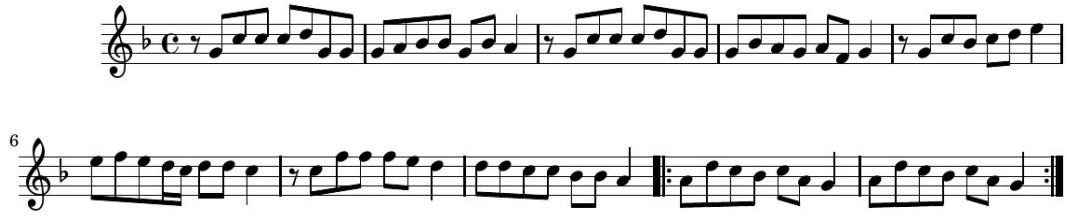


Abbildung 3.7: Eine Zeit-/Frequenz–Darstellung. Die Lage der „Punkte“ gibt die Tonhöhe, also die Frequenz an, die Form der Punkte die Tondauer und damit auch die Zeit, zu der der Ton zu klingen hat. Notensatz mit `lilypond`.

Definition 3.10 (Zeit-/Frequenzlokalisierung) *Die **Zeitlokalisierung** einer Funktion*[115] $f \in L_2(\mathbb{R})$ *ist als*

$$\mu = \mu(f) = \frac{1}{\|f\|_2^2} \int_{\mathbb{R}} t\,|f(t)|^2 \,\mathrm{d}t, \tag{3.12}$$

*definiert, die **Frequenzlokalisierung** von f entsprechend als*

$$\widehat{\mu} = \widehat{\mu}(f) = \frac{1}{\|\widehat{f}\|_2^2} \int_{\mathbb{R}} \xi \left|\widehat{f}(\xi)\right|^2 \,\mathrm{d}\xi = \frac{1}{2\pi\,\|f\|_2^2} \int_{\mathbb{R}} \xi \left|\widehat{f}(\xi)\right|^2 \,\mathrm{d}\xi. \tag{3.13}$$

*Die **Zeitvariation** und die **Frequenzvariation** sind schließlich*

$$\sigma^2 = \frac{1}{\|f\|_2^2} \int_{\mathbb{R}} (t-\mu)^2\,|f(t)|^2 \,\mathrm{d}t, \qquad \widehat{\sigma^2} = \frac{1}{2\pi\,\|f\|_2^2} \int_{\mathbb{R}} (\xi-\widehat{\mu})^2 \left|\widehat{f}(\xi)\right|^2 \,\mathrm{d}\xi, \tag{3.14}$$

Bemerkung 3.11 *Der Name “Lokalisierung” für (3.12) erklärt sich ganz einfach: Ist $f \sim \delta_x$ für ein $x \in \mathbb{R}$, also konzentriert sich die ganze Masse von f um den Punkt x, sagen wir* $\operatorname{supp} f \subset [x-\varepsilon, x+\varepsilon]$, $\varepsilon > 0$, *dann ist*

$$\begin{aligned} |\mu - x| &= \frac{1}{\|f\|_2^2} \left| \int_{\mathbb{R}} (t-x)\,|f(t)|^2 \,\mathrm{d}t \right| \le \frac{1}{\|f\|_2^2} \int_{x-\varepsilon}^{x+\varepsilon} |t-x|\,|f(t)|^2 \,\mathrm{d}t \\ &\le \frac{\varepsilon}{\|f\|_2^2} \int_{\mathbb{R}} |f(t)|^2 \,\mathrm{d}t = \varepsilon, \end{aligned}$$

*also $\mu \sim x$. Generell könnte man μ und $\widehat{\mu}$ auch als **Zeitschwerpunkt** bzw. **Frequenzschwerpunkt** von f bezeichnen.*

[115] Die Funktionenräume kann man im Kontext der Bildverarbeitung normalerweise eher locker sehen, eher so im Sinne von *„Die Menge aller Funktionen, für die die Rechnungen korrekt sind.“* Bei diskreten Implementierungen ist dann sowieso wieder alles ganz anders, **da** muss man sich dann allerdings Gedanken machen.

Eine Funktion hat also ihren Schwerpunkt in Zeit/Frequenz an der Stelle $(\mu, \widehat{\mu})$ und die Variationen $\sigma_t, \widehat{\sigma}$ geben an, wie gut konzentriert oder eben lokalisiert die Funktion in Zeit und Frequenz ist. Den Bereich

$$H(f) := [\mu(f) - \sigma(f), \mu(f) + \sigma(f)] \times [\widehat{\mu}(f) - \widehat{\sigma}(f), \widehat{\mu}(f) + \widehat{\sigma}(f)] \tag{3.15}$$

nennt man **Heisenberg–Box** oder **Heisenberg–Rechteck** zu f, und er beschreibt, wie stark die Funktion in Zeit und Frequenz streut oder verschmiert. Daß diese Bereiche nicht beliebig klein werden können, besagt der folgende Klassiker, dessen kurzen Beweis wir uns gleich mit ansehen können.

Satz 3.12 (Heisenbergsche Unschärferelation) *Für $f \in L_2(\mathbb{R})$ ist*

$$\sigma(f)\,\widehat{\sigma}(f) \geq \frac{1}{2}. \tag{3.16}$$

Insbesondere ist sowohl $\sigma(f) = 0$ als auch $\widehat{\sigma}(f) = 0$ unmöglich.

Beweis: Der Beweis aus (Mallat, 1999) folgt einem "Original" von H. Weyl und macht die etwas stärkere Annahme, daß

$$\lim_{t\to\infty} \sqrt{t} f(t) = 0$$

ist. Da diese Funktionen aber dicht in $L_2(\mathbb{R})$ liegen, ist das keine wirkliche Einschränkung[116]. Ersetzt man ausserdem f durch $g := e^{-i\widehat{\mu}\cdot} f(\cdot + \mu)$, dann ist $\|g\| = \|f\|$ und $\mu_t(g) = \widehat{\mu}(g) = 0$ ist, was wir eigentlich auch gleich von f annehmen können.

Dann gilt mit ein bisschen Manipulationen an den Fouriertransformierten und der Schwarzschen Ungleichung, daß[117]

$$\begin{aligned}
\sigma_t^2(f)\,\widehat{\sigma}^2(f) &= \frac{1}{2\pi\|f\|_2^4} \int_{\mathbb{R}} |t\,f(t)|^2 \,dt \int_{\mathbb{R}} \underbrace{\left|\xi \widehat{f}(\xi)\right|^2}_{=\left|i\xi\widehat{f}(\xi)\right|^2} d\xi \\
&= \frac{1}{2\pi\|f\|_2^4} \int_{\mathbb{R}} |t\,f(t)|^2 \,dt \int_{\mathbb{R}} \left|(f')^\wedge(\xi)\right|^2 d\xi = \frac{1}{\|f\|_2^4} \int_{\mathbb{R}} |t\,f(t)|^2 \,dt \int_{\mathbb{R}} |f'(t)|^2 \,dt \\
&\geq \frac{1}{\|f\|_2^4} \left(\int_{\mathbb{R}} |t\,f(t)\,f'(t)| \,dt\right)^2 \geq \frac{1}{\|f\|_2^4} \left(\int_{\mathbb{R}} \frac{t}{2} \left(f^2(t)\right)' dt\right)^2 \\
&= \frac{1}{4\|f\|_2^4} \left(\int_{\mathbb{R}} t \left(f^2(t)\right)' dt\right)^2 = \frac{1}{4\|f\|_2^4} \left(\left[t f^2(t)\right]_{t=0}^{\infty} - \int_{\mathbb{R}} |f(t)|^2 \,dt\right)^2 \\
&= \frac{1}{4\|f\|_2^4} \left(\int_{\mathbb{R}} |f(t)|^2 \,dt\right)^2 = \frac{1}{4},
\end{aligned}$$

ganz wie in (3.16) behauptet. □

[116] „Dicht" bedeutet, daß jedes $f \in L_2(\mathbb{R})$ Grenzwert einer Folge von solchen „braven" Funktionen ist und wenn für alle Elemente der Folge (3.16) erfüllt ist, dann eben auch für den Grenzwert.

[117] Wir nehmen hier stillschweigend an, daß f reell ist, dann sparen wir uns in den Rechnungen einiges an komplexer Konjugation.

Bemerkung 3.13 (Heisenbergsche Unschärferelation)

1. *Die „Musikervariante" von Satz 3.12 ist:*

   > *Man kann auf dem Basspedal einer Orgel keinen Jig spielen.*

   Ein schnelles Musikstück verlangt eine gute Zeitlokalisierung, also einen sehr geringen Wert von $\sigma(f)$*, weswegen* $\widehat{\sigma} \geq (2\sigma(f))^{-1}$ *automatisch groß werden muss. Bei einer hohen Frequenz ist das aber wesentliche unproblematischer als bei einer niedrigen Frequenz, denn eine Tonhöhenvariation ist ja immer ein* relativer *und kein absoluter Fehler*[118]

2. *So gesehen ist die Heisenbergsche Unschärferelation wirklich ein fundementales Resultat, das uns sagt, daß unsere Tonwahrnehmung aus prinzipiellen Gründen beschränkt ist.*

3. *Und noch eine Interpretation, diesmal aber im Sinne von (3.15):*

   > *Jede Heisenberg–Box hat mindestens die Fläche* 2*.*

4. *Bei Ungleichungen wie (3.16) ist es ganz wichtig, zu wissen, für welche Funktionen Gleichheit eintritt, denn das sind ja in diesem Fall die Funktionen mit* ***bester*** *Zeit-/Frequenz–Lokalisierung. Diese extremalen Funktionen kann man im Falle der Heisenbergschen Unschärferelation sogar explizit angeben, es sind gerade alle Funktionen der Form*

$$f(t) = a\, e^{i\omega t - b\,(t-u)^2} = a\, e^{i\omega t}\, e^{-b(t-u)^2}, \qquad u, \omega \in \mathbb{R},\ a, b \in \mathbb{C}. \tag{3.17}$$

   Derartige Funktionen nennt man[119] *auch gerne* ***modulierte Gauß–Funktionen****.*

3.3 Wavelets

Eine moderne und leistungsfähige Methode zur Zeit-/Frequenz–Analyse ist die kontinuierliche Wavelet–Transformation, mit der wir uns nun beschäftigen wollen. Man muss bei Wavelets wieder einmal ein bisschen aufpassen, da sie überall in der "Standardliteratur", beispielsweise (Daubechies, 1992; Holschneider, 1995; Louis *et al.*, 1998; Mallat, 1999), ein wenig anders eingeführt und behandelt werden. Apropos Literatur: Einen sehr fouilletonistischen bzw. journalistischen, aber durchaus korrekten und definitiv sehr lesbaren und lesenswerten Einstieg in die Materie gibt (Hubbard, 1996).

[118] Eine Abweichung von 10 cent (das in der Musik übliche Maß der Stimmgenaugkeit) bedeutet ja für die Frequenzen ω, ω', daß $2^{-1/120}\omega \leq \omega' \leq 2^{1/120}\omega$, bzw. $\omega' \in \left[2^{-1/120}, 2^{1/120}\right]\omega$ und die Breite dieses Toleranzbereichs hängt ganz offensichtlich von ω ab!

[119] Wegen des $e^{i\omega t}$–Terms vorne, den nennt man eben auch **Modulation**.

Definition 3.14 (Wavelets und Zulässigkeit) *Eine möglicherweise komplexertige Funktion* $\psi \in L_2(\mathbb{R})$ *heisst* Wavelet[120], *wenn sie* mittelwertfrei *ist , d.h., wenn*

$$\int_{\mathbb{R}} \psi(t)\,dt = 0 \tag{3.18}$$

gilt. Ein Wavelet heißt normalisiert, *wenn* $\|\psi\|_2 = 1$ *und* zulässig, *wenn*

$$C_\psi := \int_{\mathbb{R}} \frac{\left|\widehat{\psi}(\xi)\right|^2}{|\xi|}\,d\xi < \infty \tag{3.19}$$

ist.

Der Name "Wavelet" für eine mittelwertfreie Funktion begründet sich dadurch, daß diese Funktion Anteile oberhalb und unterhalt der x–Achse haben muss und daher zumindest eine "wellenähnliche" Struktur hat.

Die Zulässigkeitsbedingung (3.19) verlangt insbesondere, daß

$$0 = \widehat{\psi}(0) = \int_{\mathbb{R}} \psi(t)\,dt$$

ist, jede zulässige Funktion ist also automatisch ein Wavelet, weswegen es sinnvoll ist, überhaupt nur von zulässigen Wavelets und nicht von zulässigen Funktionen zu sprechen. Ist außerdem ψ ein *relles Wavelet*, als $\psi : \mathbb{R} \to \mathbb{R}$, dann ist ja[121]

$$\widehat{\psi}(-\xi) = \int_{\mathbb{R}} \psi(t)\,e^{i\xi t}\,dt = \int_{\mathbb{R}} \overline{\psi(t)\,e^{-i\xi t}}\,dt = \overline{\widehat{\psi}(\xi)}$$

und (3.19) vereinfacht sich ein wenig zu

$$\begin{aligned} \infty &> \int_{\mathbb{R}} \frac{\left|\widehat{\psi}(\xi)\right|^2}{|\xi|}\,d\xi = \int_0^\infty \frac{\left|\widehat{\psi}(\xi)\right|^2}{|\xi|}\,d\xi + \int_0^\infty \frac{\left|\widehat{\psi}(-\xi)\right|^2}{|-\xi|}\,d\xi \\ &= 2\int_0^\infty \frac{\left|\widehat{\psi}(\xi)\right|^2}{\xi}\,d\xi, \end{aligned}$$

also zu

$$C_\psi = \int_0^\infty \frac{\left|\widehat{\psi}(\xi)\right|^2}{\xi}\,d\xi < \infty. \tag{3.20}$$

Definition 3.15 *Zu einem normalisierten Wavelet*[122] ψ *und* $f \in L_2(\mathbb{R})$ *ist die* Wavelettransformation *als*

$$W_\psi f(u,s) := \int_{\mathbb{R}} f(t)\,\frac{1}{\sqrt{|s|}}\overline{\psi\left(\frac{t-u}{s}\right)}\,dt, \qquad (u,s) \in \Gamma = \mathbb{R}\times\mathbb{R}, \tag{3.21}$$

definiert.

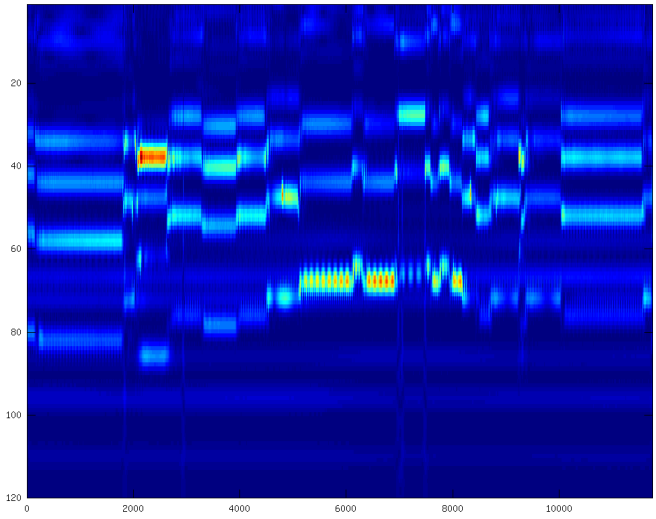


Abbildung 3.8: **Skalogramm** eines Melodiefragments der Melodie aus Abb 3.7, Abtastfrequenz 8000Hz. Das Skalogramm plottet $\left|W_\psi f(u,s)\right|$ gegen u und s. Verwendet wurde hierbei das Morlet–Wavelet aus Beispiel 3.16

Der etwas sperrige Term $1/\sqrt{|s|}$ sorgt dafür, daß auch das skalierte Wavelet

$$\psi_s := \frac{1}{\sqrt{|s|}}\psi\left(s^{-1}\cdot\right), \qquad s \in \mathbb{R}, \tag{3.22}$$

normiert ist:

$$\|\psi_s\|_2 = \frac{1}{|s|}\int_{\mathbb{R}} |\psi(t/s)|^2 \, dt = \int_{\mathbb{R}} |\psi(t)|^2 \, dt = \|\psi\|_2 = 1.$$

Schauen wir uns nun zuerst einmal die "klassischen" Beispiele für Wavelets an.

Beispiel 3.16 (Wavelets)

1. *Das **Haar–Wavelet** ist die unstetige Funktion*

$$\psi := \chi(\cdot + 1) - \chi = \begin{cases} 1, & x \in [-1,0), \\ -1, & x \in (0,1], \\ 0, & \textit{sonst}, \end{cases}$$

wobei wieder $\chi = \chi_{[0,1]}$ ist. Das Haar–Wavelet ist das einzige unter den "klassischen" Wavelets, das kompakten Träger hat.

[120] Ursprünglich *"Ondelette"*, was auch nichts anderes als *"Wellchen"* bedeutet.
[121] Diese Symmetrie gilt für die Fouriertransformierte *jeder* reellwertigen Funktion!
[122] Die Definition eines Wavelets beinhaltet automatisch $\psi \in L_2(\mathbb{R})$.

2. *Das **Mexican–Hat–Wavelet**[123] ist als*

$$\psi(t) := \left(1 - t^2\right) e^{-t^2/2} = -\frac{d^2}{dt^2} e^{-t^2/2}, \qquad t \in \mathbb{R},$$

definiert. Es hat keinen kompakten Träger mehr, klingt aber für $x \to \infty$ exponentiell ab und verfügt daher immer noch über gute Zeitlokalisierung. Die zugehörige Fouriertransformation ist

$$\widehat{\psi}(\xi) = \sqrt{2\pi}\, \xi^2\, e^{-\xi^2/2}$$

und stimmt damit im wesentlichen mit dem Wavelet selbst überein.

3. *Das **Morlet–Wavelet** bzw. "Morlet's Gaussian wavelet", siehe (Holschneider, 1995), ist der komplexe Bruder des Mexican Hat und kombiniert dessen Abklingrate mit einer Phasenmodulation:*

$$\psi(t) = e^{i\omega t}\, e^{-t^2/2}, \qquad t \in \mathbb{R}, \qquad \omega \in \mathbb{R}_+.$$

Die Oszillationsfrequenz ω und t_0 ist ein Parameter, den man frei wählen kann und mit dem sich steuern, wie oft das Wavelet oszilliert, bis es so weit abgeklungen ist, da man es eigentlich nicht mehr sieht. Genau genommen ist das Morlet–Wavelet, so wie es hier steht gar kein Wavelet, da $\int \psi \neq 0$ ist[124]*, aber das lässt sich relativ einfach durch einen passenden Korrekturterm beheben.*

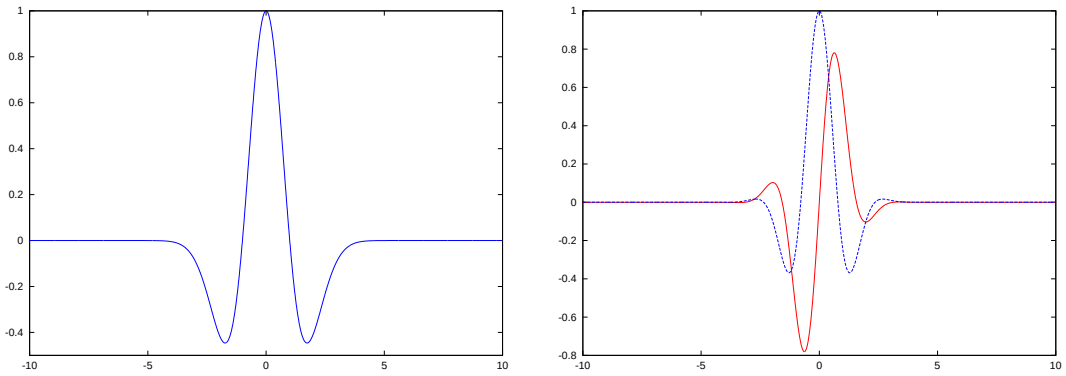


Abbildung 3.9: Der Mexican Hat *(links)* und das Morlet–Wavelet *(rechts)*, wobei Real– und Imaginärteil separat geplottet sind.

[123] Ins Deutsche auch gerne als "Mexikanischer Hut" übersetzt, obwohl im Deutschen eigentlich dann der spanisch/mexikanische Begriff *Sombrero* angemessener wäre. Das kommt davon, wenn man weder Englisch noch seine eigene Sprache so richtig spricht.

[124] Genauer gesagt: Es ist kein zulässiges Wavelet, aber da sind wir schon wieder in der Welt der Sprechweisen, denn in (Holschneider, 1995) heißt es, es wäre kein "progressives Wavelet".

Bemerkung 3.17 *Die "Grundfunktion" $e^{-(\cdot)^2/2}$ beim Mexican Hat und dem Morlet–Wavelet ist natürlich kein Zufall. Wenn wir uns am Bemerkung 3.13 erinnern, hat diese Funktion gerade die dort angegebene Form (3.17) und besitzt somit die optimale Zeit-/Frequenz–Auflösung. Damit ist sie natürlich der Ideale Kandidat, um in der Wavelet–Ursuppe zu schwimmen.*

Übung 3.3 Bestimmen Sie die Fouriertransformierte des Haar–Wavelets und des Mexican Hat. ◊

Für zulässige Funktionen ist auch diese Transformation wieder invertierbar, nur mit einem etwas anderen Integral.

Satz 3.18 *Für ein normalisiertes, zulässiges Wavelet ψ und $f \in L_2(\mathbb{R})$ ist*

$$f(t) = \frac{1}{C_\psi} \int_{\mathbb{R}} \int_{\mathbb{R}} W_\psi f(u,s) \, \frac{1}{\sqrt{|s|}} \psi\left(\frac{t-u}{s}\right) du \frac{ds}{s^2} \tag{3.23}$$

Beweis: Zuerst bemerken wir noch schnell[125], daß die Konjugation einer komplexwertigen Funktion f die Fouriertransformation

$$\left(\overline{f}\right)^\wedge(\xi) \;=\; \int_{\mathbb{R}} \overline{f(t)} e^{-i\xi t} \, dt = \overline{\int_{\mathbb{R}} f(t) \, e^{i\xi t} \, dt} = \overline{\widehat{f}(-\xi)}$$

hat.

Im ersten "richtigen" Schritt unseres Beweises bestimmen wir wieder wie im Beweis von Satz 3.7 eine Fouriertransformierte der Wavelettransformation, nämlich

$$\begin{aligned} (W_\psi f(\cdot, s))^\wedge(\xi) &= \left(\int_{\mathbb{R}} f(t) \, \overline{\psi_s(t-\cdot)} \, dt \right)^\wedge(\xi) \\ &= \left(f * \overline{\psi_s(-\cdot)} \right)^\wedge(\xi) = \widehat{f}(\xi) \, \widehat{\psi}_s(\xi) \end{aligned}$$

also

$$(W_\psi f(\cdot, s))^\wedge(\xi) = \sqrt{s} \, \widehat{f}(\xi) \, \widehat{\psi}(s\xi), \qquad \xi \in \mathbb{R}, \quad s \in \mathbb{R}, \tag{3.24}$$

und somit, für $s \in \mathbb{R}$,

$$\begin{aligned} &\int_{\mathbb{R}} W_\psi f(u,s) \, \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) du \\ &= \frac{1}{2\pi} \int_{\mathbb{R}} (W_\psi f(\cdot, s))^\wedge(\xi) \; (\psi_s(t-\cdot))^\wedge(\xi) \, du \\ &= \frac{1}{2\pi} \int_{\mathbb{R}} |s| \, \widehat{f}(\xi) \, \widehat{\psi}(s\xi) e^{i\xi t} \overline{\widehat{\psi}(s\xi)} d\xi = \frac{|s|}{2\pi} \int_{\mathbb{R}} e^{i\xi t} \widehat{f}(\xi) \left| \widehat{\psi}(s\xi) \right|^2 d\xi. \end{aligned}$$

[125] Da wir im "Fourierkapitel" ja immer nur reellwertige Funktionen betrachtet haben, müssen wir halt jetzt schnell in den saueren Apfel beissen.

Mit einer Integrationsvertauschung und der Variablentransformation $\omega = s\xi$ bekommen wir so für das volle Integral

$$\begin{aligned}
&\frac{1}{C_\psi}\int_{\mathbb{R}}\int_{\mathbb{R}} W_\psi f(u,s)\,\frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right)\mathrm{d}u\frac{\mathrm{d}s}{s^2}\\
&= \frac{1}{2\pi\,C_\psi}\int_{\mathbb{R}}\widehat{f}(\xi)e^{i\xi t}\int_{\mathbb{R}}\frac{\left|\widehat{\psi}(s\xi)\right|^2}{|s|}\,\mathrm{d}s\,\mathrm{d}\xi = \frac{1}{2\pi\,C_\psi}\underbrace{\int_{\mathbb{R}}\widehat{f}(\xi)e^{i\xi t}\,\mathrm{d}\xi}_{=\left(\widehat{f}\right)^\vee(t)=f(t)}\underbrace{\int_{\mathbb{R}}\frac{\left|\widehat{\psi}(\omega)\right|^2}{|\omega|}\,\mathrm{d}\omega}_{=C_\psi}\\
&= f(t),
\end{aligned}$$

genau wie behauptet. □

Mit genau denselben Methoden und unter Berücksichtigung der Tatsache, daß für relle ψ die Beziehung $\left|\widehat{\psi}(-\xi)\right|^2 = \left|\widehat{\psi}(\xi)\right|^2$ gilt, erhält man für *relle* Wavelets eine etwas einfachere Invertierungsformel, siehe (Mallat, 1999).

