# Bildverarbeitung

**Zusammenfassung**

Thomas Lang Manuel Pauli Sebastian Schweikl

13. Juli 2016

# Inhaltsverzeichnis


**1. Mathematische Grundlagen der Signalverarbeitung** **3**
- 1.1. Signalräume . . . 3
- 1.2. Fourier . . . 7
- 1.3. Der Abtastsatz . . . 21
- 1.4. Filter . . . 24
- 1.5. Filter für Bilder . . . 30
- 1.6. Die Schnelle Fourier-Transformation (FFT) . . . 38
  - 1.6.1. Die Diskrete Fourier-Transformation (DFT) . . . 38
  - 1.6.2. Diskret vs. Diskretisiert . . . 42
  - 1.6.3. Die Schnelle Fourier-Transformation . . . 42
  - 1.6.4. Fourier und Bilder . . . 44

**2. Transformationen** **46**
- 2.1. Die Hough-Transformation . . . 46

**Appendices** **50**
- A. Satz von Tonelli . . . 50
- B. Dichtheit von $L_p$-Räumen . . . 50
- C. Sinus Cardinalis . . . 50
  - C.1. Integrierbarkeit . . . 50
  - C.2. Flächeninhalt . . . 51
- D. Faltungseigenschaften . . . 52
  - D.1. Kommutativität . . . 52
  - D.2. Assoziativität . . . 53
  - D.3. Distributivität . . . 53

# 1. Mathematische Grundlagen der Signalverarbeitung

## 1.1. Signalräume

**Definition 1.1** (Torus) Als *Torus* bezeichnet man die Menge der Äquivalenzklassen definiert durch

$$\mathbb{T} := \mathbb{R}/2\pi\mathbb{Z},$$

gesprochen »$\mathbb{R}$ modulo $2\pi\mathbb{Z}$«.

**Bemerkung 1.2** (Torus) Einfach gesagt: Alle