Korollar 3.19 *Für ein* reelles *Wavelet* ψ *und* $f \in L_2(\mathbb{R})$ *gilt*

$$f(t) = \frac{1}{C_\psi}\int_0^\infty\int_{\mathbb{R}} W_\psi f(u,s)\,\frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right)\mathrm{d}u\frac{\mathrm{d}s}{s^2}, \qquad C_\psi = \int_0^\infty \frac{\left|\widehat{\psi}(\xi)\right|^2}{\xi}\,\mathrm{d}\xi. \quad (3.25)$$

Bemerkung 3.20 (Wavelettransformation & Umkehrformel)

1. *Natürlich gilt auch die Umkehrformel für die Wavelettransformation nur im* L_2*–Sinn, also eigentlich nicht punktweise. Außerdem haben wir im Beweis ziemlich sorglos Integrationsgrenzen vertauscht, beinahe durch Null dividiert und so weiter. "Genauere" Versionen der Umkehrformel mit Beweisen finden sich beispielsweise in (Daubechies, 1992) oder (Louis* et al.*, 1998).*

2. *Aus (3.24) kann man auch sehr schön die Redundanz der Wavelettransformation sehen, denn für* $s, s' \in \mathbb{R}_+$ *und* $\xi \in \mathbb{R}$ *ist*

$$\widehat{f}(\xi) = \frac{(W_\psi f(\cdot,s))^\wedge(\xi)}{\sqrt{s}\widehat{\psi}(s\xi)} = \frac{(W_\psi f(\cdot,s'))^\wedge(\xi)}{\sqrt{s'}\widehat{\psi}(s'\xi)},$$

also

$$(W_\psi f(\cdot,s))^\wedge(\xi) = \sqrt{\frac{s}{s'}}\,\frac{\widehat{\psi}(s\xi)}{\widehat{\psi}(s'\xi)}\,(W_\psi f(\cdot,s'))^\wedge(\xi). \quad (3.26)$$

Mit anderen Worten:

> *Kennt man die Wavelettransformierte für eine Skala* s*, dann kennt man sie eigentlich schon für alle* s*.*

3. *Das Auftauchen der des $1/s^2$–Terms in der Umkehrformel mag zuerst ein wenig überraschen und man kann sich natürlich auf den Standpunkt stellen, daß damit halt einfach der Beweis funktioniert und das war's. Das ist aber noch nicht einmal die halbe Wahrheit, denn es gibt durchaus gute Gründe, warum das so sein* ***muss****, siehe (Grossmann* et al.*, 1985) und auch (Sauer, 2012). Das hat einiges mit abstrakter harmonischer Analysis, dualen Gruppen und der affinen Gruppe zu tun, und zeigt, daß es sich durchaus lohnt, Mathematik zu kennen und zu verstehen.*

Als nächstes wollen wir uns schnell die Heisenberg–Rechtecke bei der Wavelettransformation ansehen, wozu wir die **Zeit–Frequenz–Atome**

$$\psi_{u,s} = \psi_s(\cdot - u) = \psi\left(\frac{\cdot - u}{s}\right), \qquad (u, s) \in \mathbb{R} \times \mathbb{R}_+$$

betrachten müssen. Da für beliebiges $f \in L_2(\mathbb{R})$ und $u \in \mathbb{R}$

$$\mu_t\left(f(\cdot - u)\right) = \int_{\mathbb{R}} t\,|f(t-u)|^2\,dt = \int_{\mathbb{R}} (t+u)\,|f(t)|^2\,dt = \mu_t(f) + u$$

ist, können wir durch eine geeignete Verschiebung immer annehmen, daß das Wavelet ψ **zentriert** ist, daß also $\mu_t(\psi) = 0$ gilt. Die Frequenzlokalisierung $\mu_\xi(\psi)$ ist hingegen eine Konstante, die uns die "Grundoszillation" des Wavelets angibt.

Proposition 3.21 *Das Heisenberg–Rechteck $H(\psi_{u,s})$ zu einem zentrierten Wavelet ψ hat den Mittelpunkt $(u, s^{-1}\mu_\xi(\psi))$ sowie die Seitenlängen $s\sigma_t(\psi)$ und $\sigma_\xi(\psi)/s$.*

Beweis: Dann rechnen wir halt:

$$\begin{aligned}
\mu_t(\psi_{u,s}) &= \int_{\mathbb{R}} t\left|\frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right)\right|^2 dt = \frac{1}{s}\int_{\mathbb{R}} (t+u)\left|\psi\left(\frac{t}{s}\right)\right|^2 dt \\
&= \int_{\mathbb{R}} st\,|\psi(t)|^2\,dt + u\int_{\mathbb{R}} |\psi(t)|^2\,dt = s\mu_t(\psi) + u = u,
\end{aligned}$$

sowie für $s > 0$[126]

$$\begin{aligned}
\mu_\xi(\psi_{u,s}) &= \int_{\mathbb{R}} \xi\left|\widehat{\psi}_{u,s}\right|^2 d\xi = \int_{\mathbb{R}} \xi\left|e^{i\xi u}\widehat{\psi}_s\right|^2 d\xi = \int_{\mathbb{R}} \xi\left|\sqrt{s}\widehat{\psi}(s\xi)\right|^2 d\xi \\
&= \int_{\mathbb{R}} s\xi\left|\widehat{\psi}(s\xi)\right|^2 d\xi = \frac{1}{s}\int_{\mathbb{R}} \xi\left|\widehat{\psi}(\xi)\right|^2 d\xi = \frac{\mu_\xi(\psi)}{s}.
\end{aligned}$$

Für die Varianzen erhalten wir entsprechend

$$\begin{aligned}
\sigma_t^2(\psi_{u,s}) &= \int_{\mathbb{R}} (t-\mu_t)^2\left|\frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right)\right|^2 dt \\
&= \int_{\mathbb{R}} (t-u)^2\left|\frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right)\right|^2 dt = \frac{1}{s}\int_{\mathbb{R}} t^2\left|\psi\left(\frac{t}{s}\right)\right|^2 dt \\
&= \int_{\mathbb{R}} (st)^2\,|\psi(t)|^2\,dt = s^2\,\sigma_t^2(\psi)
\end{aligned}$$

[126] Für den ohnehin nicht so interessanten und leicht kontraintuitiven Fall $s < 0$ muss man ein paar Betragsstriche geeignet setzen.

und

$$\begin{aligned}\sigma_\xi^2(\psi_{u,v}) &= \int_{\mathbb{R}} \left(\xi - s^{-1}\mu_\xi\right)^2 \left|\widehat{\psi}_{u,s}(\xi)\right|^2 \, d\xi = s \int_{\mathbb{R}} \left(\xi - s^{-1}\mu_\xi\right)^2 \left|\widehat{\psi}(s\xi)\right|^2 \, d\xi \\ &= \int_{\mathbb{R}} \left(\frac{\xi - \mu_\xi}{s}\right)^2 \left|\widehat{\psi}(\xi)\right|^2 \, d\xi = \frac{1}{s^2} \int_{\mathbb{R}} (\xi - \mu_\xi)^2 \left|\widehat{\psi}(\xi)\right|^2 \, d\xi = \frac{\sigma_\xi^2(\psi)}{s^2}.\end{aligned}$$

□

Übung 3.4 Zeigen Sie: Die Heisenberg–Rechtecke zu den Atomen $\phi_{u,\xi}$ der Gabortransformation sind von der Form

$$H_{u,\xi} = [u - \sigma(\phi), u + \sigma(\phi)] \times [\xi - \widehat{\sigma}(\phi), \xi + \widehat{\sigma}(\phi)] \,.$$

♢

Bemerkung 3.22 *Proposition 3.21 zeigt sehr schön den fundamentalen Unterschied zwischen der Gabortransformation und der Wavelettransformation: Während die Gabortransformation mit* ***konstanter*** *Zeit- und Frequenzauflösung arbeitet, verwendet die Wavelettransformation eine* ***relative*** *Zeit- und Frequenzauflösung. In Bereichen hoher Frequenz, also kleiner Werte des Skalierungsparameters* s *ist die Zeitauflösung schärfer, dafür aber die* absolute Frequenzauflösung *geringer während die* relative Frequenzauflösung *konstant bleibt. In Bereichen niedriger Frequenz ist es genau umgekehrt, hier wird die Zeit unschärfer aufgelöst*[127] *während die Frequenzen genauer lokalisiert werden.*

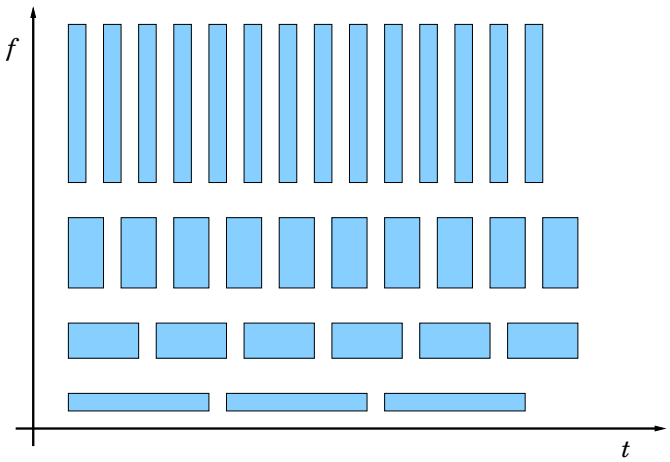


Abbildung 3.10: Die (schematische) Darstellung der Heisenberg–Boxen zu einem Wavelet. Nach „oben“, also zu höheren Frequenzen und damit *kleineren* Skalen werden die Rechtecke schmaler, also die Zeitauflösung höher, nach unten ist es umgekehrt.

[127] Was unserer Philosophie entspricht: Ein Ton niedriger Frequenz braucht einfach ein klein wenig länger, um wenigstens eine Schwingung hinzubekommen.

Abb 3.10 zeigt die Heisenberg–Boxen zu einem Wavelet. Diese haben nach Proposition 3.21 alle dieselbe Fläche

$$s\sigma(\psi)\,\frac{\widehat{\sigma}(\psi)}{s} = \sigma(\psi)\,\widehat{\sigma}(\psi),$$

sind aber im niederfrequenten Bereich, also für große Skalen, eher flach und breit, im hochfrequenten Bereich niedriger Skalen[128] dagegen sind die Heisenberg–Boxen[129] hoch und schmal. Man kann das eigentlich sehr schön in der folgenden Faustregel zusammenfassen:

> Wavelets bieten eine **relative Genauigkeit** in der Zeit- und Frequenzlokalisierung.

Um nun eine möglichst gleichmäßige Überdeckung der Ebene mit solchen Rechtecken zu erhalten, ist es naheliegend die Skalen *geometrisch* zu wählen, also

$$s_j = s_0\,\sigma^j, \qquad j \in \mathbb{Z}_M, \qquad \sigma > 1, \tag{3.27}$$

für eine **Ausgangsskala** s_0 und eine **Skalenprogression** σ zu setzen. Das hat einen Menge von Vorteilen, die sich auch tatsächlich numerisch begründen lassen, siehe (Klein, 2011). Als nette Randbemerkung ist (3.27) übrigens auch als die temperierte Skala musikalischer Intervalle interpretierbar.

Bleibt noch die Frage, wie wir s_0, σ und M wählen sollten. Dazu folgende Faustregeln:

1. Die untere Skala s_0 entspricht der höchsten auftretenden Frequenz. Diese Frequenz sollte entsprechend dem Shannonschen Abtastsatz, Satz 2.16, so niedrig sein, daß die Abtastung noch die Voraussetzungen des Satzes erfüllt. Dabei ist zu beachten, daß die numerische Stabilität natürlich umso mehr leidet, je mehr man sich der kritischen Frequenz annähert. Zumindest das Zentrum der Heisenberg–Box zu s_0 sollte also noch im zulässigen Frequenzbereich liegen, besser noch die ganze Box.

2. Die obere Skala $s_{M-1} = s_0\sigma^{M-1}$ sollte so klein bleiben, daß der „wesentliche" Teil des Wavelets immer noch im abgetasteten Bereich liegt, denn sonst hängt der wirkliche Wert der Wavelettransformation (3.21) signifikant von Werten ausserhalb des Abtastbereichs ab. Es sollte also wenigstens eine Heisenberg–Box zu s_{M-1} ganz im Abtastbereich liegen.

3. Der Rest ist ein Zusammenspiel zwischen σ und M. Hält man die Progression σ fest, so ist M durch die Skaleneinschänkungen nach oben beschränkt, hält man umgekehrt die Skalenzahl M fest, so muss man halt σ entsprechend fein wählen.

[128] Um's mal wieder gruppentheoretisch zu sagen: Eigentlich liegen die Skalen ja in $\mathbb{R}_+$ und da ist der Mittelpunkt das neutrale Element 1.

[129] Das nach neuer „Rechtschreibung" zumindest zulässige, wenn nicht angeratene *Heisenberg Boxen* liest sich eher wie eine Kampfsportart.

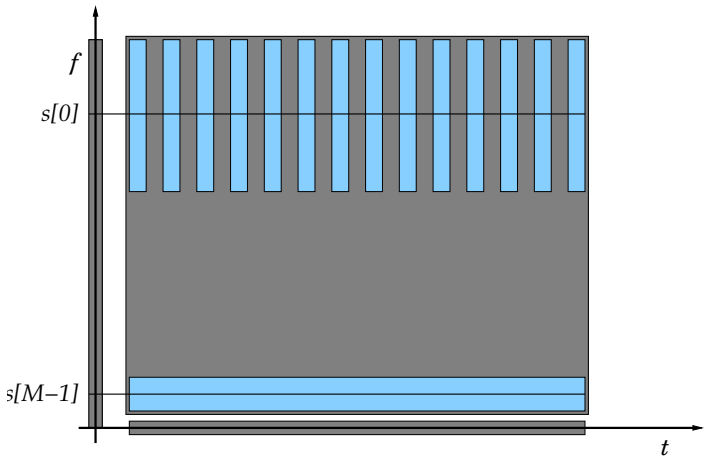


Abbildung 3.11: Die Einschränkungen an die Skalen in schematischer Darstellung. Die „unterste“ und breiteste Heisenberg–Box sollte noch komplett im Inneren des Abtastbereichs liegen, die „obersten“ und schmalsten aber höchsten Boxen hingegen im zulässigen Frequenzbereich.

Die Bedeutung dieser Faustregeln ist in Abb 3.11 dargestellt. Durch die Abtastung wird ein Zeit-/Frequenzfenster vorgegeben, in dem diese sinnvoll ist. Das Zeitfenster wird dabei durch den Abtastbereich $[t_0, t_{N-1}]$ bestimmt, das Frequenzfenster ergibt sich über den Shannonschen Abtastsatz aus der Schrittweite h. Und die Wahl der Skalen sollte nun so erfolgen, daß alle Heisenberg–Boxen innerhalb dieses Zeit-/Frequenzfensters liegen, denn alle Parameterwerte (s, u), die zu Wavelets führen, deren Heisenberg–Boxen das Fenster verlassen, verwenden Information, die in der Abtastung nicht entalten ist.

3.3.1 Die Implementierung der Wavelettransformation

Nach so viel Theorie gehen wir jetzt in eine ganz andere Richtung und sehen uns einmal genauer an, wie eine Wavelettransformation eigentlich numerisch berechnet wird, wie man also zu den schönen Bildern kommt. Dazu bemerken wir zuerst einmal, daß die zu transformierende Funktion eigentlich nie als eine solche vorliegt, sondern nur in Form einer **Abtastung**

$$f(t_j), \qquad t_0 < \cdots < t_N, \qquad N \in \mathbb{N},$$

und daß diese Abtastung aus begreiflichen praktischen Gründen **endlich** ist. In vielen Fällen

1. sind, meist aufgrund technischer Gegebenheiten, diese Stellen t_j **äquidistant**, das heisst, $t_j = t_0 + jh$, $h > 0$,
2. steht die Zahl N der Abstastungen a priori gar nicht genau fest,
3. kann man die Abtastpunkte weder vorschreiben noch in irgendeiner Form beeinflussen.

Ein naiver Ansatz zur Berechnung der Wavelettransformation könnte nun einfach diese Punkte t_j verwenden und das Integral durch eine Summe

$$W_\psi(s,u) \sim \sum_{j=0}^{N} w_j\, f(t_j)\; \frac{1}{\sqrt{s}}\overline{\psi\Big(\frac{t_j-u}{s}\Big)}, \qquad w_j > 0, \tag{3.28}$$

als eine **Quadraturformel**, siehe (Isaacson & Keller, 1966; Gautschi, 1997; Sauer, 2000a) ersetzen. Im Falle der **Rechtecksregel** ist beispielsweise $w_j = t_{j+1} - t_j$, $j = 0, \ldots, N-1$, und w_N passend eine vernünftige Wahl, oder bei äquidistanten Knoten gerne auch $w_j = h$, $j = 0, \ldots, N$. Für festes s haben wir dann für jedes u einen Rechenaufwand von $O(N)$ und wenn wir, was intuitiv naheliegend ist, die Wavelettransformation für die Stellen $u_j = t_j$, $j = 0, \ldots, N$, auf diese Art und Weise bestimmen, dann kostet uns das $O(N^2)$ Rechenoperationen.

Ausserdem müssen wir dafür das Wavelet an den Stellen $\frac{t_j - t_k}{s}$, $j, k = 0, \ldots, N$, auswerten. Das ist einerseits nicht so schlimm, denn die Wavelets sind in vielen Fällen *explizit* bekannt, siehe Beispiel 3.16, macht es aber zumindest empfehlenswert, das Wavelet wegen der Lage dieser Auswertungspunkte um den Ursprung zu zentrieren[130]. Etwas mehr Vorsicht benötigt schon die Tatsache, daß diese fixe Auswertung bereits den Skalenbereich einschränkt! Wavelets fallen normalerweise sehr schnell ab, und das in Zeit und Frequenz:

$$\lim_{x\to\pm\infty} \psi(x) = 0, \qquad \lim_{\xi\to\pm\infty} \psi(\xi) = 0,$$

wobei die Abklingrate gerne auch mal exponentiell ist, beispielsweise bei den Wavelets aus Beispiel 3.16[131]. Ist bei so einem Wavelet nun aber s klein genug, dann hat die Quadraturformel (3.28) die Form

$$W_\psi(s, t_j) \sim w_j\, f(t_j)\; \frac{1}{\sqrt{s}}\overline{\psi(0)}, \tag{3.29}$$

und das ist nicht unbedingt das, was man haben möchte.

Die numerische Realisierung der Wavelettransformation ist dennoch eigentlich sehr einfach, wir müssen nämlich nur die Faltungsstruktur ausnutzen und aus daraus ein Produkt der Fouriertransformierten machen. Dazu setzen wir (2.40) in (3.24) ein und erhalten

$$(W_\psi f(s,\cdot))^\wedge\Big(\frac{2\pi k}{N}\Big) \sim (W_\psi f_\varphi(s,\cdot))^\wedge\Big(\frac{2\pi k}{N}\Big) = \sqrt{s}\, h\, \widehat{f_h}(k)\, \widehat{\varphi}\Big(\frac{2\pi k}{h N}\Big)\widehat{\psi}\Big(\frac{2\pi k s}{N}\Big),$$

bzw.

$$\mathrm{DFT}\,(\sigma_h W_\psi f(s,\cdot)) \sim \sqrt{s}\, h\, \widehat{f_h}(k)\, \widehat{\varphi}\Big(\frac{2\pi k}{h N}\Big)\widehat{\psi}\Big(\frac{2\pi k s}{N}\Big), \tag{3.30}$$

und durch eine **inverse DFT** bzw. **inverse FFT** erhalten wir aus (3.30) die Berechnungsvorschrift

$$[W_\psi(s,t_j) \;:\; j \in \mathbb{Z}_N] \leftarrow \sqrt{s} h\ \mathrm{IDFT}\Big[\widehat{f_h}(k)\, \widehat{\varphi}\Big(\frac{2\pi k}{h N}\Big)\widehat{\psi}\Big(\frac{2\pi k s}{N}\Big) \;:\; k \in \mathbb{Z}_N\Big], \tag{3.31}$$

[130] Das ist natürlich überhaupt kein Problem und damit etwas, das man bestenfalls falsch machen kann.

[131] Eigentlich haben alle „praxisrelevanten“ Wavelets diese Eigenschaft, da sie unerlässlich für eine gute Zeit-/Frequenz–Lokalisierung ist, siehe z.B. (Sauer, 2008).

die man als **schnelle Wavelettransformation** bezeichnen könnte, auch wenn der Begriff später im Filterbank–Kontext noch für etwas anderes benutzt werden wird. Diese Berechnungsmethode hat zwei direkte Konsequenzen:

1. Die Berechnung der Wavelettransformation kostet nur noch $O(N \log N)$ pro Skala, wobei die $\widehat{\phi}$–Werte vorberechnet und unabhängig von s verwendet werden können[132]

2. Die skalenweisen Multiplikationen sind voneinander unabhänging und können damit wunderbar parallelisiert werden, was eine Realisierung auf einer GPU[133] nahelegt.

3. Die Fouriertransformierte $\widehat{\psi}$ des Wavelets müssen wir hingegen für jedes s unterschiedlich abtasten, aber diese ist bei vielen praxisrelevanten Wavelets ja sogar explizit bekannt. Das hat sogar die Konsequenz, daß man Wavelets eigentlich eher im Fourier- denn im „Zeit"–Bereich entwirft.

Bei der Wahl der „Frequenzen", genauer der Skalen, an denen die Wavelettransformation berechnet werden soll, sind wir hingegen noch völlig frei und könnten da nun prinzipiell jede aufsteigende Folge s_j, $j \in \mathbb{Z}_M$, verwenden. Dennoch gibt es aber eine natürliche Skalenwahl, die man mit ein klein wenig Theorie auch gut begründen kann.

3.3.2 Inverse Transformation & Pferdefüße

Wenn wir nun schon die Wavelettransformation schnell realisieren können, wie sieht es dann mit der inversen Wavelettransformation aus? Dazu werfen wir nochmals einen Blick auf die Invertierungsformel (3.23),

$$f = \frac{1}{C_\psi} \int_{\mathbb{R}} \underbrace{\int_{\mathbb{R}} W_\psi f(s,u)\, \frac{1}{\sqrt{|s|}} \psi\left(\frac{\cdot - u}{s}\right) du}_{= W_\psi f(s,\cdot) * \psi_s = \left(\widehat{W_\psi}(s,\cdot)\, \widehat{\psi}_s\right)^\vee} \frac{ds}{s^2}$$

und berechnen, einem numerischen Grundprinzip folgend[134], diese wieder über eine FFT. Es bleibt also nur noch das Integral $\int \frac{ds}{s^2}$, bei dem wir um eine Quadratur nicht herumkommen. Die **Knoten** dieser Quadraturformel sind nun die Skalen s_j an denen wir eine Wavelettransformation berechnet haben – letztendlich gehen wir bei der inversen Transformation ja immer davon aus, daß wir nur die diskreten Werte

$$W_\psi f(s_j, t_k), \qquad j \in \mathbb{Z}_M,\ k \in \mathbb{Z}_N, \tag{3.32}$$

[132] Und selbst $\widehat{\varphi} \equiv 1$ ist möglich, wenn auch nicht besonders empfehlenswert, denn das wäre eine Faltung mit der eher langsam abfallenden sinc–Funktion.

[133] **G**raphics **P**rocessing **U**nit, also eigentlich Grafikkarte. Da die guten und teueren Stücke sich aber heute immer dann langweilen, wenn sie keine 3D–Grafiken berechnen müssen, kann man sie als Parallelrechner beschäftigen.

[134] Dies mag manchen Leuten banal vorkommen, aber die einfache Idee, Faltungsstrukturen durch die FFT schnell zu berechnen, spielt in vielen Anwendungen bis hin zur Berechnung des Produkts großer ganzer Zahlen eine fundamentale Rolle, siehe (Knuth, 1998; Sauer, 2001).

kennen. Diese Invertierung, die erstmals in (Domes, 2007) untersucht wurde und in (Sauer, 2011) zu finden ist, bestimmt nun wieder zuerst eine Matrix[135] im Sinne von (3.30), und zwar

$$F := \sqrt{s}h \left[\mathrm{IDFT}\left((W_\psi f(s_j,\cdot))^\wedge_h \, \widehat{\varphi}\left(\frac{2\pi\cdot}{h\,N}\right)\widehat{\psi}\left(\frac{2\pi\cdot s_j}{h\,N}\right)\right)(k) \;:\; \begin{array}{l} j \in \mathbb{Z}_M \\ k \in \mathbb{Z}_N \end{array} \right].$$

Das sind die Werte, die nun bezüglich s integriert bzw. bezüglich der s_j summiert werden müssen, um letztendlich den Vektor der Funktionswerte zu liefern, was wir als

$$f(t_k) = \sum_{j\in\mathbb{Z}_M} w_j F_{jk} = w^T F$$

schreiben können und wofür wir natürlich noch die **Gewichte** w_j der Quadraturformel bestimmen müssen. Der einfachste Fall wäre hier eine (zusammengesetzte) **Rechtecksregel**, bei der wir

$$w_j = \int_{s_j}^{s_{j+1}} \frac{ds}{s^2}, \qquad j \in \mathbb{Z}_M,$$

setzen und ein irgendwie passendes s_M festlegen müssen und die konstante Funktionen exakt reproduziert. Jetzt erweist sich die natürliche Wahl (3.27) der Skalen auch wieder als hilfreich, denn damit ergibt sich

$$w_j = \int_{s_0\sigma^j}^{s_0\sigma^{j+1}} \frac{ds}{s^2} = -\frac{1}{s}\Big|_{s_j}^{s_j\sigma} = \frac{1}{s_j}\left(1-\sigma^{-1}\right),$$

also

$$w = \frac{\sigma-1}{\sigma}\left[s_j^{-1} \;:\; j \in \mathbb{Z}_M\right]. \tag{3.33}$$

Entsprechend liessen sich natürlich auch andere Quadraturformeln höherer Ordnung adaptieren.

Übung 3.5 Bestimmen Sie den Gewichtsvektor w für das Analogon der zusammengesetzten Trapezregel. ◊

Eigentlich haben wir es uns aber viel zu schwer gemacht, denn wenn wir uns (3.24) noch einmal ansehen, so können wir diese Formel in

$$\widehat{f}(\xi) = \frac{(W_\psi f(s,\cdot))^\wedge(\xi)}{\sqrt{s}\widehat{\psi}(s\xi)}, \qquad \xi \in \mathbb{R}\setminus\{0\}, \quad s \in \mathbb{R}_+,$$

umformen und demzufolge $\widehat{f}$ und damit f aus **einer einzigen Skala** $s > 0$ der Wavelettransformierten rekonstruieren, solange nur $\widehat{\psi}(\xi) \neq 0$ für alle $\xi \neq 0$ erfüllt ist. Dieselbe Idee würde ausgehend von (3.30) die „einfache" Invertierungsformel

$$\widehat{f}_h = \left[\frac{(W_\psi f(s,\cdot))^\wedge\left(\frac{2\pi k}{N}\right)}{\sqrt{s}\,h\,\widehat{\varphi}\left(\frac{2\pi k}{hN}\right)\widehat{\psi}\left(\frac{2\pi k s}{hN}\right)} \;:\; k \in \mathbb{Z}_N \right]$$

[135] Oder eine Vektor von Vektoren.

liefern, die allerdings alles andere als praktisch oder praktikabel ist. Man kann nämlich recht leicht nachrechnen (Sauer, 2011), daß die Struktur der Invertierungsformel (3.23) dafür sorgt, daß lokale Effekte auf der Waveletseite auch lokal auf der Signalseite sind. Allerdings zeigt sich eines ganz deutlich:

> Die Wavelettransformation ist hochgradig redundant, eigentlich ist fast alle Information über f bereits in einer Skala von $W_\psi f$ enthalten.

Die inverse Wavelettransformation (3.23) kann ja im Prinzip auf *jede* Funktion in den zwei Variablen s und u angewendet werden und liefert dann als Ergebnis eine Funktion in einer Variablen. Würde man allerdings nun wieder die Wavelettransformation dieser Funktion bestimmen, so ergibt sich nicht wieder die originale bivariate Funktion. Mit anderen Worten,

$$W_\psi^{-1} W_\psi = I, \qquad W_\psi W_\psi^{-1} \neq I. \tag{3.34}$$

Das sollte man sich dann doch ein wenig genauer ansehen. Dazu bemerken wir zuerst, daß die Wavelettransformation $W_\psi f$ als bivariate Funktion in s, u die **Kompatibilitätsbedingung**

$$\frac{(W_\psi f(s,\cdot))^\wedge(\xi)}{(W_\psi f(s',\cdot))^\wedge(\xi)} = \sqrt{\frac{s}{s'}}\frac{\widehat{\psi}(s\xi)}{\widehat{\psi}(s'\xi)}, \qquad s, s' \in \mathbb{R}_+, \quad \xi \in \mathbb{R}, \tag{3.35}$$

erfüllt.

Definition 3.23 *1. Eine bivariate Funktion* $g : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{C}$ *ψ–**kompatibel**, wenn sie (3.35) erfüllt.*

2. Zwei Funktionen $g, g' : \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ *heißen verwandt, wenn* $W_\psi^{-1} g = W_\psi^{-1} g'$ *gilt.*

Die Wavelet–Transformation ist natürlich eine ψ–kompatible Funktion, so ist die Eigenschaft ja gerade gebaut. Andererseits sind die Wavelettransformationen aber auch die einzigen kompatiblen Funktionen und kommen ausserdem in den besten Familien vor.

Lemma 3.24 *Zu jeder Funktion* $g : \mathbb{R}\times\mathbb{R} \to \mathbb{C}$ *gibt es eine ψ–kompatible Verwandte g', die außerdem ihrerseits eine Wavelettransformierte ist.*

Beweis: Wir setzen einfach $g' = W_\psi W_\psi^{-1} g$, dann ist

$$W_\psi^{-1}(g - g') = W_\psi^{-1} g - \underbrace{W_\psi^{-1} g W_\psi}_{=I} W_\psi^{-1} g = W_\psi^{-1} g - W_\psi^{-1} g = 0.$$

Damit sind g und g' verwandt und g' ist offensichtlich eine Wavelettransformierte. □

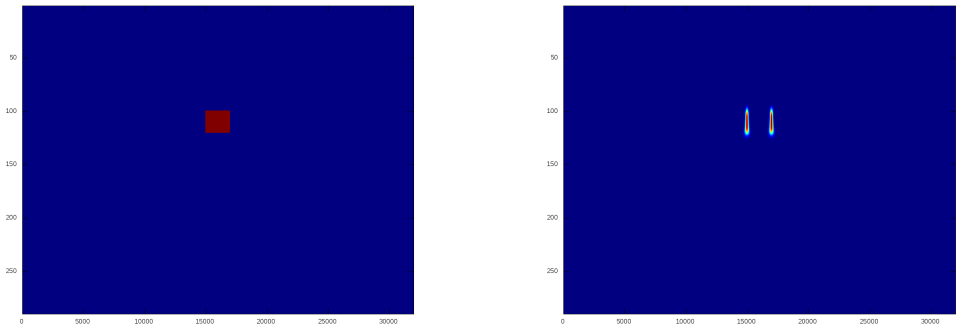


Abbildung 3.12: Ein zweidimensionales Signal *(links)* und dessen verwandte Wavelettransformation *(rechts)*. Schon ein kleiner Unterschied.

Das erklärt dann auch schon die Mehrdeutigkeit der inversen Wavelettransformation: Wenn wir Äquivalenklassen modulo W_ψ^{-1} in $L_2(\mathbb{R}\times\mathbb{R})$ bilden und damit verwandte Funktionen identifizieren, dann enthält nach Lemma 3.24 jede solche Äquivalenzklasse genau[136] eine Wavelettransformation und $W_\psi W_\psi^{-1}$ ist der Projektor auf diese Repräsentanten. Und dieser Repräsentant ist dann natürlich auch das einzige Element der Äquivalenzklasse, das von $W_\psi W_\psi^{-1}$ reproduziert wird.

3.3.3 Beispiele – Musik und Kanten

Musik- oder Audioanalyse im Allgemeinen befasst sich mit Tönen bzw. mit Tonfolgen. Lokal definieren wir Töne folgendermaßen.

Definition 3.25 *Ein **Ton**[137] ist ein sich periodisch wiederholendes Ereignis, also eine Funktion mit der Eigenschaft* $f(\cdot+\omega^{-1}) = f$. *Die Größe* ω *bezeichnet man als **Frequenz** des Tons, sie wird normalerweise in **Hertz** (*$1Hz = 1s^{-1}$*) angegeben, also der Anzahl der Schwingungen pro Sekunde.*

Für periodische Funktionen kennen wir aus der Analysis eine Darstellungsmethode, nämlich die **Fourierreihe**. Setzen wir die Periodenlänge $1/\omega$ auf 2π, so erhalten wir die folgende Aussage.

Proposition 3.26 *Jede 2π–periodische Funktion f kann eindeutig durch ihre Fourierreihe*

$$\frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\cdot) + \sum_{k=1}^{\infty} b_k \sin(k\cdot) \tag{3.36}$$

[136] Wären $W_\psi f$ und $W_\psi f'$ in derselben Äquivalenzklasse, dann wäre $0 = W_\psi^{-1}(W_\psi f - W_\psi f') = f - f'$, was die Eindeutigkeit so einfach macht, daß man sie wirklich in einer Fußnote verstecken muss.

[137] Es sei darauf hingewiesen, daß beispielsweise Perkussionsinstrumente keinen Ton erzeugen, sondern lediglich ein Geräusch!

dargstellt werden, wobei

$$a_k = \frac{1}{\pi}\int_0^{2\pi} f(t)\,\cos kt\,dt, \qquad b_k = \frac{1}{\pi}\int_0^{2\pi} f(t)\,\sin kt\,dt \tag{3.37}$$

ist.

Bemerkung 3.27 *Proposition 3.26 ist mit ein wenig Vorsicht zu geniessen!*

1. *Zwar sind alle Koeffizienten der Fourierreihe durch (3.37) für „brave" Funktionen wohldefiniert, endlich und eineindeutig*[138]*, aber über die Konvergenz der Fourierreihe (3.36) ist hier* ***nichts*** *gesagt, ebensowenig, ob und wo dieser Grenzwert wirklich mit* f *übereinstimmt.*

2. *Generell ist die Antwort auch tatsächlich recht enttäuschend! Wie DuBois–Reymond 1873 zeigte, gibt es sogar* stetige *Funktionen, deren Fourierreihe an mindestens einer Stelle divergiert, siehe (Sauer, 2002a).*

3. *Andererseits ist es aber auch nicht so schlimm: Normalerweise konvergiert die Fourierreihe braver Funktionen fast überall, siehe (Hardy & Rogosinsky, 1956), aber das wird dann mathematisch schon ein wenig aufwendiger.*

Natürlich haben es alle aufmerksamen Leser[139] schon gemerkt, daß eigentlich Definition 3.25 einen Ton auf ziemlich realitätsferne Art und Weise einführt, denn als periodisches Signal müsste ein Ton ja wieder über alle Zeiten konstant klingen und das passiert ja doch eher selten in der Realität, wo ein Ton natürlich nur eine gewisse endliche Dauer hat. Daher ein paar Annahmen an einen realistischen Ton:

1. Die Dauer des Tons ist wesentlich größer als die Zeit, die für eine Schwingung benötigt wird[140]. Damit ist das Signal zumindest über eine gewisse Zeit periodisch.

2. Die Lautstärke des Tons bleibt über die Dauer seines Erklingens konstant.

Wenn diese beiden Voraussetzungen[141] erfüllt sind, dann können wir zumindest lokal annehmen, wir hätten es mit einer periodischen Funktion zu tun und dann können wir die Fourierreihe (3.36) auch akustisch interpretieren:

> *Jeder Ton lässt sich in* ***Partialtöne*** *zerlegen, deren Frequenzen ganzzahlige Vielfache des Tons selbst sind. Die Fourierkoeffizienten zu diesen Partialtönen werden als* ***Spektrum*** *des Tons bezeichnet und beschreiben die* ***Klangfarbe*** *des Tons.*

[138] Das bedeutet, daß zwei unterschiedliche Funktionen auch unterschiedliche Fourierreihen haben.

[139] Und gibt es überhaupt andere?

[140] Das bedeutet insbesondere, daß tiefe Töne länger gespielt werden müssen als hohe, weswegen es unmöglich ist, einen Jig auf dem Bassregister einer Orgel zu spielen.

[141] Die nun auch noch viele Saiteninstrumente, insbesondere Klavier und Konsorten, ausschließen.

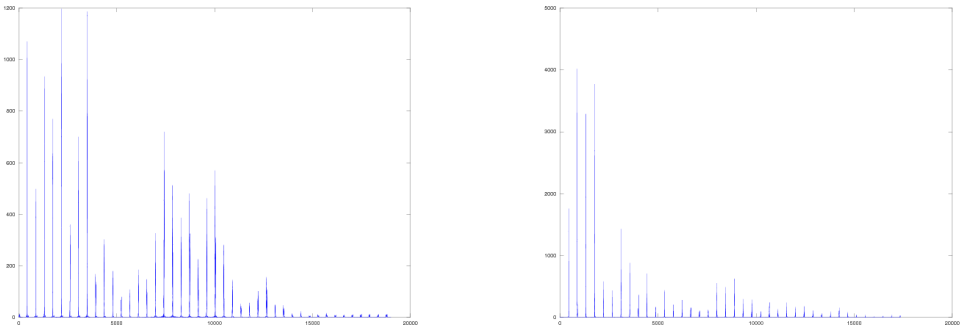


Abbildung 3.13: Spektren zweier Musikinstrumente für denselben Ton (a 440Hz). Der höhere Anteil hochfrequenter Partialtöne links zeigt, daß dieses Instrument sich durch einen „schärferen“ Klang auszeichnet.

Dieser Klangfarben–Effekt ist in Abb. 3.13 zu sehen. Über das Spektrum kann man tatsächlich Klänge sichtbar machen – deswegen werden derartige Methoden auch zur Stimm- und Spracherkennung genutzt.

Das alles funktioniert aber nur für *Dauertöne*, denn in dem Augenblick, wo zwischen zwei Frequenzen (also zwei Tönen) gewechselt wird, ändert sich ja die Länge der Periodisierung. Man braucht dann das Konzept einer **Momentanfrequenz**, siehe (Mallat, 1999), bzw. der **Zeit-/Frequenz–Analyse**, wobei man entweder die Wavelettransformation oder aber die Gabortransformation verwenden kann. Bei der Wavelettransformation sollte man allerdings darauf achten, daß man ein „musikalisches“ Wavelet verwendet, also eines, das einem modulierten Ton entspricht – hier ist wieder einmal Morlet eine gute Wahl.

Den Unterschied zwischen der reinen Frequenzanalyse und der Zeit-/Frequenz–Analyse kann man sehr schöne an einem weiteren akustischen Phänomen aufzeigen, den sogenannten **Schwebungen**. Schwebungen sind die akustische Realisierung des Additionstheorems

$$\cos\omega + \cos\omega' = 2\,\cos\frac{\omega+\omega'}{2}\,\cos\frac{\omega-\omega'}{2}, \tag{3.38}$$

wobei man nun aber die beiden Seiten von (3.38) akustisch/musikalisch interpretiert. Die linke Seite sind zwei gleichzeitig gespielte Töne mit den Frequenzen ω und ω', auf der rechten Seite hat man einen Ton der mittleren Frequenz $(\omega+\omega')/2$, der aber mit der Differenzfrequenz $(\omega-\omega')/2$ amplitudenmoduliert wird. Und je nach verwendeter Transformation bekommt man, siehe Abb. 3.14, nun auch die linke oder die rechte Seite von (3.38) geliefert. Nett ist das Ganze auch bei der Gabortransformation, denn hier kann man dann auch den Einfluss der Fenstergröße sehen wie in Abb. 3.14. Ist das Fenster sehr klein, so schlägt die linke Seite von (3.38) zu, ist bei wachsendem Fenster verschmiert die Frequenz mehr und mehr, und die Schwebung benimmt sich wie die rechte Seite von (3.38). Auch ist zu beachten, daß sich zwischen dem mittleren und rechten Bild

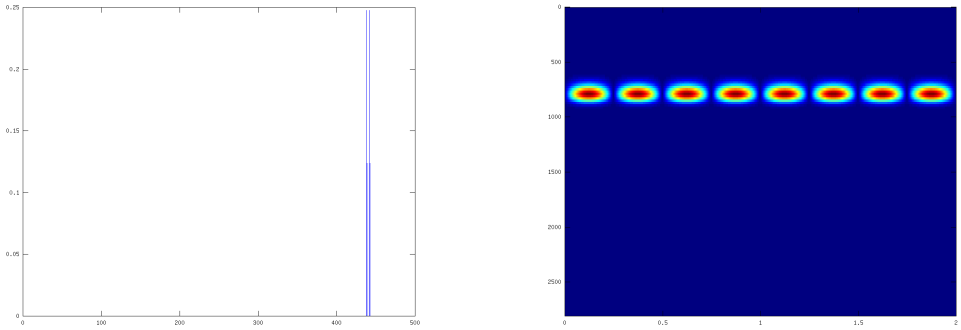


Abbildung 3.14: Spektrum und Wavelettransformierte des Ausdrucks aus (3.38). Das Spektrum reflektiert die linke Seite, die Wavelettransformierte zeigt einen amplitudenmodulierten „verschmierten“ Ton.

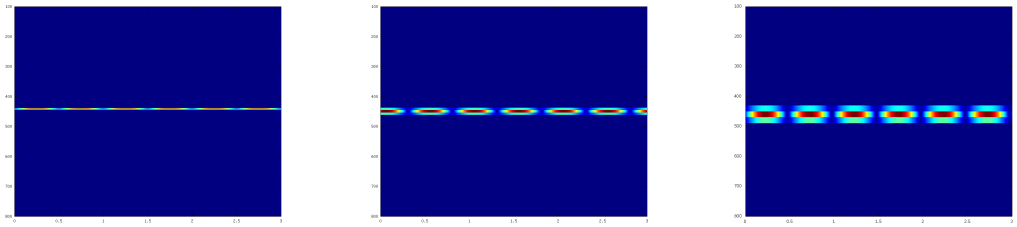


Abbildung 3.15: Drei Gabortransformation der Schwebung mit verschiedenen Fenstergrößen. Links ein sehr goßes Fenster, also fast eine Fouriertransformation, rechts das kleinste Fenster.

in Abb. 3.14 die Phase der Schwebung verschiebt, die Frequenz aber konstant bleibt.

Ein weiteres bisschen Zeit-/Frequenz kann an der Betrachtung einer einfachen Melodie, genauer, eines einfachen „Dreiklangs“ erkennen, der in Abb 3.16 zu sehen ist. Die drei individuellen Töne erscheinen eigentlich alle recht sauber abgegrenzt[142], aber im Dreiklang „verschmieren“ der höchste und er mittlere Ton. Dies ist ein weitere unvermeidbarer Effekt der Wavelettransformation:

> Je höher die Frequenz ist, desto höher wird die Zeitauflösung der Wavelettransformation um den Preis einer schlechteren Frequenzauflösung.

Wavelets versuchen immer, die Frequenz mit einer *relativen* Genauigkeit anzunähern, im Gegensatz dazu versucht es die Gabortransformation mit einer *absoluten*

[142] Daß die „Streifen“ immer eine gewisse Ausdehnung haben, ist eine Konsequenz der Heisenbergschen Unschärferelation – wer Zeit- **und** Frequenzauflösung will, muss ein gewisses Maß an Ungenauigkeit in Kauf nehmen.

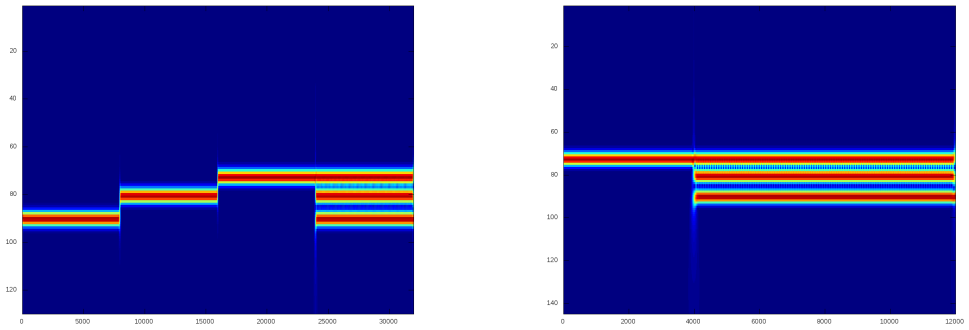


Abbildung 3.16: Eine ganz einfache Melodie, ein aufgelöster und dann unisono gespielter Dreiklang, alles in der Wavelettransformation. Rechts ist nur ein Zoom in die Stelle, an der der Dreiklang beginnt.

Genauigkeit, wobei das genau die Konzepte von absolutem und relativem Fehler sind, die man auch aus der Numerik kennt, (Sauer, 2000a).

Zum „krönenden“ Abschluss noch ein kleier Ausschnitt aus einem Stück echter Musik mit einem Borduninstrument. Den Bordun, also den konstanten Dauerton, erkennt man sehr schön den waagerechten blauen Streifen. Auch die Melodiefolge und die Obertöne sind gut zu sehen, ebenso die Tatsache, daß manche der Partialtöne des Melodieinstruments sich mit Partialtönen der Bordune treffen und so hervorgehoben werden, siehe Abb. 3.18. Dieser Effekt tritt nur dann auf, wenn bei dem Instrument eine **reine Stimmung** verwendet wird, bei der die Frequenzen rationale Vielfache des Grundtons sind, was, im Gegensatz zur heute normalerweise verwendet gleichschwebend oder gleichstufig temperierten Stimmung zu leicht unterschiedlich großen Halbtonintervallen führt. Der Nachteil der reinen Stimmung besteht darin, daß sie nicht transponieren kann, das Instrument also auf eine oder zwei Tonarten festgelegt ist.

Nach diesem kleinen Ausflug in die Welt der Musik wollen wir uns noch eine andere Anwendung ansehen, bei der Wavelets einen echten Mehrwert gegenüber der Gabortransformation liefern. Wegen des $(i\xi)^k$–Terms, der bei der Fouriertransformierten einer Ableitung auftaucht, fällt die Fouriertransformierte

$$\widehat{g}(\xi) = (i\xi)^{-k}\widehat{f}(\xi), \qquad f = g^{(k)},$$

einer **Stammfunktion** g für $\xi \to \pm\infty$ ab, und das umso schneller, je glatter die Funktion g ist, wobei Glattheit im Sinne von Differenzierbarkeit zu verstehen ist. Allerdings sind diese Glattheitssachen natürlich immer globaler Natur und die Abfallrate wird sozusagen durch die „unglatteste“ Stelle von f bestimmt.

Wavelets erlauben hier ein ungleich detaillierteres Vorgehen bei der Analyse wie das folgenden Resultat von Jaffard zeigt, siehe (Mallat, 1999). Die Formulierung ist wortwörtlich aus (Sauer, 2008) übernommen.

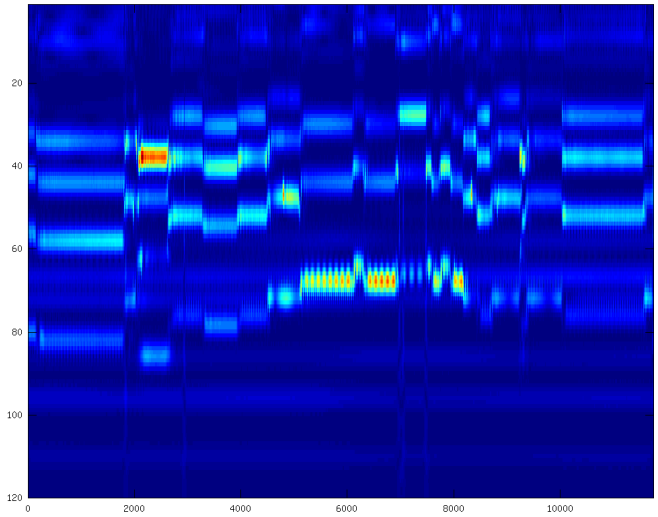


Abbildung 3.17: Wavelettransformation eines Stücks „echter“ Musik mit einem Borduninstrument.

Satz 3.28 *Sei ψ ein Wavelet mit n verschwindenden Momenten und schnell abklingenden Ableitungen der Ordnung $\leq n$.*

1. *Ist $f \in L_2(\mathbb{R})$ Lipschitz–stetig von der Ordnung $\alpha < n$ an $x \in \mathbb{R}$, dann existiert eine Konstante C, so daß*

$$|W_\psi f(u,s)| \leq C\, s^{\alpha+\frac{1}{2}} \left(1 + \left|\frac{u-x}{s}\right|^\alpha\right), \qquad (u,s) \in \Gamma = \mathbb{R} \times \mathbb{R}_+. \tag{3.39}$$

2. *Ist umgekehrt $\alpha \notin \mathbb{N}$ und existieren C sowie $\alpha' < \alpha$, so daß*

$$|W_\psi f(u,s)| \leq C\, s^{\alpha+\frac{1}{2}} \left(1 + \left|\frac{u-x}{s}\right|^{\alpha'}\right), \qquad (u,s) \in \Gamma, \tag{3.40}$$

dann ist f Lipschitz–stetig von der Ordnung α an x.

Anschaulich bedeutet Satz 3.28, daß man die *lokale* Regularität, und Lipschitz–Stetigkeit ist eine Art reelle Erweiterung der Differenzierbarkeit, durch die Abfallrate der Waveletkoeffizienten an dieser Stelle entlang der Skalen im wesentlichen sogar charakterisieren kann, wie man in Abb 3.19 auch sehr gut sehen kann. Das schränkt zwar das Wavelet ein wenig ein, das dann n verschwindende Momente haben muss, das heisst,

$$\int_{\mathbb{R}} t^k\, \psi(t)\, dt = 0, \qquad k = 0, \ldots, n-1,$$

aber solche Wavelets gibt es durchaus.

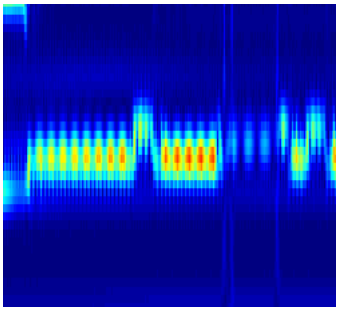


Abbildung 3.18: Ausschnitt aus Abb. 3.17, der zeigt, wie die reine Dur–Sexte sich mit den Partialtönen des Grundtons trifft und so massiv verstärkt. wird.

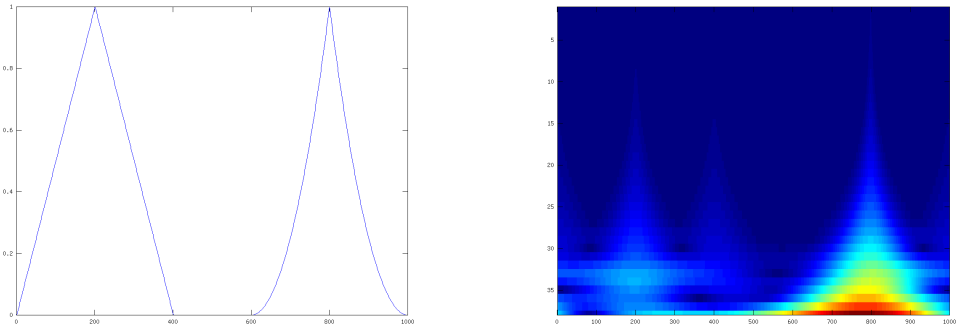


Abbildung 3.19: Eine einfache Testfunktion und ihre Wavelettransformierte

Übung 3.6 Wieviele verschwindende Momente hat das Haar–Wavelet? ◊

Da ein Bild ja ohnehin mehr als die berühmten 1000 Worte sagt, sehen wir uns einfach Abb. 3.19 an, aus dem deutlich erkennbar ist, daß die Wavelettransformierte die drei Ecken durch „Pfeile“ markiert und daß die Deutlichkeit dieser „Pfeile“ wohl offensichtlich etwas mit dem Öffnungswinkel der Ecken zu tun. Denn je spitzer so eine Ecke ist, desto geringer ist ja die Lischitz–Stetigkeit und desto „sichtbarer“ ist der Pfeil, da die Wavelettransformierte entsprechend langsam abfällt.

3.4 Filterbänke

Es gibt ein zweites, volldiskretes Konzept der Wavelets, mit dem wir uns jetzt ein wenig beschäftigen wollen. Dieses Konzept basiert wieder auf der Idee der *digitalen Filter*, wie wir sie in 2.4 kennengelernt haben. Bei deren Behandlung hilft uns das folgende mathematische Konzept.

Definition 3.29 (z–Transformation) *Die z–**Transformation** eines diskreten Signals* $c \in \ell(\mathbb{Z})$ *ist die* formale ***Laurentreihe***[143]

$$c^*(z) := \sum_{k\in\mathbb{Z}} c(k)\, z^{-k}, \qquad z \in \mathbb{C}_\times = \mathbb{C} \setminus \{0\}. \tag{3.41}$$

Die z–Transformation ist mit der Fouriertransformation eng verwandt, schließlich gilt für $\theta \in \mathbb{T}$

$$c^*\left(e^{i\theta}\right) = \sum_{k\in\mathbb{Z}} c(k)\, e^{-ik\theta} = \widehat{c}(\theta), \tag{3.42}$$

und da die Koeffizienten bei der z–Transformierten wie der Fouriertransformierten *eindeutig* sind, kann man die eine aus der anderen rekonstruieren. Somit ist es nicht verwunderlich, daß sich z–Transformation und Fouriertransformation ähnlich verhalten, insbesondere gilt

$$(c * d)^*(z) = c^*(z)\, d^*(z). \tag{3.43}$$

Übung 3.7 Beweisen Sie (3.43). ◊

Die Idee des **Subband Coding**, das insbesondere bei der Datenkompression gerne verwendet wird, besteht darin, ein Signal in mehrere Teilsignale, sogenannte **Subbänder**, zu zerlegen und jedes dieser Teilsignale separat zu codieren – am besten natürlich so, daß sich aus diesen Subbändern das Ausgangssignal wieder rekonstruieren lässt.

Beispiel 3.30 *Die naheliegendste Idee wäre natürlich, für die Subband–Zerlegung verschiedene Bandpass–Filter*

$$\chi^\vee_{[t_j,t_{j+1}]}, \qquad 0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = 2\pi$$

zu verwenden und so das vollständige Frequenzband mittels exakter Bandpass–Filter aufzuspalten und diese Teilsignale weiterzuverarbeiten[144]*. Dieser Ansatz erfreut zwar mit Sicherheit den Phono–Freak, ist aber ziemlich aufwendig und auch näherungsweise nur mit sehr großen und damit verzögernden Filtern zu realisieren. Daher wollen wir uns eine einfachere Zerlegung ansehen, die sehr leicht zu realisieren ist und uns obendrein zu Wavelets führen wird.*

Definition 3.31 *Sei* $n \geq 2$ *ist. Der Operator* $\downarrow_n$*, der* $c \in \ell(\mathbb{Z})$ *das Signal*

$$\downarrow_n c = c(n\cdot)$$

*zuordnet, heißt **Downsampling–Operator** der Ordnung* n*, der Operator* $\uparrow_n$ *mit*

$$\uparrow_n c(j) = \begin{cases} c(j/n), & j \in n\mathbb{Z}, \\ 0, & j \in \mathbb{Z} \setminus n\mathbb{Z}, \end{cases}$$

*heißt **Upsampling–Operator** der Ordnung* n*.*

[143] Eine Laurentreihe ist eigentlich fast dasselbe wie eine Potenzreihe, nur daß auch negative Exponenten zugelassen sind. Das macht algebraisch durchaus einen Unterschied, aber der ist für uns an dieser Stelle nicht relevant.

[144] Das ist das, was ein "idealer" Equalizer in einer Stereoanlage wohl machen würde.

Offensichtlich gilt $\downarrow_n\uparrow_n = I \neq \uparrow_n\downarrow_n$. Mit Hilfe des Downsampling–Operators können wir nun ein Signal c in die "Teilbänder"

$$c_j = \downarrow_n \tau_j c, \qquad j \in \mathbb{Z}_n,$$

zerlegen, die man über die Formel

$$c = \sum_{j\in\mathbb{Z}_n} \tau_j \uparrow_n c_j$$

wieder zu c kombinieren kann. Der Zerlegungsprozess liefert hierbei

$$\begin{array}{cccccccccc}
\dots & c(-1) & \boxed{c(0)} & c(1) & \dots & c(n-1) & \boxed{c(n)} & c(n+1) & \dots & \to c_0 \\
\dots & c(0) & \boxed{c(1)} & c(2) & \dots & c(n) & \boxed{c(n+1)} & c(n+2) & \dots & \to c_1 \\
 & \vdots & \boxed{\vdots} & \vdots & & \vdots & \boxed{\vdots} & \vdots & & \\
\dots & c(n-2) & \boxed{c(n-1)} & c(n) & \dots & c(2n-2) & \boxed{c(2n-1)} & c(2n) & \dots & \to c_{n-1}
\end{array}$$

so daß

$$c_j = c\,(n\cdot + j)\,, \qquad j \in \mathbb{Z}_n,$$

ist, was sich mit Upsampling und Translation zu

$$\begin{array}{cccccccccc}
c_0 \to & c(0) & 0 & c(1) & \dots & 0 & c(n) & 0 & \dots & \\
c_1 \to & 0 & c(0) & 0 & \dots & c(1) & 0 & c(n+1) & \dots & \\
 & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & & \\
c_{n-1} \to & c(-1) & 0 & c(1) & \dots & c(n-2) & c(n-1) & 0 & \dots & \\
\hline
 & c(-1) & c(0) & c(1) & \dots & c(n-2) & c(n-1) & c(n) & \dots & \to c
\end{array}$$

kombiniert wie behauptet. Beim **Subband Coding** verschieben wir nicht nur einfach, sondern filtern zuerst die Daten mit n Filtern F_j, $j \in \mathbb{Z}_n$, und wenden dann auf jedes der Resultate den Downsampling–Operator an, also

$$c_j = \downarrow_n F_j\, c, \qquad j \in \mathbb{Z}_n. \tag{3.44}$$

Als schematisches Bild sieht das dann wie folgt aus:

$$\begin{array}{ccccccc}
 & & \nearrow & \boxed{F_0} & \to & \boxed{\downarrow_n} & \to\ c_0 \\
c & \to & \odot & \vdots & & \vdots & \vdots \\
 & & \searrow & \boxed{F_{n-1}} & \to & \boxed{\downarrow_n} & \to\ c_{n-1}
\end{array} \tag{3.45}$$

Die Zerlegung aus (3.45) bezeichnet man als **Analyse–Filterbank** $F = [F_j : j \in \mathbb{Z}_n]$. Durch das Downsampling enthält jede Komponente c_j von c nur den n–ten Teil der Information des Signals $F_j c$, also auch nur den n–ten Teil der Information von c, vorausgesetzt, die Filter sind vernünftig.

Wir haben in (3.45) die Anzahl der Filter bzw. Subbänder gleich dem Dezimierungsfaktor n, also dem Faktor der Unterabtastung gewählt. Eine derartige Filterbank nennt man **kritisch abgetastet** bzw. **critically sampled**, ist die Anzahl

der Subbänder größer als der Dezimierungsfaktor, so spricht man von **Überabtastung**[145], ist sie kleiner, von **Unterabtastung**. Am meisten Spass machen aber die kritisch abgetasteten.

Wir kennen ja bereits das einfachste Beispiel einer Filterbank, nämlich den Fall, daß $F_j = \tau_j$, $j \in \mathbb{Z}_n$, ist, dann entspricht die Filterbank tatsächlich der Zerlegung des Signals modulo n.

Als nächstes wollen wir eine weitere Beschreibung der Filterbank geben, wobei wir wieder annehmen, daß jeder Filter F_j ein LTI–Filter mit Impulsantwort f_j ist, daß also $F_j c = f_j * c$ gilt. Das führt zum folgenden Begriff.

Definition 3.32 (Modulationsmatrix) *Die **Modulationsmatrix** $M(z)$ zu der Filterbank*

$$F = [F_j \,:\, j \in \mathbb{Z}_n]$$

ist definiert als

$$\begin{aligned} M(z) &:= \frac{1}{n}\left[f_j^*\left(e^{2\pi i k/n} z\right) \,:\, j,k \in \mathbb{Z}_n\right] \\ &= \frac{1}{n}\begin{bmatrix} f_0^*(z) & f_0^*\left(e^{2\pi i/n} z\right) & \dots & f_0^*\left(e^{2\pi i(n-1)/n} z\right) \\ \vdots & \vdots & \ddots & \vdots \\ f_{n-1}^*(z) & f_{n-1}^*\left(e^{2\pi i/n} z\right) & \dots & f_{n-1}^*\left(e^{2\pi i(n-1)/n} z\right) \end{bmatrix}. \end{aligned} \tag{3.46}$$

Eine besonders einfache Gestalt hat die Modulationsmatrix wieder einmal für $n = 2$, wo man

$$M(z) = \begin{bmatrix} f_0^*(z) & f_0^*(-z) \\ f_1^*(z) & f_1^*(-z) \end{bmatrix}$$

erhält. Generell sind die Vorfaktoren $e^{2\pi i j/n}$ vor der Variablen z ja n–te Einheitswurzeln und spielen somit die Rolle verallgemeinerter Vorzeichen. Die Bedeutung der Modulationsmatrix liegt aber darin, daß sie die Aktion der Filterbank beschreibt.

Satz 3.33 *Für die Filterbank aus (3.45) gilt*

$$\left[c_j^*(z^n) \,:\, j \in \mathbb{Z}_n\right] = M(z)\,\left[c^*\left(e^{2\pi i j/n} z\right) \,:\, j \in \mathbb{Z}_n\right], \tag{3.47}$$

das heißt,

$$\begin{bmatrix} c_0^*(z^n) \\ \vdots \\ c_{n-1}^*(z^n) \end{bmatrix} = M(z) \begin{bmatrix} c^*(z) \\ c^*\left(e^{2\pi i/n} z\right) \\ \vdots \\ c^*\left(e^{2\pi i(n-1)/n} z\right) \end{bmatrix} \tag{3.48}$$

Übung 3.8 Bestimmen Sie die Modulationsmatrix für die Translationsfilter $F_j = \tau_j$, $j \in \mathbb{Z}_n$. ◊

[145] Die Zerlegung enthält dann „mehr Information“ als das Ausgangssignal.

Definition 3.34 (Polyphase–Vektor) *Der Vektor*

$$\left[c^*\left(e^{2\pi ij/n}z\right) \,:\, j \in \mathbb{Z}_n\right] \tag{3.49}$$

zu einem Signal c *heißt* ***Polyphasen–Vektor*** *oder* ***Polyphase Vector*** *zu* c.

Im Fall $n = 2$ wird (3.48) dann zu

$$\begin{bmatrix} c_0^*(z^2) \\ c_1^*(z^2) \end{bmatrix} = \begin{bmatrix} f_0^*(z) & f_0^*(-z) \\ f_1^*(z) & f_1^*(-z) \end{bmatrix} \begin{bmatrix} c^*(z) \\ c^*(-z) \end{bmatrix}.$$

Jetzt aber an den Beweis von Satz 3.33. Dazu benötigen wir zuerst etwas Information, wie sich die Up- und Downsampling–Operatoren auf ein Signal auswirken.

Lemma 3.35 *Für* $n \in \mathbb{N}$ *ist*

$$(\downarrow_n c)^*(z^n) = \frac{1}{n}\sum_{k\in\mathbb{Z}_n} c^*\left(e^{2\pi ik/n}z\right), \qquad (\uparrow_n c)^*(z) = c^*(z^n), \tag{3.50}$$

Beweis: Da

$$\frac{1}{n}\sum_{k\in\mathbb{Z}_n} e^{-2\pi ijk/n} = \begin{cases} 1, & j \in n\mathbb{Z}, \\ 0, & j \notin n\mathbb{Z}, \end{cases}$$

siehe Lemma 2.28, ergibt sich die linke Indentität in (3.50) aus

$$\begin{aligned}(\downarrow_n c)^*(z^n) &= \sum_{j\in\mathbb{Z}} (\downarrow_n c)(j)\, z^{-nj} = \sum_{j\in\mathbb{Z}} c(nj)\, z^{-nj} = \sum_{j\in\mathbb{Z}} c(j)\, z^{-j}\left[\frac{1}{n}\sum_{k\in\mathbb{Z}_n} e^{-2\pi ijk/n}\right] \\ &= \frac{1}{n}\sum_{j\in\mathbb{Z}} c(j)\sum_{k\in\mathbb{Z}_n}\left(e^{2\pi ik/n}z\right)^{-j} = \frac{1}{n}\sum_{k\in\mathbb{Z}_n} c^*\left(e^{2\pi ik/n}z\right),\end{aligned}$$

Die rechte Seite von (3.50) erhält man hingegen aus der einfachen Rechnung

$$(\uparrow_n c)^*(z) = \sum_{j\in\mathbb{Z}} c(j)\, z^{-nj} = c^*(z^n).$$

□

Besonders einfach wird (3.50) natürlich wieder im Fall $n = 2$. Da $e^{i\pi} = -1$ ist, ergibt sich dann nämlich

$$(\downarrow_2 c)^*\left(z^2\right) = \frac{1}{2}\left(c^*(z) + c^*(-z)\right), \qquad (\uparrow_2 c)^*(z) = c^*\left(z^2\right). \tag{3.51}$$

Es gibt aber noch eine andere Interpretation von (3.50). Dazu schreiben wir $c^*(z)$ als

$$\begin{aligned}c^*(z) &= \sum_{j\in\mathbb{Z}} c(j)z^{-j} = \sum_{k\in\mathbb{Z}_n}\sum_{j\in\mathbb{Z}} c(nj+k)\, z^{-nj-k} = \sum_{k\in\mathbb{Z}_n} z^{-k}\sum_{j\in\mathbb{Z}} (\uparrow_n \tau_k c)(j)\, z^{-nj} \\ &= \sum_{k\in\mathbb{Z}_n} z^{-k}\,(\uparrow_n \tau_k c)^*(z^n) =: \sum_{k\in\mathbb{Z}_n} z^{-k}\widetilde{c}_k^*(z^n),\end{aligned}$$

wobei $\widetilde{c}_k$ dasjenige Signal ist, das wir durch die einfachste Filterbank erhalten, die nur Anteile modulo n bestimmt. Setzen wir jetzt $z = e^{2\pi ij/n}z$ ein, dann ist

$$c^*\left(e^{2\pi ij/n}z\right) = \sum_{k\in\mathbb{Z}_n} e^{-2\pi ijk/n} z^{-k}\widetilde{c}_k^*\left(z^n\right)$$

und somit

$$\begin{aligned}\frac{1}{n}\sum_{j\in\mathbb{Z}_n} c^*\left(e^{2\pi ij/n}z\right) &= \frac{1}{n}\sum_{j\in\mathbb{Z}_n}\sum_{k\in\mathbb{Z}_n} e^{-2\pi ijk/n} z^{-k}\widetilde{c}_k^*\left(z^n\right)\\ &= \sum_{k\in\mathbb{Z}_n}\underbrace{\left(\frac{1}{n}\sum_{j\in\mathbb{Z}_n} e^{-2\pi ijk/n}\right)}_{=\delta_{0k}} z^{-k}\widetilde{c}_k^*\left(z^n\right) = \widetilde{c}_0^*\left(z^n\right).\end{aligned}$$

Multipliziert man nun mit $e^{2\pi ik/n}$, dann erhält man, daß

$$z^{-k}\widetilde{c}_k^*\left(z^n\right) = e^{2\pi ik/n}\,\frac{1}{n}\sum_{j\in\mathbb{Z}_n} c^*\left(e^{2\pi ij/n}z\right), \qquad k\in\mathbb{Z}_n. \tag{3.52}$$

Übung 3.9 Wie sieht (3.52) im Fall $n = 2$ aus? ◊

Beweis von Satz 3.33: Da $c_j = F_j c = f_j * c$ gesetzt wurde, erhalten wir für $j\in\mathbb{Z}_n$

$$\begin{aligned}c_j^*\left(z^n\right) &= \left[\downarrow_n (f_j * c)\right]^*\left(z^n\right) = \frac{1}{n}\sum_{k\in\mathbb{Z}_n}\left(f_j * c\right)^*\left(e^{2\pi ik/n}z\right)\\ &= \frac{1}{n}\sum_{k\in\mathbb{Z}_n} f_j^*\left(e^{2\pi ik/n}z\right)\, c^*\left(e^{2\pi ik/n}z\right).\end{aligned}$$

Daraus folgt (3.48) sofort durch Übergang zur Matrix–Vektor–Schreibweise. □

Da es ja keinen Unterschied macht, ob man die Dinge auf ganz $\mathbb{C}$ oder nur auf dem Einheitskreis betrachtet[146], können wir die Modulationsmatrix dank (3.42) auch für die Fouriertransformierte der Signale betrachten. Und in der Tat: Setzen wir $z = e^{i\xi/n}$ in (3.48) ein, dann folgt, daß

$$\begin{aligned}&\begin{bmatrix}\widehat{c_0}(\xi)\\ \vdots\\ \widehat{c_{n-1}}(\xi)\end{bmatrix} = \begin{bmatrix}c_0^*\left(z^n\right)\\ \vdots\\ c_{n-1}^*\left(z^n\right)\end{bmatrix} = M(z)\left[c^*\left(e^{2\pi ij/n}\,z\right)\,:\,j\in\mathbb{Z}_n\right]\\ &= M\left(e^{i\xi/n}\right)\left[c^*\left(e^{2\pi ij/n}\,e^{i\xi/n}\right)\,:\,j\in\mathbb{Z}_n\right] = M\left(e^{i\xi/n}\right)\left[c^*\left(e^{i(\xi+2j\pi)/n}\right)\,:\,j\in\mathbb{Z}_n\right]\\ &= \begin{bmatrix} f_0^*\left(e^{i\xi/n}\right) & f_0^*\left(e^{i(\xi+2\pi)/n}\right) & \dots & f_0^*\left(e^{i(\xi+2(n-1)\pi)/n}\right)\\ \vdots & \vdots & \ddots & \vdots\\ f_{n-1}^*\left(e^{i\xi/n}\right) & f_{n-1}^*\left(e^{i(\xi+2\pi)/n}\right) & \dots & f_{n-1}^*\left(e^{i(\xi+2(n-1)\pi)/n}\right)\end{bmatrix}\begin{bmatrix} c^*\left(e^{i\xi/n}\right)\\ c^*\left(e^{i(\xi+2\pi)/n}\right)\\ \vdots\\ c^*\left(e^{i(\xi+2(n-1)\pi)/n}\right)\end{bmatrix}\end{aligned}$$

[146] Das ist auch wieder die Sache mit der Substitution $z = e^{i\theta}$

$$= \begin{bmatrix} \widehat{f_0}\left(\frac{\xi}{n}\right) & \widehat{f_0}\left(\frac{\xi}{n}+2\pi\frac{1}{n}\right) & \dots & \widehat{f_0}\left(\frac{\xi}{n}+2\pi\frac{n-1}{n}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{f_{n-1}}\left(\frac{\xi}{n}\right) & \widehat{f_{n-1}}\left(\frac{\xi}{n}+2\pi\frac{1}{n}\right) & \dots & \widehat{f_{n-1}}\left(\frac{\xi}{n}+2\pi\frac{n-1}{n}\right) \end{bmatrix} \begin{bmatrix} \widehat{c}\left(\frac{\xi}{n}\right) \\ \widehat{c}\left(\frac{\xi}{n}+2\pi\frac{1}{n}\right) \\ \vdots \\ \widehat{c}\left(\frac{\xi}{n}+2\pi\frac{n-1}{n}\right) \end{bmatrix},$$

also

$$\left[\widehat{c_j}(\xi) \ : \ j \in \mathbb{Z}_n\right] = \left[\widehat{f_j}\left(\frac{\xi+2k\pi}{n}\right) \ : \ j,k \in \mathbb{Z}_n\right] \left[\widehat{c}\left(\frac{\xi+2j\pi}{n}\right) \ : \ j \in \mathbb{Z}_n\right] \tag{3.53}$$

Die Matrix

$$\left[\widehat{f_j}\left(\frac{\xi+2\pi k}{n}\right) \ : \ j,k \in \mathbb{Z}_n\right]$$

bezeichnet man ebenfalls als **Polyphase Matrix** zur Filterbank. Nun sollte natürlich auch wieder zusammenwachsen, was wir gerade eben so mühsam getrennt haben – wir möchten also den Analyseprozess umkehren und die Subband–Daten c_j, $j \in \mathbb{Z}_n$, wieder zu *einem* Signal c zusammensetzen[147]. Dazu gehen wir genau den umgekehrten Weg und führen erst ein Upsampling der Daten durch, filtern diese mit Filtern G_j, $j \in \mathbb{Z}_n$, und summieren schließlich die Ergebnisse auf, also

$$c' = \sum_{j\in\mathbb{Z}_n} G_j \ \uparrow_n c_j. \tag{3.54}$$

Das führt zur **Synthese–Filterbank**

$$\begin{array}{ccccccc} c_0 & \to & \boxed{\uparrow_n} & \to & \boxed{G_0} & \searrow & \\ \vdots & & \vdots & & \vdots & & \oplus \ \to c \\ c_{n-1} & \to & \boxed{\uparrow_n} & \to & \boxed{G_{n-1}} & \nearrow & \end{array} \tag{3.55}$$

was sich im Kalkül der z–Transformation als

$$c^*(z) = \sum_{j\in\mathbb{Z}_n} \left(G_j \ \uparrow_n c_j\right)^*(z) = \sum_{j\in\mathbb{Z}_n} g_j^*(z) \left(\uparrow_n c_j\right)^*(z) = \sum_{j\in\mathbb{Z}_n} g_j^*(z)\, c_j^*\left(z^n\right) \tag{3.56}$$

schreiben lässt. So weit, so gut, so einfach. Um aber wieder zurück zur Modulationsmatrix zu kommen, erinnern wir uns daran, daß die Eingangsdaten für diese aus dem Vektor $\left[c^*\left(e^{2\pi ij/n}z\right) \ : \ j \in \mathbb{Z}_n\right]$ bestanden haben, und den erhalten wir, indem wir z in (3.56) durch $e^{2\pi ij/n}z$, $j \in \mathbb{Z}_n$, ersetzen, berücksichtigen, daß $\left(e^{2\pi ij/n}z\right)^n = z^n$ ist, und schließlich das Ganze wieder in Matrix–Vektor–Form als

$$\begin{bmatrix} c^*(z) \\ \vdots \\ c^*\left(e^{2\pi i(n-1)/n}z\right) \end{bmatrix} = \underbrace{\begin{bmatrix} g_0^*(z) & \dots & g_{n-1}^*(z) \\ \vdots & \ddots & \vdots \\ g_0^*\left(e^{2\pi i(n-1)/n}z\right) & \dots & g_{n-1}^*\left(e^{2\pi i(n-1)/n}z\right) \end{bmatrix}}_{=:\widetilde{M}(z)} \begin{bmatrix} c_0^*\left(z^n\right) \\ \vdots \\ c_{n-1}^*\left(z^n\right) \end{bmatrix} \tag{3.57}$$

[147] Wobei wir nicht unbedingt annehmen müssen, daß die c_j jemals durch eine Analyse–Filterbank aus diesem c enstanden sein sollen.

bzw.

$$\left[c^*\left(e^{2\pi ij/n}z\right) \,:\, j \in \mathbb{Z}_n\right] = \widetilde{M}(z)\left[c_j^*\left(z^n\right) \,:\, j \in \mathbb{Z}_n\right] \tag{3.58}$$

schreiben. Was wir aber letztendlich betrachten wollen, ist ja nicht das Analyse- oder das Synthesesystem für sich, sondern das Gesamtsystem, die **Filterbank**

$$\begin{array}{ccccccccccccc} & & \nearrow & \boxed{F_0} & \to & \boxed{\downarrow_n} & \to & c_0 & \to & \boxed{\uparrow_n} & \to & \boxed{G_0} & \searrow & & \\ c & \to & \odot & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \oplus & \to c' \\ & & \searrow & \boxed{F_{n-1}} & \to & \boxed{\downarrow_n} & \to & c_{n-1} & \to & \boxed{\uparrow_n} & \to & \boxed{G_{n-1}} & \nearrow & & \end{array} \tag{3.59}$$

Eine natürliche "Minimalforderung" für so eine Filterbank ist sicherlich, daß bei dem System das, was man vorne reinsteckt auch hinten wieder rauskommt.

Definition 3.36 (Perfect Reconstruction) *Die Filterbank* (F, G) *liefert* ***perfekte Rekonstruktion***[148]*, wenn in (3.59)* $c' = c$ *für alle Eingabedaten* c *gilt.*

Bemerkung 3.37

1. *Unter Verwendung der Modulationsmatrizen kann man die perfekte Rekonstruktion besonders nett beschreiben. Setzt man nämlich (3.47) in (3.58) ein, so erhält man, daß perfekte Rekonstruktion äquivalent zu*

$$\left[c^*\left(e^{2\pi ij/n}z\right) \,:\, j \in \mathbb{Z}_n\right] = \widetilde{M}(z)\,M(z)\left[c^*\left(e^{2\pi ij/n}z\right) \,:\, j \in \mathbb{Z}_n\right] \tag{3.60}$$

   ist. Eine Filterbank erlaubt also mit Sicherheit perfekte Rekonstruktion, wenn

$$\widetilde{M}(z)\,M(z) = I$$

   ist.

2. *Manchmal ist man etwas großzügiger und erlaubt, daß die Filterbank zwar die Eingabedaten rekonstruiert, erlaubt aber* Zeitverzögerungen, *d.h.,* $c' = \tau_j c$ *für ein* $j \in \mathbb{Z}$*. Nachdem*

$$(\tau_k c)^*(z) = z^{-k}\,c^*(z)$$

   ist, ergibt sich somit, daß in diesem Fall

$$\left[e^{-2\pi ijk/n}z^{-k}c^*\left(e^{2\pi ij/n}z\right) \,:\, j \in \mathbb{Z}_n\right] = \widetilde{M}(z)\,M(z)\left[c^*\left(e^{2\pi ij/n}z\right) \,:\, j \in \mathbb{Z}_n\right]$$

   was für

$$\widetilde{M}(z)\,M(z) = \operatorname{diag}\left[e^{-2\pi ijk/n}z^{-k} \,:\, j \in \mathbb{Z}_n\right]$$

   erfüllt ist.

3. *Man sieht leicht, daß*

$$\widetilde{M}(z)\,M(z) = \left[\sum_{\ell\in\mathbb{Z}} g_\ell^*\left(e^{2\pi ij/n}\,z\right) f_\ell^*\left(e^{2\pi ik/n}\,z\right) \,:\, j,k \in \mathbb{Z}_n\right]$$

   ist.

[148] In Englisch *"perfect reconstruction"*.

Es ist klar, daß die perfekte Rekonstruktion folgt, wenn $\widetilde{M}\,M = I$ ist – man muß das ja nur in (3.60) einsetzen. Was nicht ganz so offensichtlich ist, ist die Tatsache, daß auch die Umkehrung gilt.

Satz 3.38 (Perfect Reconstruction) *Eine Filterbank* (F, G) *besitzt die Fähigkeit zur perfekten Rekonstruktion genau dann, wenn* $\widetilde{M}(z)\,M(z) = I$, $z \in \mathbb{C}$.

Beweis: Die Richtung $\Leftarrow$ ist klar, das haben wie ja schon in Bemerkung 3.37 gesehen. Für die Umkehrung betrachten wir die äquivalente Form

$$0 = \left(I - \widetilde{M}(z)\,M(z)\right)\left[c^*\left(e^{2\pi ij/n}z\right) \,:\, j \in \mathbb{Z}_n\right]$$

von (3.60) und setzen $c = \tau_k\delta$, $k \in \mathbb{Z}_n$. Da in diesem Fall $c^*(z) = z^k$, also

$$\left[c^*\left(e^{2\pi ij/n}z\right) \,:\, j \in \mathbb{Z}_n\right] = z^k\left[e^{2\pi ijk/n} \,:\, j \in \mathbb{Z}_n\right]$$

ist und da $z^k \neq 0$ auf $\mathbb{T}$ ist, erhalten wir, daß

$$0 = \left(I - \widetilde{M}(z)\,M(z)\right)\left[e^{2\pi ijk/n} \,:\, j \in \mathbb{Z}_n\right].$$

Nun sind aber die Vektoren $\left[e^{2\pi ijk/n} \,:\, j \in \mathbb{Z}_n\right]$, $k \in \mathbb{Z}_n$, *linear unabhängig*, da sie zusammen die Inverse der DFT[149] aus Lemma 2.28 aufspannen. Und daher muß $I - \widetilde{M}(z)\,M(z)$ für alle $z \in \mathbb{T}$, also auch für alle $z \in \mathbb{C}$ gleich Null sein. □

3.5 Subdivision, Funktionen, Wavelets

Nachdem wir nun also die „guten" Filterbänke, also die mit perfekter Rekonstruktion, charakterisiert haben, werden wir jetzt mit **Kaskaden** von Filterbänken arbeiten und dabei letztendlich einen Bezug zu Funktionenzerlegungen herstellen. Da alle auszuführenden Rechenoperationen aber nach wie vor rein diskreter Natur sein werden, erhalten wir so einen Bezug zwischen der diskreten und der kontinuierlichen Welt.

Der Einfachheit halber[150] betrachten wir jetzt nur den Fall $n = 2$, in dem unsere **Analysefilterbank** ja von der Form

$$c \to \odot \begin{array}{l} \nearrow \boxed{F_0} \to \boxed{\downarrow_2} \to c_0 \\ \searrow \boxed{F_1} \to \boxed{\downarrow_2} \to c_1 \end{array}$$

ist. Und jetzt können wir natürlich c_0 und/oder c_1 wieder in dieselbe Analyse–Filterbank stecken und die Daten so weiter zerlegen, also

$$c \to \odot \begin{array}{l} \nearrow \boxed{F_0} \to \boxed{\downarrow_2} \to c_0 \to \odot \begin{array}{l} \nearrow \boxed{F_0} \to \boxed{\downarrow_2} \to c_{00} \\ \searrow \boxed{F_1} \to \boxed{\downarrow_2} \to c_{01} \end{array} \\ \searrow \boxed{F_1} \to \boxed{\downarrow_2} \to c_1 \to \odot \begin{array}{l} \nearrow \boxed{F_0} \to \boxed{\downarrow_2} \to c_{10} \\ \searrow \boxed{F_1} \to \boxed{\downarrow_2} \to c_{11} \end{array} \end{array} \tag{3.61}$$

[149] Bis auf den Normalisierungsfaktor $\frac{1}{n}$ natürlich.

[150] Das Ganze lässt sich aber auch mit beliebigem n machen, das ist nun wirklich überhaupt kein Mehraufwand, nur formales Getue.

und so weiter – auf diese Weise erhält man eine *Baumstruktur* von Signalen. In der "normalen" Waveletanalysis zerlegt man nur die Daten weiter, die aus dem "Tiefpassfilter" herauskommen, also

$$\begin{array}{ccccccccccc} & & \nearrow & \boxed{F_0} & \to & \boxed{\downarrow_2} & \to & c_0 \to \odot & \nearrow & \boxed{F_0} \to \boxed{\downarrow_2} & \to c_{00} \\ c \to \odot & & & & & & & & \searrow & \boxed{F_1} \to \boxed{\downarrow_2} & \to c_{01} \\ & & \searrow & \boxed{F_1} & \to & \boxed{\downarrow_2} & \to & c_1 & & & \end{array} \tag{3.62}$$

und so weiter, aber damit das alles sinnvoll ist, müssen F_0 und F_1 ein **Tiefpassfilter** bzw. ein **Hochpassfilter** sein und nachdem wir bisher ja noch nicht einmal spezifiziert haben, ob wir F_0 oder F_1 als Tiefpass ansehen, und was das überhaupt ist, bleibt uns erst einmal nichts anderes übrig, als symmetrisch zu zerlegen und so bei der Baumstruktur anzukommen. Diesen Ansatz bezeichnet man auch als **Wavelet Packages**.

Nach r derartigen Kaskadenschritten erhalten wir somit die Ausgabesignale

$$c_j^r \in \ell(\mathbb{Z}), \qquad j = 0, \dots, 2^r - 1,$$

wobei die *Binärdarstellung* des Index j genau die Filterungskaskade angibt, der unser Signal unerworfen wurde. Genauer: Ist

$$j = \sum_{k=0}^{r-1} \epsilon_k \, 2^k =: \epsilon_{r-1} \cdots \epsilon_0, \qquad \epsilon_k \in \{0, 1\},$$

dann ist

$$c_j^r = \downarrow_2 \, F_{\epsilon_0} \cdots \downarrow_2 \, F_{\epsilon_{r-1}} \, c.$$

Umgekehrt wird natürlich auch die Synthese wieder durch Kaskaden realisiert, diesmal durch Kaskaden der **Synthesefilterbank**:

$$\begin{array}{lllll} c_{00} \to \boxed{\uparrow_2} \to \boxed{G_0} & \searrow & & & \\ c_{01} \to \boxed{\uparrow_2} \to \boxed{G_1} & \nearrow & \oplus \to c_0 \to \boxed{\uparrow_2} \to \boxed{G_0} & \searrow & \\ & & & & \oplus \to c \\ c_{10} \to \boxed{\uparrow_2} \to \boxed{G_0} & \searrow & & & \\ c_{11} \to \boxed{\uparrow_2} \to \boxed{G_1} & \nearrow & \oplus \to c_1 \to \boxed{\uparrow_2} \to \boxed{G_1} & \nearrow & \end{array} \tag{3.63}$$

Diese Synthese–Kaskade erlaubt es uns nun, das Problem aus einer ganz anderen Persepktive zu sehen, indem wir die Analyse mal für einen Moment vergessen und die Synthese als Methode ansehen, Funktionen aus „einfachen" Eingabedaten zu generieren – das ist dann auch schon die Idee der **Subdivision**.

Und zwar geben wir uns jetzt ein Signal x vor, speisen das in c_0^r ein und sehen uns das Signal $c_r = c[x, r]$ an, das auf diese Art und Weise herauskommt. Offensichtlich ist dann

$$c_r = (G_0 \uparrow_2)^r \, x,$$

also, nach Lemma 3.35,

$$\begin{aligned} c_r^*(z) &= [(G_0 \uparrow_2)^r \, x]^*(z) = g_0^*(z) \left[\uparrow_2 \, (G_0 \uparrow_2)^{r-1} \, x\right]^*(z) \\ &= g_0^*(z) \left[(G_0 \uparrow_2)^{r-1} \, x\right]^*\left(z^2\right) = g_0^*(z) \, c_{r-1}^*\left(z^2\right), \end{aligned}$$

und somit, per Iteration,

$$c_r^*(z) = g_0^*(z)\cdots g_0^*\left(z^{2^{r-1}}\right)\, x^*\left(z^{2^r}\right) = \left[\prod_{j=0}^{r-1} g_0^*\left(z^{2^j}\right)\right] x^*\left(z^{2^r}\right). \tag{3.64}$$

Wegen des Upsamplings enthält die Folge $c = (G_0 \uparrow_2)^r\, x$ deutlich mehr Daten[151] als x, nämlich das 2^r–fache. Außerdem ist

$$S_G x := G_0 \uparrow_2 x = g_0 * (\uparrow_2 x) = \sum_{k\in\mathbb{Z}} g_0\,(\cdot - 2k)\; x(k)$$

und somit

$$\tau_2 S_G = S_G \tau \qquad \text{bzw.} \qquad \tau_{2^r} S_G^r = S_G \tau,$$

weswegen man die Folge $S_G^r x$ als diskrete Funktion an den Abszissen $2^{-r}k$, $k \in \mathbb{Z}$, auffassen sollte: Hat beispielsweise x zumindest lokal ein k–periodisches Verhalten[152], dann hat $S_G^r x$ lokal ein $2^r k$–periodisches Verhalten. Und was auch noch sofort auffällt: Da S_G ein linearer Operator ist und wir jede Folge x formal als

$$x = \sum_{k\in\mathbb{Z}} x(k)\,\tau_k\,\delta, \qquad \delta(j) = \delta_{j0}, \quad j \in \mathbb{Z},$$

schreiben können, ist

$$c_r^*(z) = \sum_{k\in\mathbb{Z}} z^k\, x(k) \left[\prod_{j=0}^{r-1} g_0^*\left(z^{2^j}\right)\right] \underbrace{\delta^*\left(z^{2^r}\right)}_{=1},$$

weswegen es genügt, sich auf $x = \delta$ zu beschränken.

Diese diskrete Funktion c, genauer, die diskrete Funktion $c\,(2^{-r}\cdot) \in \ell\,(2^{-r}\mathbb{Z})$, die an den dyadischen Punkten der Ordnung r definiert ist, wollen wir uns nun mal in der Fouriertransformation ansehen, das heißt, wir betrachten die Funktionen

$$\varphi_r(\xi) = \left[c_r\,(2^{-r}\cdot)\right]^\wedge(\xi), \qquad r \in \mathbb{N}_0.$$

Natürlich ist diese Funktion so erst einmal gar nicht wirklich definiert. Erinnern wir uns aber an die Beziehung (3.42) zwischen z–Transformation und Fouriertransformierter, so erhält man die Folge φ_r von trigonometrischen Polynomen[153] auch formal korrekt als

$$\varphi_r(\xi) = 2^{-r}\,\widehat{c}\,(\xi/2^r) = 2^{-r} c^*\left(e^{i\xi/2^r}\right) = \prod_{j=0}^{r-1} \frac{1}{2}\, g_0^*\left(e^{i\xi/2^{r-j}}\right) = \prod_{j=1}^{r} \frac{1}{2}\, g_0^*\left(e^{i\xi/2^j}\right),$$

[151] Das ist der Unterschied zwischen Daten und Information! Kennen wir x, wissen wir alles über unsere Folge, aber in c sind deutlich mehr Daten vorhanden als in x, wie man sich leicht überlegt, indem man $x = \delta$ betrachtet, denn dann enthält c_r ja 2^r von Null verschiedene Komponenten.

[152] Und dazu gehört insbesondere ein kompakter Träger!

[153] Wir bleiben jetzt bei $x = \delta$!

und wir können ganz vorsichtig die Grenzfunktion

$$\varphi(\xi) := \varphi_\infty(\xi) := \lim_{r\to\infty} \varphi_r(\xi) = \prod_{j=1}^{\infty} \frac{1}{2}\, g_0^*\left(e^{i2^{-j}\xi}\right) \tag{3.65}$$

definieren – vorausgesetzt natürlich, das unendliche Produkt konvergiert. Dann hat φ eine sehr nette Eigenschaft, nämlich

$$\begin{aligned}\varphi(\xi) &= \frac{1}{2}\, g_0^*\left(e^{i\xi/2}\right) \prod_{j=2}^{\infty} \frac{1}{2}\, g_0^*\left(e^{i2^{-j}\xi}\right) = \frac{1}{2}\, g_0^*\left(e^{i\xi/2}\right) \underbrace{\prod_{j=1}^{\infty} \frac{1}{2}\, g_0^*\left(e^{i2^{-j}(\xi/2)}\right)}_{=\varphi(\xi/2)} \\ &= \frac{1}{2}\, g_0^*\left(e^{i\xi/2}\right)\, \varphi\left(\frac{\xi}{2}\right). \end{aligned} \tag{3.66}$$

Wenn wir nun annehmen, daß $\varphi \in L_1(\mathbb{R})$ ist[154], denn dann können wir invers fouriertransformieren und erhalten so eine gleichmäßig stetige Funktion $\phi = \varphi^\vee$. Mit $\varphi = \widehat{\phi}$ wird (3.66) dann zu

$$\widehat{\phi}(\xi) = \frac{1}{2}\, g_0^*\left(e^{i\xi/2}\right) \widehat{\phi}\left(\frac{\xi}{2}\right) = \frac{1}{2}\widehat{g_0}\left(\frac{\xi}{2}\right) \widehat{\phi}\left(\frac{\xi}{2}\right) = \left[(g_0 * \phi)\,(2\cdot)\right]^\wedge(\xi),$$

also

$$\phi = (g_0 * \phi)\,(2\cdot) = \sum_{k\in\mathbb{Z}} g_0(k)\, \phi\,(2\cdot - k)\,. \tag{3.67}$$

Diese Gleichung, die man als **Verfeinerungsgleichung**, **Refinement Equation** oder **Zweiskalenbeziehung** bezeichnet, bedeutet, daß man die Funktion ϕ als Überlagerung von Translaten ihrer "gestauchten" Version darstellen kann, siehe Abb. 3.20. So etwas schafft nicht jede Funktion, sondern es ist eine *Forderung* an die Funktion. Und tatsächlich erhält man Funktionen, die (3.67) erfüllen, eigentlich auch nur als Ergebnis des Subdivision–Prozesses. Aber zu diesem Zweck braucht man natürlich Kriterien für die Konvergenz des unendlichen Produkts in (3.65). Klar ist natürlich, daß die Konvergenz eines unendlichen Produkts erzwingt, daß die einzelnen Glieder des Produkts gegen 1 konvergieren, so daß[155] ohne

$$1 = \frac{1}{2} \lim_{r\to\infty} g_0^*\left(e^{i2^{-r}\xi}\right) = \frac{1}{2}\, g_0^*(1) = \frac{1}{2}\, \widehat{g_0}(0) \tag{3.68}$$

gar nichts läuft. Eine *hinreichende* Bedingung für die Existenz einer stetigen Funktion ϕ, die die Zweiskalenbeziehung (3.67) liefert das folgende Resultat, das auf Daubechies (Daubechies, 1988) zurückgeht und das mitsamt Beweis aus (Vetterli & Kovačević, 1995) entnommen ist.

Proposition 3.39 (Existenz stetiger verfeinerbarer Funktionen) *Läßt sich* g_0^* *als*

$$g_0^*(z) = \left(\frac{1+z}{2}\right)^k q(z) \qquad \textit{mit} \qquad \max_{z\in\mathbb{T}} |q(z)| < 2^k \quad \textit{und} \quad q(1) = 2, \tag{3.69}$$

schreiben, dann existiert eine ***stetige*** *Lösung von (3.67).*

[154] Hier ist $\mathbb{R}$ anstelle von $\mathbb{T}$ wichtig, denn φ_r ensteht ja durch *Dilatation*, also Streckung, des trigonometrischen Polynoms $\widehat{c}$.

[155] Zumindest für rationale Filter f_0^* ohne Pol an $z = 1$, und nur solche interessieren uns ja hier eigentlich.

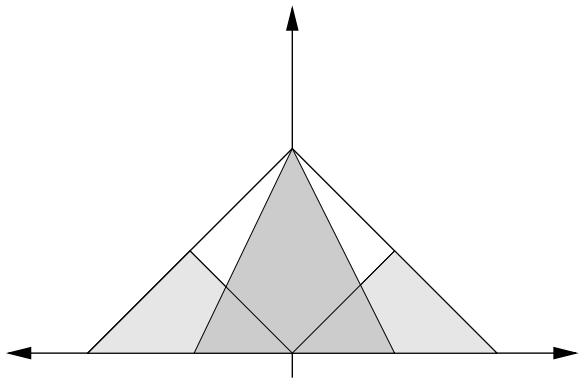


Abbildung 3.20: Wie man die stückweise lineare "Dachfunktion" als Zweiskalenbeziehung darstellt.

Beweis: Die Idee besteht darin, zu zeigen, daß unter der Voraussetzung (3.69) das unenedliche Produkt (3.65) konvergiert und eine L_1–Funktion liefert, deren inverse Fouriertransformation dann existieren muß und als ϕ gewählt werden kann.
Dazu zerlegen wir das Produkt in

$$\prod_{j=1}^{\infty} \frac{1}{2}\widehat{g_0}\,(2^{-r}\xi) = \prod_{j=1}^{\infty}\left(\frac{1+e^{i2^{-r}\xi}}{2}\right)^k \prod_{j=1}^{\infty} \frac{1}{2}\widehat{q}\,(2^{-r}\xi) \tag{3.70}$$

und behandeln die beiden Faktoren auf der rechten Seite separat. Da

$$\begin{aligned}\frac{1-e^{i\xi}}{\xi} &= \frac{1+e^{i\xi/2}}{2}\,\frac{1-e^{i\xi/2}}{\xi/2} = \frac{1+e^{i\xi/2}}{2}\,\frac{1+e^{i\xi/4}}{2}\,\frac{1-e^{i\xi/4}}{\xi/4}\\ &= \cdots = \underbrace{\frac{1-e^{i2^{-r}\xi}}{2^{-r}\xi}}_{\to i}\prod_{r=1}^{N}\frac{1+e^{i2^{-r}\xi}}{2},\end{aligned}$$

also

$$\prod_{r=1}^{\infty}\frac{1+e^{i2^{-r}\xi}}{2} = \frac{1-e^{i\xi}}{i\,\xi}$$

ist, hat das erste Produkt den Wert[156]

$$\left(\frac{1-e^{i\xi}}{i\,\xi}\right)^k = \left(e^{i\xi/2}\,\frac{e^{-i\xi/2}-e^{i\xi/2}}{\xi}\right)^k = (-i)^k e^{ik\xi/2}\left(\frac{\sin\xi/2}{\xi/2}\right)^k,$$

was im Absolutbetrag $\le C_1\,(1+|\xi|)^{-k}$ für eine passende Konstante $C_1 > 0$ ist.
Jetzt zum zweiten Faktor in (3.70), wo wir die Abkürzung $h = \frac{1}{2}q$ verwenden wollen. Da $h(1) = 1$ ist, gibt es eine Konstante $C_2 > 0$, so daß für $|\xi| \le 1$ die

[156]Wenn man genau hinschaut, erkennt man hier die Fouriertransformierte des zentrierten kardinalen B–Splines. Diese tauchen also wieder einmal an ganz zentraler Stelle auf.

Abschätzung[157] $|h(e^{i\xi})| \le 1 + C_2|\xi| \le e^{C_2|\xi|}$ erfüllt ist, also gilt für $|\xi| \le 1$ die Abschätzung

$$\prod_{j=1}^{\infty}\left|h\left(e^{i2^{-r}\xi}\right)\right| \le \prod_{j=1}^{\infty} e^{C_2 2^{-r}|\xi|} = \exp\left(\sum_{j=1}^{\infty}\frac{C_2|\xi|}{2^r}\right) = e^{C_2|\xi|} \le e^{C_2}. \tag{3.71}$$

Für beliebiges $\xi \in \mathbb{R}$ wählen wir nun $n \in \mathbb{N}$ so, daß $2^{n-1} \le |\xi| < 2^n$, und verwenden die folgende Aufspaltung zusammen mit (3.71):

$$\begin{aligned}\prod_{j=1}^{\infty}\left|h\left(e^{i2^{-r}\xi}\right)\right| &= \prod_{j=1}^{n}\left|h\left(e^{i2^{-r}\xi}\right)\right| \prod_{j=n+1}^{\infty}\left|h\left(e^{i2^{-r}\xi}\right)\right| \\ &= \prod_{j=1}^{n}\left|h\left(e^{i2^{-r}\xi}\right)\right| \prod_{j=1}^{\infty}\left|h\left(e^{i2^{-r}(\xi/2^n)}\right)\right| \le \prod_{j=1}^{n}\left|h\left(e^{i2^{-r}\xi}\right)\right| \; e^{C_2} \le B^n\, e^{C_2}\end{aligned}$$

mit

$$B := \max_{z\in\mathbb{T}}|h(z)| \le 2^{k-1-\varepsilon} \qquad \text{für} \varepsilon > 0.$$

Damit ist

$$B^n \le 2^{n(k-1-\varepsilon)} \le \underbrace{\left(2\,2^{n-1}\right)}_{\le 2\,|\xi| \le 1+|\xi|}{}^{k-1-\varepsilon} \le 2^k\;(1+|\xi|)^{k-1-\varepsilon}$$

und somit

$$\prod_{j=1}^{\infty}\left|g_0^*\left(e^{i2^{-r}\xi}\right)\right| \le C_1\;(1+|\xi|)^{-k}\; e^{C_2 k}\, 2^k\;(1+|\xi|)^{k-1-\varepsilon} \le C_3\,(1+|\xi|)^{-1-\varepsilon}.$$

Also ist das Produkt überall absolut konvergent und[158] die resultierende Funktion gehört zu $L_1(\mathbb{R})$ und läßt daher eine inverse Fouriertransformierte zu. □

Übung 3.10 Zeigen Sie: Ist ϕ eine (nichttriviale) Lösung der Zwei–Skalen–Gleichung (3.67), dann ist $\widehat{\phi}(0) \neq 0$ und $\widehat{g}(0) = 2$. ◊

Bemerkung 3.40 *Die Forderung (3.39) betrifft ja eigentlich f_0 und nicht die für die Zwei–Skalen–Gleichung (3.67) relevante Folge g_0. Da wir aber in (3.39) ja ohne weiteres z durch z^{-1} ersetzen können, ist es völlig irrelevant, ob die hinreichende Bedingung für f_0 oder g_0 formuliert wird.*

Sei nun also ϕ die[159] Lösung der Zwei–Skalen–Gleichung (3.67), dann gibt es drei Möglichkeiten, wie wir diese Funktion konstruierten können:

[157] Für wen diese Abschätzung etwas suspekt erscheint: Die Funktionen $1 + cx$ und e^{cx} haben für $x = 0$ denselben Wert aber die Ableitungen c und $c\,e^{cx} \le c$, also wächst die zweite Funktion schneller.

[158] Wer's genau haben will, braucht jetzt "dominated convergence"

[159] Das mit "die" oder "eine" ist in Wirklichkeit gar nicht so trivial!

1. Über die Fourier–Transformierte:

$$\phi = \left[\prod_{j=1}^{\infty} \frac{1}{2}\,\widehat{g_0}\left(-\frac{\cdot}{2}\right)\right]^{\vee}; \tag{3.72}$$

diese Werte könnte man an ganzzahligen Werten ausrechnen und dann die inverse FFT zur Bestimmung von ϕ verwenden.

2. Über das *Kaskaden–Schema*:

$$\phi = \lim_{j\to\infty} T_G^j\,\psi, \qquad T_G\psi := (g_0 * \psi)\,(2\cdot)\,, \tag{3.73}$$

mit einer beliebigen[160] Startfunktion ψ. Hier hat man es mit einem Verfahren zu tun, daß gegen den *Fixpunkt* ψ des *Transferoperators* T_G konvergiert.

3. Über das Subdivision–Schema:

$$\phi = \lim_{j\to\infty} S_G^j\,\delta, \qquad \lim_{j\to\infty}\sup_{k\in\mathbb{Z}} |\phi\,(2^{-r}k) - S_G\delta(k)| = 0, \tag{3.74}$$

wobei die Grenzfunktion an einem immer dichteren Gitter von dyadischen Punkten bestimmt wird.

Korollar 3.41 *Konvergeriert das Subdivision–Schema im Sinne von (3.74), so gilt für alle Ausgangsdaten* c

$$S_G^r c = \phi * c = \sum_{k\in\mathbb{Z}} c(k)\,\phi\,(\cdot - k)\,.$$

Und wo sind nun die Wavelets? Nur noch eine Interpretation weit weg. Wir betrachten jetzt, wo wir unsere schöne Funktion ϕ haben, ein Signal c nicht mehr als ein diskrete Funktion, sondern als **Koeffizienten** der Funktion

$$f_c := c * \phi = \sum_{k\in\mathbb{Z}} c(k)\,\phi(\cdot - k), \tag{3.75}$$

also der **Grenzfunktion** des Subdivision–Schemas. Das einfachste Beispiel für solche Approximanten sind die stückweise konstanten oder stückweise linearen Funktionen mit $\phi = \chi_{[0,1]}$ bzw. $\phi = \chi_{[0,1]} * \chi_{[0,1]}$ – letzteres ist die "Dachfunktion" aus Abb. 3.20. Da $\phi = T_G\phi$ ist

$$\begin{aligned} f_c &= \sum_{j\in\mathbb{Z}} c(j)\,\phi(\cdot - j) = \sum_{j\in\mathbb{Z}} c(j)\;(T_G\phi)\,(\cdot - j) = \sum_{j\in\mathbb{Z}} c(j)\;\sum_{k\in\mathbb{Z}} g_0(k)\,\phi\,(2\cdot - 2j - k) \\ &= \sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} c(j)\,g_0(k)\,\phi\,(2\cdot - 2j - k) = \sum_{k\in\mathbb{Z}} \underbrace{\sum_{j\in\mathbb{Z}} g_0\,(k-2j)\;c(j)}_{=S_G c(k)}\,\phi\,(2\cdot - k) \\ &= (S_G c * \phi)\,(2\cdot), \end{aligned}$$

[160] Na gut, mehr oder weniger beliebigen.

und damit entspricht $S_G c$ den Koeffizienten bezüglich der "gestauchten" Funktion $\phi(2\cdot)$. Und das genau bringt uns nun zu den Wavelets: Unser Ein- und damit auch Ausgangssignal c der "perfect reconstruction" Filterbank

$$c \to \odot \begin{array}{c} \nearrow \boxed{F_0} \to \boxed{\downarrow_2} \to c_0 \to \boxed{\uparrow_2} \to \boxed{G_0} \searrow \\ \searrow \boxed{F_1} \to \boxed{\downarrow_2} \to c_1 \to \boxed{\uparrow_2} \to \boxed{G_1} \nearrow \end{array} \oplus \to c \tag{3.76}$$

interpretieren wir als Koeffizienten einer Funktion $f = c * \phi(2\cdot)$, die zum Raum

$$V_1 = \text{span}\,\{\phi\,(2\cdot - k)\ :\ k \in \mathbb{Z}\}$$

gehört. Die Funktionen in diesem Raum sind „gestauchte" und um $k/2$, $k \in \mathbb{Z}$, verschobene Kopien der verfeinerbaren **Skalierungsfunktion** ϕ. Wegen der Zweiskalenbeziehung (3.67) ist auch $\phi \in V_1$ und da V_1 obendrein **translationsinvariant**[161] ist, erhalten wir, daß

$$V_1 \supseteq V_0 := \text{span}\,\{\phi\,(\cdot - k)\ :\ k \in \mathbb{Z}\}\,. \tag{3.77}$$

Setzen wir nun

$$V_j = \text{span}\left\{\phi\left(2^j \cdot - k\right)\ :\ k \in \mathbb{Z}\right\},$$

dann ist natürlich[162]

$$V_0 \subseteq V_1 \subseteq V_2 \subseteq \cdots$$

Damit bilden die Räume V_j, $j \in \mathbb{N}$ oder $j \in \mathbb{Z}$, eine **Multiresolution Analysis** oder **MRA**, wie sie von Mallat eingeführt wurde, siehe beispielsweise (Daubechies, 1992; Louis *et al.*, 1998; Mallat, 1989; Mallat, 1999; Vetterli & Kovačević, 1995) und noch viele weitere mehr. Die (minimalen) Eigenschaften einer MRA sind:

1. Eine aufsteigende Skala von Räumen $V_0 \subseteq V_1 \subseteq \cdots$
2. (Ganzzahlige) Translationsinvarianz der Räume V_j.
3. Skalenbeziehung: $f \in V_j \Rightarrow f(2\cdot) \in V_{j+1}$.

Wir haben in der Filterbank (3.76) unser Signal c oder alternativ die dazugehörige Funktion $c * \phi(2\cdot)$ in zwei Signale c_0 und c_1 zerlegt und müssen nun c_0 und c_1 passend interpretieren. Nach unserer Konstruktion entspricht nun die Filterung $c_0 = \downarrow_2 F_0\, c$, das heißt die Bestimmung der Funktion $c_0 * \phi$, einer **Projektion** von V_1 auf V_0, das heißt, es muss

$$c * \phi(2\cdot) \in V_0 \qquad \Leftrightarrow \qquad c_1 = 0. \tag{3.78}$$

gelten – das ist gerade die Eigenschaft, eine Projektion zu sein. Aber (3.78) folgt nun ganz einfach aus der Tatsache, daß die Modulationsmatrix M invertierbar ist: Gäbe es nämlich zwei Darstellungen c_0, c_1 und $\widetilde{c}_0, \widetilde{c}_1$, so daß

$$G_0\ \uparrow_2\ c_0 + G_1\ \uparrow_2\ c_1 = G_0\ \uparrow_2\ \widetilde{c}_0 + G_1\ \uparrow_2\ \widetilde{c}_1$$

[161] Ist $f \in V_1$, so ist auch $f(\cdot - k) \in V_1$ für alle $k \in \mathbb{Z}$.

[162] Das ist immer und immer wieder die Zweiskalenbeziehung (3.67), ohne die hier überhaupt nichts geht.

ist, dann wäre, nach Übergang zur z–Transform

$$0 = \widetilde{M}(z)\left(\begin{bmatrix} c_0^*(z^2) \\ c_1^*(z^2) \end{bmatrix} - \begin{bmatrix} \widetilde{c}_0^*(z^2) \\ \widetilde{c}_1^*(z^2) \end{bmatrix}\right) \qquad \Rightarrow \qquad \begin{bmatrix} c_0^*(z^2) \\ c_1^*(z^2) \end{bmatrix} = \begin{bmatrix} \widetilde{c}_0^*(z^2) \\ \widetilde{c}_1^*(z^2) \end{bmatrix}.$$

Da nun aber

$$V_0 \ni f = c * \phi = (S_G c * \phi)(2\cdot)$$

gilt, ist die Kombination $c_0 = S_G c$ und $c_1 = 0$ gerade diese eine Möglichkeit, $f \in V_0$ darzustellen, was (3.78) beweist.

Definition 3.42 *Das **Wavelet** zur Skalierungsfunktion ϕ ist die Funktion*

$$\psi = \sum_{k\in\mathbb{Z}} g_1(k)\,\phi(2\cdot - k). \tag{3.79}$$

Bemerkung 3.43 *Ist $n \geq 2$, so hat man nicht ein Wavelet, sondern $n-1$ Wavelets, die entsprechend als*

$$\psi_j = \sum_{k\in\mathbb{Z}} g_j(k)\,\phi(2\cdot - k), \qquad j = 1,\dots,n-1, \tag{3.80}$$

definiert sind.

Lemma 3.44 *Hat die Filterbank perfekte Rekonstruktion, so gilt für jedes Signal c*

$$c * \phi(2\cdot) = c_0 * \phi + c_1 * \psi. \tag{3.81}$$

Beweis: Perfekte Rekonstruktion bedeutet ja

$$c = G_0 \uparrow_2 c_0 + G_1 \uparrow_2 c_1 = \sum_{j=0}^{1}\sum_{k\in\mathbb{Z}} g_j(\cdot - 2k)\,c_j(k)$$

und daher ist

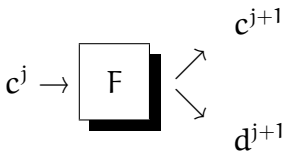
$$\begin{aligned}
c * \phi(2\cdot) &= \sum_{\ell\in\mathbb{Z}} c(\ell)\,\phi(2\cdot - \ell) \\
&= \sum_{\ell\in\mathbb{Z}}\sum_{j=0}^{1}\sum_{k\in\mathbb{Z}} g_j(\ell - 2k)\,c_j(k)\,\phi(2\cdot - \ell) \\
&= \sum_{k\in\mathbb{Z}}\sum_{\ell\in\mathbb{Z}}\sum_{j=0}^{1} g_j(\ell)\,c_j(k)\,\phi(2\cdot - (\ell + 2k)) \\
&= \sum_{k\in\mathbb{Z}} c_0(k) \underbrace{\sum_{\ell\in\mathbb{Z}} g_0(\ell)\,\phi(2(\cdot - k) - \ell)}_{=\phi(\cdot - k)} + \sum_{k\in\mathbb{Z}} c_1(k) \underbrace{\sum_{\ell\in\mathbb{Z}} g_0(\ell)\,\phi(2(\cdot - k) - \ell)}_{=\psi(\cdot - k)} \\
&= c_0 * \phi + c_1 * \psi.
\end{aligned}$$

□

Es ist fast ein bisschen enttäuschend, aber eigentlich ist das schon der wesentliche Punkt bei der Wavelet–Zerlegung von Funktionen, wir müssen das jetzt nur noch iterieren. Um zu verstehen, was wir da eigentlich machen, schreiben wir die Zerlegung der Filterbank ein klein wenig anders: Wir setzen

$$c^0 := c, \qquad c^{j+1} :=\downarrow F_0 c^j, \qquad d^{j+1} :=\downarrow F_0 c^j, \tag{3.82}$$

oder, kurz, knapp und schematisch,



und *iterieren* damit die Zerlegung immer auf dem Tiefpass–gefilterten Teil des Signals:

$$\begin{array}{ccccccccc} c = c^0 & \to & c^1 & \to & c^2 & \to \cdots \to & c^n & & c \simeq \left[c^n, d^1, \ldots, d^n\right]. \\ & \searrow & & \searrow & & & \searrow & & \\ & & d^1 & & d^2 & & d^n & & \end{array}$$

Besitzt die Filterbank perfekte Rekonstruktion, so kann man aus diesen Infomationen c auch wieder rekonstruieren, indem man den Prozess einfach umkehrt:

$$\begin{array}{ccccccccc} c^n & \to & c^{n-1} & \to & \cdots \to & c^1 & \to & c^0 = c. \\ & \nearrow & & \nearrow & & & \nearrow & \\ d^n & & d^{n-1} & & & d^1 & & \end{array}$$

Auf diese Art und Weise definiert jede perfekt rekonstruierende Filterbank eine

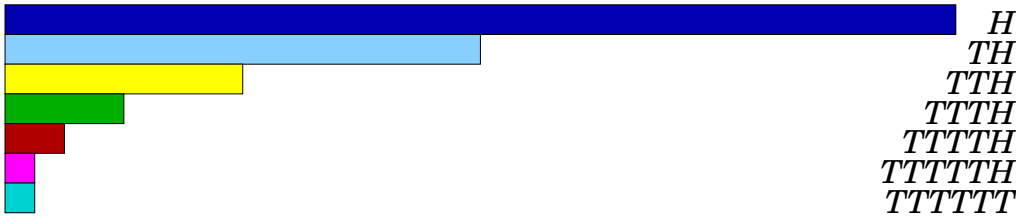


Abbildung 3.21: Die Pyramide (linksbündig und im Kopfstand). H und T indizieren die Anwendung von Hoch- bzw. Tiefpass.

umkehrbare Transformation

$$c \leftrightarrow \left[c^n, d^1, \ldots, d^n\right], \tag{3.83}$$

die man als **diskrete Wavelettransformation** oder **Pyramidenschema** bezeichnet. Der Name „Pyramidenschema“ rührt daher, daß d^j im wesentlichen die Hälfte des Informationsgehalts von c^{j-1} enthält.

Satz 3.45 (Wavelet–Zerlegung von Funktionen) *Besitzt die Filterbank perfekte Rekonstruktion, so ist*

$$c * \phi\,(2^n \cdot) = c^n * \phi + \sum_{j=1}^{n} d^j * \psi\left(2^{n-j}\cdot\right). \tag{3.84}$$

Beweis: Ist wieder erschreckend einfach: Wir verwenden einfach nur (3.81) und erhalten für jedes $x \in \mathbb{R}$

$$\begin{aligned} c * \phi\,(2^n x) &= c * \phi\left(2\left(2^{n-1}x\right)\right) = c^1 * \phi\left(2^{n-1}x\right) + d^1 * \psi\left(2^{n-1}x\right) \\ &= c^2 * \phi\left(2^{n-2}x\right) + d^2 * \psi\left(2^{n-1}x\right) + d^1 * \psi\left(2^{n-1}x\right) \\ &= \cdots = c^n * \phi(x) + \sum_{j=1}^{n} d^j * \psi\left(2^{n-j}x\right). \end{aligned}$$

und das ist genau[163] (3.84). □

Bemerkung 3.46 (Interpretation der diskreten Wavelettransformation) *Interessant ist nun die **Interpretation** der Aussage von Satz 3.45:*

1. *Auch wenn die Zerlegung eine Zerlegung im Sinne von Funktionen ist, laufen alle* Berechnungen *nur auf der Ebene der Koeffizienten und damit volldiskret ab. Und es werden nur die einfachen und effizient parallelisierbaren bzw. vektorisierbaren Strukturen der Filterbank verwendet, die sich „schlimmstenfalls“ sogar in Hardware implementieren liesse.*

2. *Die Funktion $c * \phi\,(2^n \cdot)$ auf der linken Seite von (3.84) ist wieder einmal ein **Quasiinterpolant**, so wie beispielsweise im Shannonschen Abtastsatz. In unserem Fall werden um den Faktor 2^n gestauchte, also ziemlich stark lokalisierte, und um $k/2^n$ verschobene, also ziemlich genau abtastende, Kopien von ϕ verwendet.*

3. *Diese „hochdetaillierte“ Funktion wird dann in Funktionen der Auflösungsstufen $2^{n-1}, 2^{n-2}, \ldots, 1$ zerlegt, die wesentlich weniger gestaucht und damit wesentlich geringer auflösend sind. Man geht also ganz bewusst vom hohen Detailniveau zu immer gröberen Darstellungen des Signals.*

4. *Wo bei der Zerlegung Detailinformationen landen müssen, ist nun intuitiv relativ naheliegend: in den **Waveletkoeffizienten** d^j und je feiner die Details sind, desto kleiner ist der zugehörige Index j.*

3.6 Und was kann man damit machen?

Das war jetzt zugegebenermaßen eine ganze Menge Theorie. Aber was bringt die uns? Eine ganze Menge, wenn wir uns das folgende grundsätzliche Prinzip vor Augen führen:

[163] Es mag vielleicht Verifizierungsformalisten geben, deren cerebral integrierter „Type Checker“ Probleme mit dieser Schreibweise hat, aber erstens verwendet (3.84) eine durchaus gebräuchliche und konsistente Notation und zweitens ist das dann deren Problem.

Die Waveletkoeffizienten werden von lokalen Hochpassfiltern erzeugt.

„Lokal“ bedeutet in diesem Kontext, daß die Filter nur endlichen Träger haben, und daß deswegen zum Wert $d^1(j)$ auch nur die Werte $c^0(k)$, $|j-k| \le N$, beitragen. Setzt man die Pyramide fort, so ergibt sich ganz analog, daß $d^r(j)$ nur von den Ausgangswerten $c^0(k)$, $|j - k/2^r| \le N$, abhängt. „Hochpass“ andererseits bedeutet, daß die Waveletkoeffizienten klein sind, wenn c im entsprechenden Bereich nahezu konstant ist, also eigentlich wenig zusätzliche Information enthält. Das lässt uns hoffen, daß sich derartige Signale ohne oder mit geringem Verlust durch die Zerlegung (3.84) komprimieren lassen. Das kann man auch tatsächlich quantifizieren, beispielsweise im folgenden Resultat, das wir hier allerdings weder beweisen können[164] noch wollen[165], aber wer will, findet die ganze Theorie in (Sauer, 2002a).

Satz 3.47 *Unter gewissen Voraussetzungen and* ϕ[166] *und* ψ[167] *gibt es eine Konstante* $C > 0$*, so daß für alle* $n+1$*–mal differenzierbaren Funktionen* f

$$\left|d^k(j)\right| \le C 2^{-k(n+1)} \left\|f^{(n+1)}\right\|_2, \qquad j \in \mathbb{Z}. \tag{3.85}$$

Dieses Resultat kann auch lokalisiert werden.

Viel interessanter ist, was uns Satz 3.47 sagt. Hat f überall die maximale Differenzierbarkeit $n+1$, dann fallen die Waveletkoeffizienten d^k allesamt wie $2^{-k(n+1)}$ ab. Ist hingegen f weniger glatt, also beispielsweise nur n'–mal differenzierbar, $n' < n$, dann sind die Voraussetzungen an ϕ und ψ immer noch erfüllt[168] und wir können den Satz wieder anwenden, um dann eine Abfallrate von nur noch $2^{-k(n'+1)}$ zu erhalten. Mit anderen Worten:

Je glatter eine Funktion ist, desto höher ist die zu erwartende Abfallrate.

Auch wenn wir eine Umkehrung von (3.85) nicht beweisen können[169], da Singularitäten von f an dyadischen Punkten, also an Punkten aus $2^{-k}\mathbb{Z}$, nicht erfasst werden können, funktioniert sie doch fast immer und wir können die Regularität von f um eine Stelle x herum durch

$$\lim_{k\to\infty} -k^{-1} \log_2 \left|d^k\left(\lfloor 2^k x \rfloor\right)\right|$$

[164] Man bräuchte dafür nochmal eine ganz ordentliche Menge Theorie.

[165] Diese Theorie würde uns vom eigentlichen Ziel ziemlich weit wegführen.

[166] Das sind kompakter Träger und die sogenannten **Strang–Fix–Bedingungen**, die dafür sorgen, daß sich Polynome vom Grad n in der Form $c * \phi$ darstellen lassen.

[167] Ebenfalls kompakter Träger und Achtung: Die Größe des Trägers geht direkt in die Konstante ein, die also explodieren würde, wenn die Endlichkeit des Trägers nicht mehr gewährleistet ist. Ausserdem braucht man hier wieder einmal $n+1$ **verschwindende Momente**, die ihrerseits stark an die Strang–Fix–Bedingung von ϕ gekoppelt sind.

[168] Es sind dann einfach weniger Bedingungen, die an die beiden Funktionen gestellt werden.

[169] Weil es sie schlicht und einfach in dieser Allgemeinheit nicht gibt!

abschätzen. Man kann die Abfallrate der Koeffizienten also zur Regularitätsbestimmung nutzen.

Und damit sind wir zurück bei der Anwendung in der Bild- und Signalverarbeitung, denn interessante Bestandteile von Bildern, die **Features**, zeichnen sich ja meistens dadurch aus, daß hier die Glattheit deutlich geringer ist als bei den „braven“ Bildkomponenten. Unter dieser Maxime werden die Waveletzerlegungen plötzlich zu einer „eierlegenden Wollmilchsau“ in Sachen Bild- und Signalverarbeitung, denn sie schaffen mehrere Jobs gleichzeitig: Bei der **Kompression** bildet man eine Wavelet–Zerlegung und behält nur einen gewissen Prozentsatz der Koeffizienten, nämlich die vom größten Absolutbetrag, bei der **Feature–Erkennung** sucht man nach Stellen, an denen die korrespondierenden Waveletkoeffizienten $d^k(2^{-k}j)$ **alle** betragsmäßig groß sind und beim **Entrauschen** setzt alle betragsmäßig kleinen Waveletkoeffizienten auf Null.

Beispiel 3.48 *Wir sehen uns einmal die Ecken einer einfachen “Dachfunktion” an, siehe Abb. 3.22, und erkennen hier sehr schön, daß die normalisierten Waveletkoeffizienten alle auf die Singularität “zeigen”. Wenn man genau hinsieht, dann kann man sogar erkennen, daß die Größe der Koeffizienten bei “spitzeren” Winkeln zunimmt – nicht so überraschend, denn je spitzer der Winkel, desto größer ist die Krümmung, also die zweite Ableitung*[170].

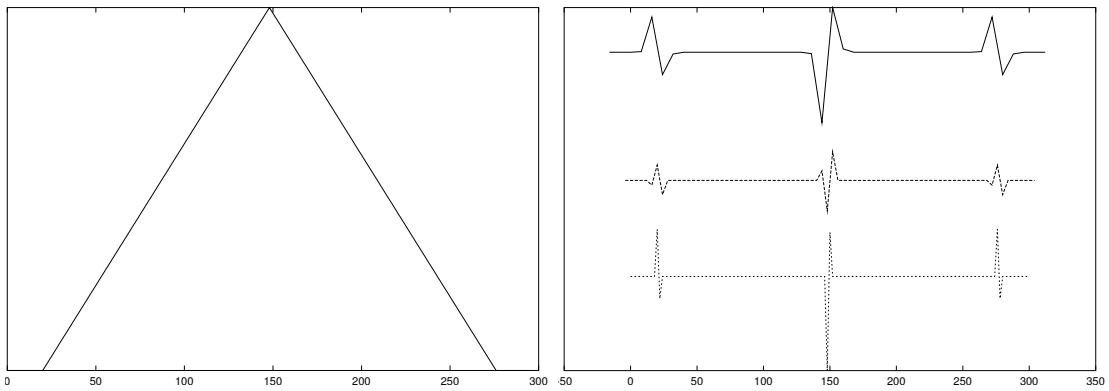


Abbildung 3.22: Eine stückweise lineare Funktion (links) und ihre Waveletkoeffizienten (rechts) bezüglich eines Wavelets mit hinreichender Strang–Fix–Ordnung.

Beispiel 3.48 ist recht eindrucksvoll, aber natürlich rein akademisch und künstlich. In der Realität sind Signale natürlich nicht so einfach und brav, sondern haben ein bißchen von allem.

Beispiel 3.49 *Das Signal in Abb 3.23 zeigt einen Ausschnitt aus einem EEG–Signal und die zugehörigen Waveletkoeffizienten. Man sieht doch, daß in diesem Fall die*

[170] Achtung: Das ist keine bewiesene Aussage, sondern lediglich pure Heuristik. Trotzdem ist sowas für die Intuition hilfreich.

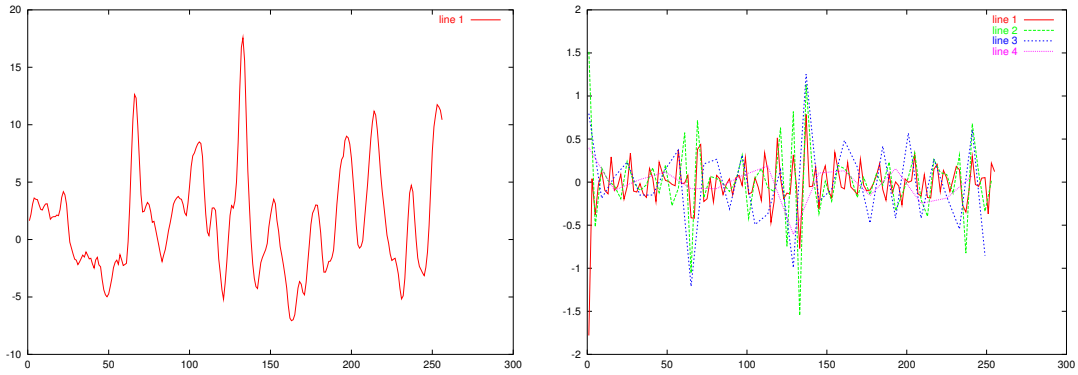


Abbildung 3.23: Ausschnitt aus einer Hirnstrommessung (links) und die zugehörigen Waveletkoeffizienten (rechts).

"Spitzen" wesentlich unschärfer sind und die "Ecken" und "Kanten" nicht so scharf charakterisieren.

Ein Problem, das wir bisher noch gar nicht beachtet haben, ist die Auswirkung der **Filterlänge** auf die Zerlegung. Klar ist, daß bei einer realistischen, getakteten Realisierung des Filters die Länge des Filters direkt zu einer Signalverzögerung führt[171], es hat aber auch Auswirkungen auf die Länge des Ergebnissignals. Sehen wir uns einmal ein Beispiel an, in dem wir $g * c$ berechnen, wobei der Filter g auf dem Intervall $[0, n]$, das Signal c auf dem Intervall $[0, N]$ „leben" soll. Die Faltung

$$g * c(j) = \sum_{k \in \mathbb{Z}} g(j-k)\, c(k) = \sum_{k=0}^{N} g(j-k)\, c(k)$$

ist nun für solche j von Null verschieden, für die $j - k \in [0, n]$, also $j \in k + [0, n]$, gilt, also für $j \in [0, N + n]$. Der Filter „verschmiert" also das Signal über dessen Grenzen hinaus und verlängert es so, und das umso intensiver je länger der Filter ist.

Bemerkung 3.50 (Artefakte) *Die Sache mit der Filterlänge hat aber noch eine wesentlich schwerwiegendere Konsequenz! Normalerweise kann man ja nur endliche Signale messen bzw. aufzeichnen*[172] *oder die Information ist von Natur aus endlich, z.B. bei Bildern. Das heißt aber nicht, daß man das Signal außerhalb des Messbereichs einfach als 0 oder periodisch fortsetzen dürfte – man weiß es einfach* ***nicht****! Nun braucht aber die Filterung am "Rand" des Signals Informationen "außerhalb" des Signals und diese kann man ja nur raten, was dann aber natürlich zu massiven Artefakten führen kann.*

Beispiel 3.51 (Artefakte) *Hier zwei Beispiele für Artefakte.*

[171] Was gerade bei Echtzeitanwendungen durchaus kritisch werden kann.
[172] Unsere Möglichkeiten sind halt immer nur endlich, leider oder glücklicherweise.

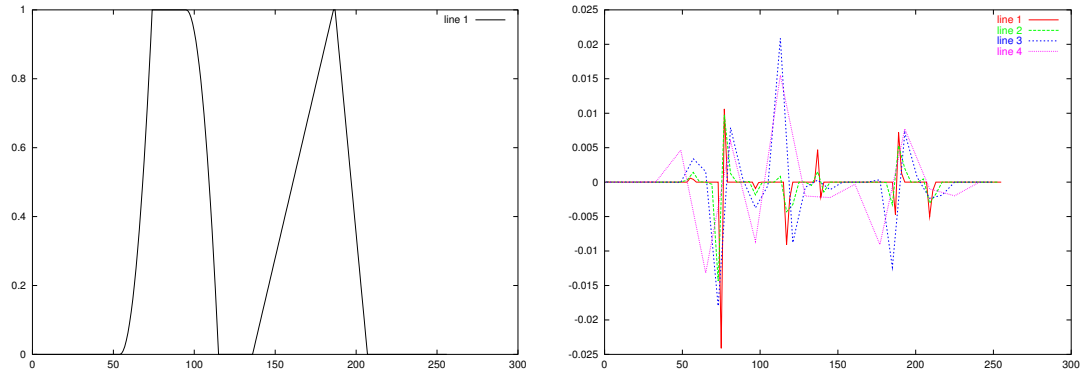


Abbildung 3.24: Ein einfaches Testsignal und die zugehörigen Waveletkoeffizienten. Wie man sieht, ist die Lokalisierung der Singularitäten weit davon entfernt, perfekt zu sein.

1. *In Abb. 3.24 ist ein "Testsignal" mit Übergängen verschiedener Glattheit zu sehen. Die Waveletkoeffizienten dazu deuten auch immer noch auf die Singularitäten, aber man sieht doch, daß die Lokalisierung nun wesentlich "unschärfer" ist. Außerdem tauchen am Rand jetzt signifikante Waveletkoeffizienten auf, die reine Artefakte sind, siehe Abb 3.25.*

2. *Auch Periodisierung ist nicht wirklich die Lösung, denn da kommt es darauf an,wie man periodisiert. Abb 3.26 zeigt dies ziemlich drastisch.*

Jetzt aber endlich zu der Art und Weise, wie man mit Wavelets Ecken erkennt, Entrauschen betreibt oder komprimiert – ganz egal, was man davon angeht, das Stichwort heißt[173] *Thresholding*.

Definition 3.52 *Für eine Konstante* $\theta \in \mathbb{R}_+$ *setzt ein* Thresholding–Operator $T_\theta : \ell(\mathbb{Z}) \to \ell(\mathbb{Z})$ *kleine Konstanten auf Null, und zwar vermittels*

1. (Hard thresholding)

$$T_\theta^h c(k) = \begin{cases} c(k), & |c(k)| \geq \theta, \\ 0, & |c(k)| < \theta, \end{cases} \qquad k \in \mathbb{Z}. \tag{3.86}$$

2. (Soft thresholding)

$$T_\theta^s c(k) = \operatorname{sgn} c(k)\ (|c(k)| - \theta)_+ = \begin{cases} c(k) - \theta, & c(k) \geq \theta, \\ 0, & |c(k)| \leq \theta, \\ c(k) + \theta, & c(k) < -\theta, \end{cases} \qquad k \in \mathbb{Z}. \tag{3.87}$$

[173] Auf gut Deutsch ...

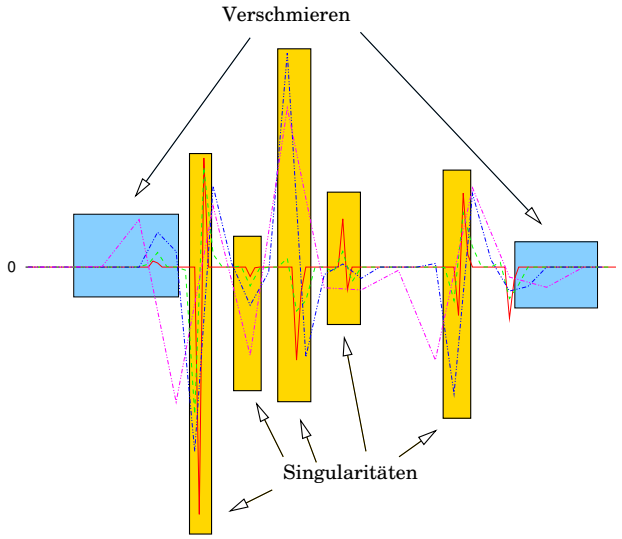


Abbildung 3.25: Erklärung der Waveletkoeffizienten in Abb 3.24: Die inneren Waveletkoeffizienten sind verschmiert, am Rand hingegen erscheinen Fortsetzungsartefakte.

Der Vorteil des "soft Thresholding" im Gegensatz zur "harten" Variante ist, daß die Abbildung

$$c \mapsto \operatorname{sgn} c \ (|c| - \theta)_+$$

stetig ist – allerdings wird auch das Signal c bei dieser Methode im "relevanten" Teil um die Konstante θ verringert. So ganz fällt allerdings auch das Soft Thresholding nicht vom Himmel, es ist nämlich Lösung eines Minimierungsproblems.

Proposition 3.53 *Für* $c \in \ell_{00}(\mathbb{Z})$ *ist*

$$T_\theta^s c = \operatorname{argmin} \|c - x\|_2^2 + 2\theta \ \|x\|_1 \tag{3.88}$$

Beweis: Nehmen wir an, daß c ausserhalb von $[0, N]$ verschwindet, dann muß natürlich auch die Minimallösung c^* von (3.88) Träger $[0, N]$ haben[174], so daß wir das zu minimierende Funktional aus (3.88) in

$$\lambda_c(x) = \sum_{j=0}^{N} (c(j) - x(j))^2 + 2\theta \sum_{j=0}^{N} |x(j)|$$

umschreiben können. Ist nun $c(j) \in \{0\} \cup [\theta, \infty)$, dann können wir $x(j) = c(j)$ wählen und besser geht es nicht. Andernfalls nehmen wir an, daß $c = c(j) \in (0, \theta)$ ist und suchen[175]

$$\min_x (x - c)^2 + 2\theta|x|, \qquad \Rightarrow \qquad 0 = 2(x - c) + 2\theta \operatorname{sgn} x$$

[174] Sonst käment ja nur unnötige Terme in (3.88) vor ...

[175] So ganz sauber ist der Beweis hier nicht, aber dafür einfach und anschaulich und er erspart uns weitere Fallunterscheidungen.

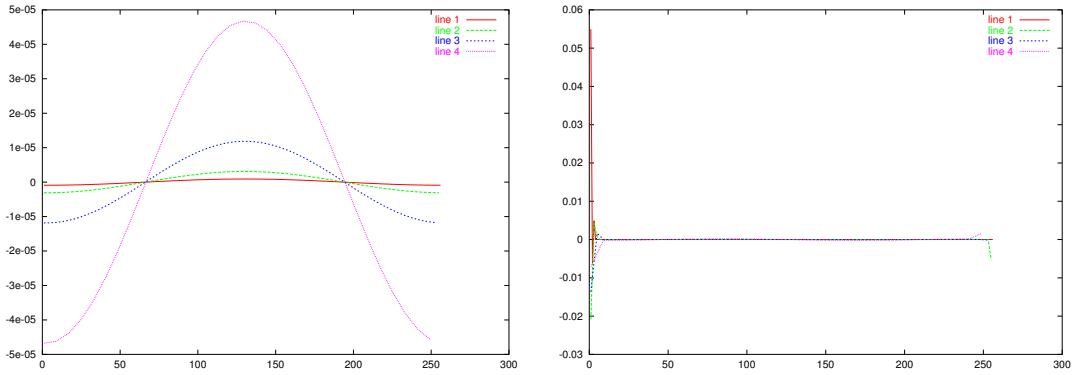


Abbildung 3.26: Waveletkoeffizienten zu periodischen Fortsetzungen der Sinunsfunktion, einmal mit Periode passend zu der des Sinus (links) und einma mit unpassender Periode (rechts). Man beachte die Skalierung der Achsen!

und da der erste Term für positives x sicherlich kleiner wird, erhalten wir, daß $x = c - \theta$. Dasselbe Argument können wir auch auf $c < 0$ anwenden, um auf $x = c + \theta$ zu kommen. Kombiniert man das, so ergibt sich gerade (3.87). □

Ein weiterer wesentlicher Freiheitsgrad besteht natürlich auch in der Wahl des Parameters θ, den man entweder fest und unabhängig von c wählen kann, aber besser in Abhängigkeit von c bestimmen sollte, beispielsweise[176]

- als $\lambda \|c\|_\infty$, $0 < \lambda < 1$, also als Bruchteil des größten auftretenden Wertes. Man wirft bei diesem Verfahren alles weg, was klein im Vergleich zu dieser Norm ist.
- als Bruchteil des Mittelwerts:
$$\theta = \lambda \frac{\|c\|_1}{\#c} = \frac{\lambda}{\#c} \sum_{k \in \mathbb{Z}} |c(k)|\,, \qquad \#c := \#\{k \,:\, c(k) \neq 0\}\,.$$
- als relativen Median[177] für $0 < \lambda < 1$:
$$\theta = |c(k)| \qquad \text{so daß} \qquad \#\{j \,:\, |c(j)| \leq |c(k)|\} \sim \lambda \#c.$$

All diese Ansätze haben ihre Vor- und Nachteile, der größte Vorteil des absoluten θ besteht aber sicherlich darin, daß dabei natürlich kein weiterer Rechenaufwand anfällt, die Bestimmung des Medians ist im wesentlichen so schwer wie Sortieren der Folge, die Berechnung der Mittelwerte hingegen eine Summation.

Unter Verwendung dieses Operators können wir nun die Anwendungen skizzieren. Zuerst berechnet man immer eine Waveletzerlegung[178] der Form

$$c \mapsto \left[d^1, \ldots, d^n, c^n\right],$$

[176] Bei diesen Ansätzen muss man natürlich annehmen, daß $c \in \ell_{00}(\mathbb{Z})$ ist.
[177] Den eigentlichen Median erhält man für $\lambda = \frac{1}{2}$.
[178] Und wir sind jetzt zurück bei den Filtern ...

auf die wir nun folgendermaßen Thresholding anwenden:

Eckenerkennung: Je nachdem welcher Ordnung[179] die Ecken sein sollen, bestimmen wir $\widehat{d^k} = T_\theta 2^{kn} d^k$ für ein relativ *großes* θ und sehen uns die Lokalisierung dieser Koeffizienten an. Häufen sie sich irgendwo, dann ist das ein Indikator für eine Singularität.

Kompression: Hier bilden wir $\widehat{d^k} = T_\theta d_k$ sowie $\widehat{c^k} = T_\theta c_n$, wobei θ die Kompressionsrate und damit auch die Qualität des Ergebnisses[180] kontrolliert, und speichern dann nur die von Null verschiedenen Koeffizienten von $\left[\widehat{d^1}, \ldots, \widehat{d^n}, c^n\right]$.

Klassifizierung: Das ist schon ein etwas subtileres Thema. Hier bestimmt man die Waveletzerlegung und extrahiert wenige *maximale* Waveletkoeffizienten, was zu einem Muster aus Werten (Absolutbetrag) und Indizes (Lage der "Ecke") führt. Mit anderen Worten: es ergibt sich ein relativ kleiner Vektor von Zahlen und dieser Vektor wird dann in ein "normales" Klassifizierungssystem[181] gefüttert.

Bei der Erweiterung auf Bilder hält sich der Mehraufwand nun massiv in Grenzen, das Stichwort ist wieder **Tensorprodukt**, so wie wir es bei der Fouriertransformation auch schon gemacht haben.

Definition 3.54 *Seien ϕ_1, ϕ_2 Skalierungsfunktionen[182], dann ist die* Tensorprodukt–Skalierungsfunktion *definiert als*

$$\Phi(x, y) = \phi(x)\,\phi(y), \qquad (x, y) \in \mathbb{R}^2.$$

Die zugehörigen Wavelets[183] sind dann

$$\Psi_1(x, y) = \phi(x)\,\psi(y), \qquad \Psi_1(x, y) = \psi(x)\,\phi(y), \qquad \Psi_3(x, y) = \psi(x)\,\psi(y).$$

Damit dieses Φ und seine Wavelets eine MRA genieren, brauchen wir in erster Linie Verfeinerbarkeit und die Zerlegung. Also beweisen wir es mal. Dazu nehmen wir an, daß wir die Zwei–Skalen–Gleichungen

$$\phi_j = a_j * \phi\,(2\cdot) = \sum_{k\in\mathbb{Z}} a_j(k)\,\phi\,(2\cdot - k)\,, \qquad j = 1, 2,$$

und die Zerlegungsformel

$$\phi_j\,(2\cdot) = c_j' * \phi_j + d_j' * \psi_j, \qquad j = 1, 2,$$

zur Verfügung haben, siehe Definition 3.14.

[179] Sprünge, Ecken, Krümmungsdefekte (= zweite Ableitung) ...

[180] Wie man ja von JPEG her weiß, laufen diese beiden Interessen einander entgegen – hohe Kompression ist halt leider nicht ohne Qualitätsverlust zu haben.

[181] Neuronales Netzwerk oder Lerntheorie heißen hier die Stichworte.

[182] Es ist in keinster Weise verboten, in unterschiedliche Richtungen unterschiedliche Wavelets zu verwenden.

[183] Ja, hier steht der Plural. Und zwar mit Absicht!

Satz 3.55 *Die Funktion Φ ist verfeinerbar:*

$$\Phi = \sum_{\alpha \in \mathbb{Z}^2} (a_1 \otimes a_2)(\alpha)\Phi(2\cdot - \alpha), \tag{3.89}$$

und es gilt

$$\Phi(2\cdot) = c * \Phi + \sum_{j=1}^{3} d_j * \Psi_j, \tag{3.90}$$

wobei

$$c = c_1' \otimes c_2', \qquad d_1 = c_1' \otimes d_2', \qquad d_2 = d_1' \otimes c_2', \qquad d_3 = d_1' \otimes d_2'. \tag{3.91}$$

Übung 3.11 Beweisen Sie Satz 3.55. ◊

Wir haben es jetzt also mit *drei* Wavelets zu tun: Zwei der Wavelets ergeben sich als Tensorprodukt einer Skalierungsfunktion mit einem Wavelet, und eines als Tensorprodukt der beiden Wavelets.

Beispiel 3.56 *Sehen wir uns das einmal im allereinfachsten Fall*[184] *an, daß $\phi_1 = \phi_2 = \chi_{[0,1]}$ und damit $\psi_1 = \psi_2 = \chi_{[0,\frac{1}{2}]} - \chi_{[\frac{1}{2},1]}$. Die resultierenden Funktionen sind dann in Abb. 3.27 zu sehen.*

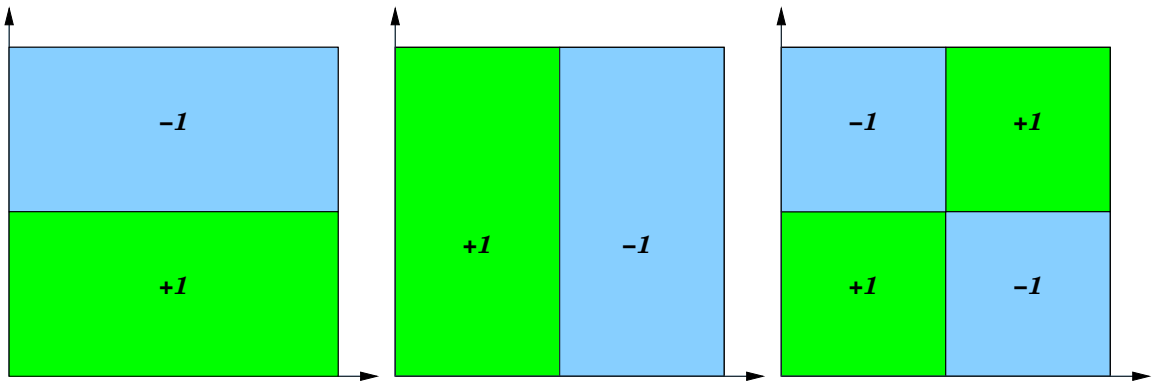


Abbildung 3.27: Die drei Tensorprodukte zum Haar–Wavelet: Skalierungsfunktion in x und Wavelet in y *(links)*, Wavelet in x und Skalierungsfunktion in y *(mitte)* und Wavelet in beiden Variablen *(rechts)*.

Die drei Wavelets haben nun unterschiedliche Fähigkeiten: Ψ_1 erkennt Kanten parallel zur x–Achse, Ψ_2 solche parallel zur y–Achse, während sich Ψ_3 um die diagonalen Kanten kümmert. Die Zerlegung eines Bildes läßt sich dann schematisch wie in Abb 3.28 darstellen, die "Durchführung" dieser Zerlegung landet wieder bei den Tensorprodukt–Filtern, die sich ja recht einfach realisieren lassen, indem man einfach die univariaten Filter in eine "Kaskade" schaltet.

Als einfaches Anwendungsbeispiel zerlegen wir einmal unser Standardbild vemittels Wavelets, wobei jetzt der Skalierungsanteil *oben links* gespeichert

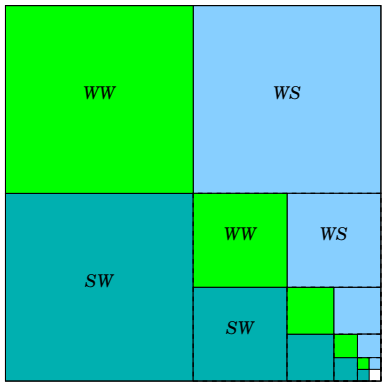


Abbildung 3.28: Die Waveletzerlegung für Bilder: Man zerlegt in drei Anteile mit Waveletbeteiligung und einen Anteil der "vergröberten" Skalierungsfunktion und zerlegt, wie bei eindimensionalen Signalen, letzteren dann weiter.

werden soll. Das ist in Abb. 3.29 zu sehen. Hierbei wurden die sogenannten **Daubechies–Wavelets** „D4" verwendet[185], die auch Bestandteil des JPEG2000–Standards sind. Und man sieht leicht, warum, denn die Waveletekoeffizienten sind in der Tat ziemlich dünn besetzt.

Tatsächlich gehören die Daubechies–Wavelets zu einer besonderen Klasse von Filterbänken, sie liefern nämlich **Spiegelfilter**, bei denen $G(z) = F(z^{-1})$ ist, die Rekonstruktionsfilter sind also nur gespiegelte Kopien der Analysefilter. Der Vorteil ist klar: Man muss deutlich weniger Koeffizienten speichern. Zu Spiegelfiltern etc. sei auf (Daubechies, 1992; Vetterli & Kovačević, 1995) verwiesen.

[184] Das ist das gute alte haar–Wavelet.
[185] Die „4" steht für die vier **Taps**, also von Null verschiedenen Einträge des Filters.

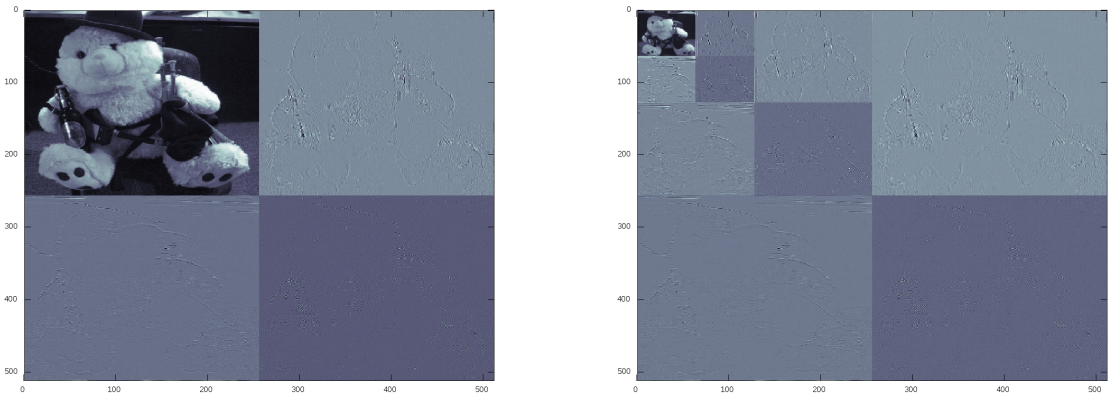


Abbildung 3.29: Eine *(links)* und drei *(rechts)* Stufen der Wavelet–Zerlegung. Man sieht recht gut die Erkennung horizontaler, vertikaler und „diagonaler“ Kanten und daß die Waveletanteile extrem dünn besetzt sind.

Es ist unglaublich, wie unwissend die studierende Jugend auf Universitäten kommt, wenn ich nur 10 Minuten rechne oder geometrisiere, so schlft ein Viertel derselben sanft ein.

G. Chr. Lichtenberg

4 Optimierung & Bildmanipulation

Leider ist es in der Realität mit der Messung von Signalen und Bildern nicht so einfach! Nur sehr selten und mit sehr großem Aufwand kann man genau messen und auch wirklich das messen, was man messen will. Oftmals sucht man ein Signal f, kann aber nur

$$g = Tf + \epsilon \tag{4.1}$$

messen, wobei T ein (hoffentlich linearer) Operator ist[186] und ϵ *zufälliges* Rauschen. Die Zufälligkeit sorgt dafür, daß ein gewisses Maß an Stochastik nicht ganz vermeidbar ist.

4.1 Regularisierung oder wie man unterbestimmte Probleme löst

Jetzt beschäftigen wir uns mit der *deterministischen* Komponente in (4.1), nämlich dem Operator T, der alles mögliche sein kann, aber selten invertierbar! Na gut, wenn T eine Matrix ist[187], dann können wir natürlich *jede* Matrix "invertierbar" machen, indem wir die Singulärwertzerlegung

$$T = U\Sigma V^T, \qquad \sigma_{11} \geq \sigma_{22} \geq \cdots \geq 0$$

mit orthogonalen Matrizen U und V und einer Diagonalmatrix[188] Σ verwenden, die auch für nichtquadratische Matrizen definiert ist, siehe z.B. (Golub & van Loan, 1996; Horn & Johnson, 1985; Sauer, 2000b). Die **Pseudoinverse** oder **Moore–Penrose–Inverse** zu T ist dann als

$$T^+ = V\Sigma^+U^T, \qquad (\Sigma^+)_{jk} = \begin{cases} 0, & \sigma_{jk} = 0, \\ \sigma_{jk}^{-1}, & \sigma_{jk} > 0, \end{cases}$$

[186] Was den Sonderfall $T = I$ direkter Messungen ja nicht ausschließt.

[187] Und bei endlichen diskreten Signalen sind alle linearen Operatoren Matrizen oder zumindest als solche darstellbar.

[188] Genau gesagt ist Σ eine Matrix, bei der alle Nicht–Diagonalelemente verschwinden, schließlich muss Σ nicht quadratisch sein!

definiert, und man könnte als Lösung $f = T^+g$ wählen, denn wenn es eine Lösung des linearen Systems $g = Tf$, dann muß auch $f = T^+g$ eine sein. Normalerweise ist das Gleichungssystem auch *unterbestimmt*, das heißt, das zugrundeliegende Modell ist komplexer als die gemessenen Größen.

Beispiel 4.1 *Wir betrachten ein univariates Signal und als Operator T die Mittelung von 4 benachbarten Werten gefolgt von Downsampling um Faktor 2, also*

$$\begin{aligned} T &= \begin{bmatrix} \ddots & & & & & & \\ & 1 & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 1 & 0 & 0 & \\ & 0 & 0 & 0 & 0 & 1 & \\ & & & & & & \ddots \end{bmatrix} \frac{1}{4} \begin{bmatrix} \ddots & & & & & & \\ & 1 & 1 & 1 & 1 & & \\ & & 1 & 1 & 1 & 1 & \\ & & & 1 & 1 & 1 & 1 \\ & & & & & & \ddots \end{bmatrix} \\ &= \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 & & & \\ & & 1 & 1 & 1 & 1 & \\ & & & & & & \ddots \end{bmatrix} \in \mathbb{R}^{n \times 2n+2}. \end{aligned}$$

Beschränken wir uns auf $n = 3$, dann ist beispielsweise

$$T\,[1\,0\,0\,0\,0\,0\,0\,1]^T = \begin{bmatrix} 1/4 \\ 0 \\ 1/4 \end{bmatrix}, \qquad T\,[1\,2\,3\,4\,5\,6\,7\,8]^T = \begin{bmatrix} 2.5 \\ 4.5 \\ 6.5 \end{bmatrix}$$

und die Rekonstruktionen über die Pseudoinverse ergeben

$$\left[\frac{1}{2}\ \frac{1}{2}\ 0\ 0\ 0\ 0\ \frac{1}{2}\ \frac{1}{2}\right] \qquad \textit{und} \qquad [2.5\ 2.5\ 2.5\ 2.5\ 6.5\ 6.5\ 6.5\ 6.5]\,,$$

was nicht sonderlich berauschend ist.

Das grundlegende Problem bei der ganzen Geschichte besteht darin, daß ein **inverses Problem** $Tf \mapsto f$ im Normalfall immer **schlecht gestellt**, auf Englisch **ill posed**, ist und daß die Pseudoinverse keine wirkliche Lösung des Problems darstellen kann und darstellt – ganz abgesehen davon, daß man es oft auch mit "richtigen Operatoren" und nicht "nur" mit Matrizen zu tun hat.

Der beste Ansatz besteht nun darin, entweder unter f unter Einhaltung der Nebenbedingung $Tf = g$ ein möglichst problembezogenes **Energiefunktional** E minimieren zu lassen, also

$$\min_f E(f), \qquad Tf = g, \tag{4.2}$$

zu bestimmen, oder aber Nebenbedingung und Energiefunktional in ein parametrisiertes Minimierungsproblem zu packen:

$$\min_f \|Tf - g\| + \lambda E(f), \qquad \lambda \in \mathbb{R}_+, \tag{4.3}$$

wobei man beim zweiten Ansatz sogar auf die „exakte" Lösung $Tf = g$ verzichtet, die in Anwesenheit von Rauschen ohnehin nicht wirklich sinnvoll ist, wenn man dafür die Funktion f „einfacher" oder „glatter" bekommen kann.

Das einfachste Energiefunktional ist sicherlich $E(f) = \|f\|_2^2$. In diesem Fall wird (4.2) für $f \in \ell_{00}(\mathbb{Z})$ zu

$$\min \sum_{j=0}^{N} f(j)^2, \qquad Tf = g,$$

was sich mittels Lagrange–Multiplikatoren, siehe z.B. (Heuser, 1983; Nocedal & Wright, 1999; Sauer, 2002b; Spellucci, 1993), in

$$0 = \nabla E(f) - \nabla^T (g - Tf)\,\mu = 2f + T^T\mu, \qquad Tf = g,$$

also als das lineare Gleichungssystem

$$\begin{bmatrix} 2I & T^T \\ T & 0 \end{bmatrix}\begin{bmatrix} f \\ \mu \end{bmatrix} = \begin{bmatrix} 0 \\ g \end{bmatrix}$$

ergibt. Für unsere Daten aus Beispiel erhalten wir auch genau dasselbe Ergebnis wie mit der Pseudoinversen, nicht weiter verwunderlich, wenn man sich die Rolle der Pseudoinversen bei der Least–Squares–Aproximation in Erinnerung ruft, siehe (Björck, 1996; Golub & van Loan, 1996). Der Ansatz aus (4.3) hingegen liefert uns als zu minimierende Funktion

$$\|Tf - g\|_2^2 + \lambda\,\|f\|_2^2 = (Tf - g)^T (Tf - g) + \lambda f^T f = f^T \left(T^T T + \lambda I\right) f - 2g^T Tf + g^T g,$$

was nach f abgeleitet und gleich Null gesetzt das Gleichungssystem

$$\left(T^T T + \lambda I\right) f = T^T g$$

liefert. Für $\lambda \to 0$ erhalten wir dann dieselbe Lösung wir für das Optimierungsproblem (4.2), für große λ eher eigenwillige Werte.

So weit also nichts neues. Versuchen wir uns also einmal an anderen Energiefunktionalen, beispielsweise an

$$\|\nabla f\|_2^2 = \sum_{j=0}^{N-1} (f(j+1) - f(j))^2 = \|Df\|^2 = f^T D^T D f, \qquad D = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}$$

Da nun $\nabla E(f) = 2D^T Df$ ist[189], erhalten wir jetzt die Gleichungssysteme

$$\begin{bmatrix} 2D^T D & T^T \\ T & 0 \end{bmatrix}\begin{bmatrix} f \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ g \end{bmatrix} \qquad \text{und} \qquad \left(T^T T + \lambda D^T D\right) f = T^T g.$$

Während hier der erste Vektor immer noch schlecht rekonstruiert wird, das Ergebnis ist der `Octave`–Vektor

[189] Im Gegensatz zu $\nabla E(f) = 2I$ vorher.

```
 0.583333
 0.416667
 0.083333
-0.083333
-0.083333
 0.083333
 0.416667
 0.583333
```

erhalten wir für unseren linearen Vektor `(1:8)` schon das deutlich bessere Ergebnis

```
1.5909
1.9545
2.6818
3.7727
5.2273
6.3182
7.0455
7.4091
```

bei der Rekonstruktion. Verwenden wir hingegen den "Laplace–Operator" $\Delta = f(\cdot + 2) - 2f(\cdot + 1) + f(\cdot)$, dann ist

$$\|\Delta f\|_2^2 = \|D_2 f\|_2^2 = f^T D_2^T D_2 f, \qquad D = \begin{bmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{bmatrix},$$

und die Ergebnisse sind

```
 0.653846
 0.346154
 0.076923
-0.076923
-0.076923
 0.076923
 0.346154
 0.653846
```

und **exakte** Rekonstruktion von `(1:8)`.

Der Vorteil quadratischer Funktionale ist offensichtlich: Man kann sie sehr leicht und systematisch in lineare Gleichungssysteme umformen, wobei die Anteile $D^T D$ in diesen Gleichungssystemen auch noch eine recht nette Struktur haben: Sie sind bandiert. Normalerweise verwendet man allerdings eher die Version (4.3), da dieser Ansatz wesentlich „rauschtoleranter" ist, denn wir haben ja eben nicht Tf gemessen, sondern $Tf + \epsilon$ ... Außerdem ist es hier auch völlig egal, ob die Nebenbedingung $Tf = g$ überhaupt erfüllbar ist, sie wird einfach so gut erfüllt wie möglich.

Die Auswahl des jeweiligen Energiefunktionals[190] ist problemabhängig und wird in der Bildverarbeitung auch lebhaft diskutiert. Das einfachste Funktional, das in der Bildverarbeitung häufig verwendet wird, ist in der kontinuierlichen Formulierung von (4.3)

$$E(f) = \int_\Omega \|\nabla f\|_2^2 + \lambda \int_\Omega (g - Tf)^2 , \tag{4.4}$$

meist mit $T = I$ verwendet.

Ein anderer Ansatz verwendet die sogenannte **TV–Norm**, die die **totale Variation** misst[191],

$$\|f\|_{TV} := \int_\Omega \|\nabla f\|_1 = \int_\Omega \sum_{j=1}^{n} \left| \frac{\partial f}{\partial x_j}(x) \right| \, dx \tag{4.5}$$

und wesentlich sensibler für Ecken ist. Der Funktionen mit endlicher TV–Norm bilden einen Banachraum, oft als $BV(\Omega)$ geschrieben, nämlich den Raum der Funktionen **von beschränkter Variation**. In der Tat kann man zeigen, siehe (Mallat, 1999, S. 37, Theorem 2.7), daß es eine enge Beziehung zwischen dieser Norm und den Konturkurven eines Bildes gibt. In unserem diskreten Beispiel ist nun

$$\|f\|_{TV} = \sum_{j=0}^{N-1} |f(j+1) - f(j)| = \|Df\|_1 .$$

Die Lösung von

$$\min_f \|f\|_{TV} = \|Df\|_1 , \qquad Tf = g$$

erhält man als Lösung eines linearen Optimierungsproblems

$$\min \sum_{j=0}^{N-1} |f(j+1) - f(j)| , \qquad Tf = g.$$

Um zu sehen, daß das wirklich ein „normales“' lineares Programm ist, setzen wir $y_i = f(j+1) - f(j)$ und die zu minimierende Summe

$$\sum_{j=0}^{N-1} |y_j| = \sum_{j=0}^{N-1} y_j + u_j, \qquad u_j = \begin{cases} 0, & y_j \geq 0, \\ -2\,y_j, & y_j < 0, \end{cases}$$

was wir in die Ungleichungsnebenbedingungen

$$0 \leq u_j, \qquad -2y_j \leq u_j, \qquad j = 0, \ldots, N-1,$$

[190] Man kann diese natürlich auch wieder kombinieren, also beispielsweise die Norm des Vektors mit denen seiner ersten und zweiten Ableitung “ausbalancieren”.

[191] Als Vektornorm wird, je nach Literaturstelle, entweder $\|\cdot\|_1$ oder $\|\cdot\|_2$ verwendet; für einen Variationsansatz ist das nicht so relevant, denn da wird das Optimierungsproblem in eine Differentialgleichung umgeformt, für den Optimierungsansatz allerdings schon, weil man dann eben nach der Diskretisierung ein lineares Programm erhält oder nicht.

umschreiben lässt – die Minimierung sorgt dann schon dafür, daß u_j automatisch den richtigen Wert annimmt. Die Zielfunktion ist dann

$$\sum_{j=0}^{N-1} (f(j+1) - f(j)) + u_j = f(N) - f(0) + \sum_{j=0}^{N-1} u_j$$

und das[192] lineare Problem hat die Form

$$\min_{f,u} f(N) - f(0) + \sum_{j=0}^{N-1} u_j, \qquad \begin{bmatrix} T & 0 \\ -T & 0 \\ 0 & I \\ 2D & I \end{bmatrix} \begin{bmatrix} f \\ u \end{bmatrix} \geq \begin{bmatrix} g \\ -g \\ 0 \\ 0 \end{bmatrix} \tag{4.6}$$

mit

$$D = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & -1 \end{bmatrix},$$

was sich nun wirklich mit dem **Siplexalgorithmus** in Angriff nehmen lässt. Zumindest im diskreten Fall und ohne Auftreten von Rauschen kann man sich also den Variationsansatz für die Minimierung der totalen Variation erst einmal sparen. Da ℓ_1–minimale Vektoren eine viel stärkere Tendenz zu kleinen Trägern haben, könnten wir hier eine bessere Rekonstruktion unseres ersten Beispielvektors erwarten.

4.2 Es rauscht mal wieder

Jetzt aber zurück zu unseren verrauschten Daten aus (4.1). Es ist klar, daß man ohne Annahmen an das Rauschen keine vernünftigen Aussagen machen kann. Und wenn man schon annimmt, dann kann man auch gleich **die** Standardannahme machen, nämlich daß das Rauschen normalverteilt mit Mittelwert Null und Standardabweichung σ ist, daß also

$$\int \epsilon = 0, \qquad \int \epsilon^2 = \sigma^2$$

gilt, denn dann sind die Nebenbedingungen unserer inversen Probleme nur unwesentlich anders und ergeben sich zu

$$0 = \int \epsilon = \int g - Tf \qquad \text{und} \qquad \sigma^2 = \int \epsilon^2 = \int (g - Tf)^2 .$$

Und wenn wir σ nicht kennen? Dann könnten wir es einfach als einen weiteren Parameter wählen und so schätzen, daß das Ergebnis möglichst gut wird – wir „erklären“ die Daten unter der Annahme des „freundlichsten“ Rauschens:

$$\min_f E(f), \qquad \int g - Tf = 0, \quad \int (g - Tf)^2 = \sigma^2. \tag{4.7}$$

[192] Zuerst einmal unbeschränkte, siehe z.B. (Sauer, 2002b).

Allerdings hat das einen kleinen Nachteil, denn neben dem Rauschen könnten auch andere „hochfrequente“ Anteile eines Bildes, nämlich Texturen, nur sehr schwer vom Rauschen zu unterscheiden sein – ein verrauschtes Fernsehbild eines Zebras ist ein schönes Beispiel. Nun ist die Definitionsgleichung des Fehlers

$$\epsilon = g - Tf$$

ja eigentlich sehr einfach, und wann immer wir eine Schätzung für f bestimmt haben, ist das geschätzte[193] Rauschen ebenfalls festgelegt. Daher verwendet man Verfahren, die *iterativ* aus dem geschätzten Rauschen eine neue Approximation bestimmen, daraus dann wieder ein neues Rauschen und so weiter.

Der Algorithmus selbst setzt einfach $\epsilon_0 = 0$ und bestimmt für $k = 1, 2, \ldots$ die Approximation f_k als Lösung des Minimierungsproblems

$$\min_f \int_\Omega \|\nabla f\|_1 + \lambda \int_\Omega (g - Tf + \epsilon_{k-1})^2 \tag{4.8}$$

und setzen dann $\epsilon_k = g - Tf_k$. In (Osher *et al.*, 2004) wird gezeigt, daß für $T = I$, also den Fall „reinen“ Entrauschens, die Folge der f_k gegen das verrauschte Signal $g + \epsilon$ konvergiert, aber in der BV–Norm sich zuerst einmal dem *unverrauschten* Originalsignal f annähert, zumindest so lange, bis $\|f_k - f\| \leq \sigma^2$ ist. Deswegen führt man Algorithmus so lange durch, bis das Rauschen der f_k, also das ϵ_k, größer wird oder bis man bis auf σ^2 an der gesuchten Lösung heran ist.

4.3 Bildverarbeitung als Optimierungsproblem

„Moderne“ Bildverarbeitung ist zu einem großen Teil Optimierung. Daß das so ist, basiert auf der Idee, die wir in diesem Kapitel bereits bei den inversen Problemen gesehen haben. Wenn wir ein irgendwie deformiertes, also beispielsweise verrauschtes, mit Artefakten behaftetes oder nur teilweise vorhandenes Bild $\widehat{B}$ haben, so sucht man immer nach dem „Idealbild“ das sich hinter $\widehat{B}$ verbirgt. Das **Idealbild** B ist in diesem Fall ein Bild, das zwei Forderungen gleichzeitig möglichst gut erfüllt:

Datentreue: Das Bild B muss immer noch nahe bei $\widehat{B}$ oder zumindest bei dem bekannten Teil von $\widehat{B}$ liegen.

Regularität: Das Bild soll gewisse Strukturannahmen erfüllen, die sich natürlich aus einem dahinterliegenden Bildmodell ergeben und die codieren, was ein „schönes“ bzw. gutes Bild ist.

Beide Eigenschaften messen wir über ein passend gewähltes **Funktional**. Die einfachste Variante für Datentreue ist natürlich

$$E(B, \widehat{B}) := \left\| B - \widehat{B} \right\|,$$

[193] Klingt besser als „geraten“.

wobei $\|\cdot\|$ eine passend gewählte Norm[194] ist, die die Ähnlichkeit der Bilder misst. Interpretiert man die Bilder als **Pixelmatrizen** $B = [b_{jk} : j,k]$ und $\widehat{B} = \left[\widehat{b}_{jk} : j,k\right]$, dann ist die einfachste Norm

$$\|A\|_2 = \sqrt{\sum_{j,k=1}^{m,n} a_{jk}^2}, \qquad \text{d.h.} \qquad \left\|B - \widehat{B}\right\|_2^2 = \sqrt{\sum_{j,k=1}^{m,n} \left|b_{jk} - \widehat{b}_{jk}\right|^2},$$

die sogenannte **Frobeniusnorm**. Auf dieser Norm beruht das bekannteste Ähnlichkeitsmaß für Bilder, die *Peak Signal to Noise Ratio*, kurz **PSNR**, die als

$$20\log_{10}\left(I\left(\frac{1}{\sqrt{mn}}\left\|B - \widehat{B}\right\|_2\right)^{-1}\right) \tag{4.9}$$

definiert ist, wobei I hier nicht für eine Einheitsmatrix, sondern für die **maximale Pixelintensität** steht. Die PSNR läuft zwischen 0 für absolute Ungleich-

Abbildung 4.1: Gleich oder nicht gleich? Zwei Fotos an unterschiedlichen Tagen aus in etwa derselben Position von in etwa demselben Objekt.

heit[195] und ∞ bei zwei identischen Bildern; im letzteren Fall ist ja $\|B - \widehat{B}\| = 0$ und die Division durch Null wird als ∞ interpretiert. Ob die PSNR wirklich ein gutes Maß für die Ähnlichkeit von Bildern ist, ist eien vieldiskutierte Frage. Die beiden Bilder in Abb 4.1 würde man normalerweise als ähnlich bezeichnen, ihre PSNR ist aber der recht bescheidene Wert von 14.628. Warum das so ist, zeigt das Differenzbild in Abb. 4.2, indem sich praktisch alle Bildstrukturen erkennen

[194] Das verdient eine lange Fußnote! Der Begriff der **Norm** axiomatisiert ja in der Mathematik nur den Begriff der Länge (eine Länge ist nie Null, streckt man einen Vektor, so streckt sich die Länge um denselben Faktor und Umwege sind immer länger) und die Norm eine Differenz gibt einen vernünftigen Abstandsbegriff. Anders gesagt: Wir messen hier, wie ähnlich die Bilder sind. Solche Normen können, vor allem, wenn sie auf wahrnehmungspsychologischen Konzepten beruhen, sehr schnell sehr komplex werden.

[195] Alle Pixel unterscheiden sich um die maximale Pixelintensität, das heisst, im einen Bild wird der Wert 0, im anderen der Wert I angenommen.

lassen. Und für die PSNR ist es völlig egal, ob nur die Helligkeitswerte von Pixeln unterschiedlich sind, oder ob die Bilder wirklich signifikant verschieden sind. Tatsächlich ist die PSNR zwischen dem linken Bild und dem Bild, das nur

Abbildung 4.2: Differenzbild der beiden Bilder aus Abb 4.1. Erstaunlicherweise sind die größten Werte bei den *gemeinsamen* architektonischen Bildelementen zu finden, aber die Helligkeitswerte sind eben unterschiedlich.

konstant aus seinem mittleren Helligkeitswert besteht, mit 10.974 gar nicht mal so katastrophal schlechter. Noch extremer ist das Beispiel in Abb. 4.3, wo zwei

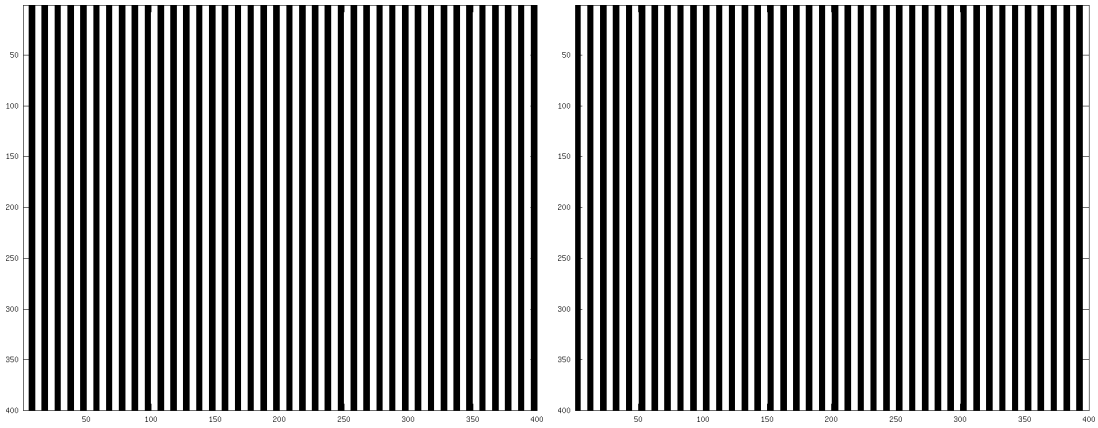


Abbildung 4.3: Zwei Bilder mit einer PSNR von 0, obwohl es sich dabei nur um periodisch verschobene Kopien voneinaner handelt.

optisch identische Bilder eine PSNR von 0 haben, weil sie gerade passend so verschoben wurden, daß die weißen und schwarzen Streifen einander überlappen. Der langen Rede kurzer Sinn: Die Wahl des Datentreue–Funktionals ist alles andere als eine einfache Sache und die hier vorgestellte einfache Wahl entspricht nicht unbedingt unseren Vorstellungen von Gleichheit. Eine einfache

Erweiterung, die man sich vorstellen könnte, ist die Frobeniusnorm zwischen geeignet verschobenen Bildern zu messen, also einen Ausdruck wie

$$E(B, \widehat{B}) := \min_{\tau} \left\| B - \tau\widehat{B} \right\|_2, \tag{4.10}$$

zu minimieren, wobei τ wieder einmal für eine Translation steht. Nur ist das Funktional in (4.9) für den Mathematiker ziemlich unangenehm, denn es ist keine Norm mehr und damit erheblich schwerer zu minimieren.

Das **Regularitätsfunktional** $R(B)$ ist schon deutlich komplexer. In vielen Fällen beinhaltet es wie schon in (4.4) und (4.5) den Gradienten ∇B, den wir ja schon länger als einen Indikator für Kanten und andere Unstetigkeiten im Bild kennen. Beispiele sind also

$$R(B) = \int_\Omega \left\| \nabla B(x) \right\|_2^2 \, dx \qquad \text{oder} \qquad R(B) = \int_\Omega \left\| \nabla B(x) \right\|_1 \, dx,$$

es gibt aber auch andere, komplexere Funktionale, die sich noch stärker and der menschlichen Bildwahrnehmung und der Form „idealer" Kanten in Bildern orientieren. Tatsächlich ist die Suche nach derartigen „guten" Regularitätsfunktionalen ein wichtiges Forschungsthema der modernen Bildverarbeitung.

Das letztendliche Optimierungsproblem ist dann relativ unspektakulär: Zu einem Parameter $\lambda > 0$, der das gewünschte Verhältnis von Datentreue zu Regularität festlegt, bestimmt man das Bild B^* als Lösung des unbeschränkten Minimierungsproblems

$$\min_B E(B, \widehat{B}) + \lambda R(B). \tag{4.11}$$

Ist $\lambda = 0$, so wird nur versucht, das Bild zu reproduzieren[196], für $\lambda \to \infty$ hingegen wird nur regularisiert und die Vorgabe $\widehat{B}$ komplett ignoriert. Die Frage, wie man diesen magischen Parameter λ wählen sollte, ist mehr als berechtigt, eine allgemeingültige Antwort oder auch nur ein gutes Patentrezept gibt es nicht wirklich. Unter gewissen Verteilungsannahmen[197] gibt es allerdings statistische Ansätze, die unter dem Namen **Kreuzvalidierung** bzw. **cross validation** bekannt sind, siehe (Craven & Wahba, 1979; Golub *et al.*, 1979).

Die Auswahl bzw. Entwicklung geeigneter Verfahren zur Lösung der Optimierungsprobleme ist ebenfalls ziemlich nichttrivial! In vielen Fällen versucht man, E und R so zu wählen, daß $E + \lambda R$ konvex ist. Zur Erinnerung[198]: Eine Funktion $f : \Omega \to \mathbb{R}$ heißt **konvex**, wenn

$$f(\alpha x + (1-\alpha)x') \le \alpha f(x) + (1-\alpha) f(x'), \qquad x, x' \in \Omega, \quad 0 \le \alpha \le 1, \tag{4.12}$$

erfüllt. Konvexe Funktionen haben die angenehme Eigenschaften nur globale Minima zu haben, aber keine globalen Maxima, das heißt, jede Lösung von $\nabla f(x) = 0$ ist automatisch ein Minimum. Noch schöner als Konvexität ist allerdings **strikte Konvexität**, bei der (4.12) für $0 < \alpha < 1$ mit *strikter Ungleichung*

[196] Sozusagen um jeden Preis.
[197] Aber auch die wollen erst einmal erfüllt sein!
[198] Oder zum Neu–Lernen!

„<“ erfüllt ist. Das verhindert dann auch so unerfreuliche Situationen wie mit beispielsweise mit der Funktion $f(x) = |x|$, die konvex ist, ein globales Minimum an $x = 0$ hat, aber erst einmal keine Stelle mit $f'(x) = 0$. Man kann das beheben, braucht aber deutlich mehr Mathematik[199].

Übung 4.1 Hat jede konvexe Funktion $f : \mathbb{R} \to \mathbb{R}$ ein globales Minimum? ◊

Eine Beschreibung der Optimierungsverfahren, die im Kontext der Bildverarbeitung Anwendung finden, ist eine Vorlesung für sich! Generell hat man es in (4.11) ja mit einem Problem zu tun, in dem über *Bilder*, also über *Funktionen* minimiert wird – schließlich haben wir das Problem ja *kontinuierlich* formuliert. Minimierungsprobleme, deren Argumente Funktionen sind, bezeichnet man als **Variationsprobleme**. Dort führt die bewährte Strategie *„ableiten und gleich Null setzen“* zu einem System von Differentialgleichungen, den sogenannten **Euler–Lagrange–Gleichungen**, und eine Methode zur Bestimmung der Optimallsung besteht darin, diese Gleichungen zu lösen. Alternativ kann man aber auch die Problemstellung diskretisieren, in dem man beispielsweise das Bild „pixelisiert“ und dan diese diskreten Probleme lösen.

4.4 Typische Anwendungen

Zum Abschluss sehen wir uns noch an, wie für ein paar typische Anwendungen der Bildverabeitung der Datentreueterm gewählt weden soll, und zwar in der kontinuierlichen Formulierung. Als Regularisierer nimmt man zumeist die **totale Variation** $\int \|\nabla B\|_1$, die sich in der Bildverarbeitung als ein Standard etabliert hat.

Entrauschen: Hier geht es natürlich im wesentlichen um Nähe zum Originalbild, so daß sich für das Datentreuefunktional das gute alte

$$\int_\Omega \left\| B(x) - \widehat{B}(x) \right\|_2^2 \, dx$$

anbietet, zumal so auch gleich die PSNR maximiert wird, was bei Ingenieuren immer einen guten Eindruck hinterlässt.

Impainting: Hierbei hat man es mit einem Bild zu tun, das nur auf einem Teilbereich $\Omega' \subset \Omega$ vorliegt, weil anderen Bildbereiche entweder entfernt wurden oder irgendwann verlorengegangen sind. Ziel ist es natürlich, der noch vorhandenen Bildinformation möglichst nahe zu kommen, weswegen

$$\int_{\Omega'} \left\| B(x) - \widehat{B}(x) \right\|_2^2 \, dx$$

[199] Der Trick besteht darin, an Stellen, an denen die Funktion nicht differenzierbar ist, den **Subgradient** ∂f zu betrachten, eine *mengenwertige* Form der Ableitung, die die Steigungen aller möglichen Tangenten zusammenfasst, was im Falle von Differenzierbarkeit mit der „normalen“ Ableitung zusammenfällt, denn da gibt es dann nur genau eine Tangente. In unserem Beispiel ist $\partial|\cdot|(0) = [-1, 1]$ und die Bedingung $f'(x) = 0$ wird dann zu $0 \in \partial f(x)$.

sicherlich eine gute Wahl ist. Will man die Originaldaten wirklich behalten, so ist das Funktional fast noch einfacher:

$$E(B, \widehat{B}) = \begin{cases} 0, & B(\Omega') = \widehat{B}(\Omega'), \\ \infty, & \text{sonst.} \end{cases}$$

Segmentierung: Hier versucht man Bildteile herauszulösen und so beispielsweise zwischen Vorder- und Hintergrund zu unterscheiden. Das so erzeugte Bild B muss nun keine direkte Ähnlichkeit mehr mit $\widehat{B}$ haben, sondern kann beispielsweise nur zwischen 0 und 1 verlaufen. Das führt zu dem Funktional

$$E(B, \widehat{B}) = \left\langle B, \widehat{B} \right\rangle + \iota_{[0,1]}(B) = \int_\Omega B(x)\widehat{B}(x)\,dx + \begin{cases} 0, & B(\Omega) \subset [0,1], \\ \infty, & \text{sonst.} \end{cases}$$

Das war also noch ein Schnelldurchlauf durch diese Methoden. Wenn man es genau nimmt, dann beginnt jetzt, wo die Vorlesung zuende ist, eigentlich die „richtige“ Bildverarbeitung, aber auch die basiert auf vielen der Ideen, die wir hier gesehen und kennengelernt haben.

4 Literatur

Akhieser, N. I. (1988). *Lectures on Integral Transforms*, volume 70 of *Translations of Mathematical Monographs*. AMS.

Björck, A. (1996). *Numerical Methods for Least Squares Problems*. SIAM.

Cooley, J. W. (1987). The re–discovery of the Fast Fourier Transform. *Mikrochimica Acta*, **3**:33–45.

Cooley, J. W. (1990). How the FFT gained acceptance. In Nash, S. G., editor, *A History of Scientific Computing*, pages 133–140. ACM–Press and Addison–Wesley.

Cooley, J. W., Tukey, J. W. (1965). An algorithm for machine calculation of complex Fourier series. *Math. Comp.*, **19**:297–301.

Craven, P., Wahba, G. (1979). Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math.*, **31**:377–403.

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Commun. on Pure and Appl. Math.*, **41**:909–996.

Daubechies, I. (1992). *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM.

DeVore, R. A., Lorentz, G. G. (1993). *Constructive Approximation*, volume 303 of *Grundlehren der mathematischen Wissenschaften*. Springer.

Doblhofer, E. (1990). *Zeichen und Wunder. Geschichte und Entzifferung verschollener Schriften und Sprachen*. Paul Neff Verlag, Wien. Lizenzausgabe Weltbild Verlag.

Domes, J. (2007). Schnelle inverse Wavelettransformation. Zulassungsarbeit zum ersten Staatsexamen, Justus–Liebig–Universität Gießen.

FFTW (2003). `FFTW` – the Fastest Fourier Transform in the West. `http://www.fftw.org`.

Foley, J., van Dam, A., Feiner, S., Hughes, J. (1990). *Computer Graphics*. Addison Wesley, 2nd edition.

Forster, O. (1984). *Analysis 3. Integralrechung im $\mathbb{R}^n$ mit Anwendungen*. Vieweg, 3. edition.

Gabor, D. (1946). Theory of communication. *J. IEEE*, **93**:429–457.

Gautschi, W. (1997). *Numerical Analysis. An Introduction*. Birkhäuser.

Golub, G., Heath, M., Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, **21**:215–223.

Golub, G., van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, 3rd edition.

Grossmann, A., Morlet, J., Paul, T. (1985). Transforms associated to square integrable group representations. i.general results. *J. Math. Phys.*, **26**:2473–2479.

Grüningen, D. C. v. (1993). *Digitale Signalverarbeitung*. VDE Verlag, AT Verlag.

Hamming, R. W. (1989). *Digital Filters*. Prentice–Hall. Republished by Dover Publications, 1998.

Handels, H. (2000). *Medizinische Bildverarbeitung*. B. G. Teubner.

Hardy, G. H., Rogosinsky, W. W. (1956). *Fourier Series*. Cambridge University Press, 3. edition. Republished by Dover Publications, 1999.

Heuser, H. (1983). *Lehrbuch der Analysis. Teil 2*. B. G. Teubner, 2. edition.

Higham, N. J. (1996). *Accuracy and stability of numerical algorithms*. SIAM.

Holschneider, M. (1995). *Wavelets: an analysis tool*. Clarendon Press, Oxford.

Horn, R. A., Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press.

Hough, P. V. C. (1962). Method and means for recognizing complex patterns. US Patent 3069654.

Hubbard, B. B. (1996). *The world according to wavelets*. A.K. Peters.

Isaacson, E., Keller, H. B. (1966). *Analysis of Numerical Methods*. John Wiley & Sons.

Jähne, B. (2002). *Digitale Bildverarbeitung*. Springer.

Kammeyer, K. D., Kroschel, K. (1998). *Digitale Signalverarbeitung*. Teubner Studienbücher Elektrotechnik. B. G. Teubner, Stuttgart.

Katznelson, Y. (1976). *An Introduction to Harmonic Analysis*. Dover Books on advanced Mathematics. Dover Publications, 2. edition.

Klein, A. (2011). *Zur Numerik kontinuierlicher Wavelet- und Matrixwavelet–Transformationen*. PhD thesis, Justus–Liebig–Universität Gießen.

Knuth, D. E. (1998). *The Art of Computer Programming. Seminumerical Algorithms*. Addison–Wesley, 3rd edition.

Langenbucher, A., Reese, S., Sauer, T., Seitz, B. (2004). Matrix based calculation schemes for toric intraocular lenses. *Ophthal. Physiol. Opt.*, **24**:511–519.

Lorentz, G. G. (1966). *Approximation of functions*. Chelsea Publishing Company.

Louis, A. K., Maaß, P., Rieder, A. (1998). *Wavelets*. B. G. Teubner, 2. edition.

Mallat, S. (1989). Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. *Trans. Amer. Math. Soc.*, **315**:69–87.

Mallat, S. (1999). *A Wavelet Tour of Signal Processing*. Academic Press, 2. edition.

Mhaskar, H. M., Pai, D. V. (2000). *Fundamentals of Approximation Theory*. Narosa Publishing House.

Nocedal, J., Wright, S. J. (1999). *Numerical Optimization*. Springer Series in Operations Research. Springer.

Olafsson, G., Quinto, E. T., editors (2006). *The Radon Transform, Inverse Problems, and Tomography*, volume 65 of *Proceedings of Symposia in Applied Mathematics*. AMS.

Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W. (2004). An iterative regularization method for total variation based image restoration. Technical report, UCLA.

Pascual-Marqui, R. D., M., M. C., Lehmann, D. (1994). Low resolution electromagnetic tomography: a new method for localizing electrical activity in the brain. *International Journal of Psychophysiology*, **18**:49–65.

Paul, R. P. (1981). *Robot Manipulators*. MIT Press.

Sauer, T. (2000a). Numerische Mathematik I. Vorlesungsskript, Friedrich–Alexander–Universität Erlangen–Nürnberg, Justus–Liebig–Universität Gießen. `http://www.math.uni-giessen.de/tomas.sauer`.

Sauer, T. (2000b). Numerische Mathematik II. Vorlesungsskript, Friedrich–Alexander–Universität Erlangen–Nürnberg, Justus–Liebig–Universität Gießen. `http://www.math.uni-giessen.de/tomas.sauer`.

Sauer, T. (2001). Computeralgebra. Vorlesungsskript, Justus–Liebig–Universität Gießen. `http://www.math.uni-giessen.de/tomas.sauer`.

Sauer, T. (2002a). Approximationstheorie. Vorlesungsskript, Justus–Liebig–Universität Gießen. `http://www.math.uni-giessen.de/tomas.sauer`.

Sauer, T. (2002b). Optimierung. Vorlesungsskript, Justus–Liebig–Universität Gießen. `http://www.math.uni-giessen.de/tomas.sauer`.

Sauer, T. (2003). Digitale Signalverarbeitung. Vorlesungsskript, Justus–Liebig–Universität Gießen. `http://www.math.uni-giessen.de/tomas.sauer`.

Sauer, T. (2007). Splinekurven und –flächen in Theorie und Anwendung. Vorlesungsskript, Friedrich–Alexander–Universität Erlangen–Nürnberg, Justus–Liebig–Universität Gießen. `http://www.math.uni-giessen.de/tomas.sauer`.

Sauer, T. (2008). Integraltransformationen. Vorlesungsskript, Justus–Liebig–Universität Gießen. `http://www.math.uni-giessen.de/tomas.sauer`.

Sauer, T. (2011). Time–frequency analysis, wavelets and why things (can) go wrong. *Human Congnitive Neurophysiology*, **4**:38–64.

Sauer, T. (2012). Wavelets. Vorlesungsskript, Justus–Liebig–Universität Gießen.

Schoenberg, I. J. (1973). *Cardinal Spline Interpolation*, volume 12 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM.

Schüßler, H. W. (1992). *Digitale Signalverarbeitung*. Springer, 3. edition.

Shannon, C. E. (1949). Communications in the presence of noise. *Proc. of the IRE*, **37**:10–21.

Spellucci, P. (1993). *Numerische Verfahren der nichtlinearen Optimierung*. Internationale Schriftenreihe zu Numerischen Mathematik. Birkhäuser.

Steger, A. (2001). *Diskrete Strukturen 1. Kombinatorik – Graphentheorie – Algebra*. Springer.

Tolstov, G. P. (1962). *Fourier Series*. Prentice–Hall. Republished by Dover Publications, 1972.

Vetterli, M., Kovačević, J. (1995). *Wavelets and Subband Coding*. Prentice Hall.

Whittaker, J. (1935). *Interpolatory function theory*, volume 33 of *Cambridge Tracts in Math. and Math. Physics*.

Yosida, K. (1965). *Functional Analysis*. Grundlehren der mathematischen Wissenschaften. Springer–Verlag.

4 Index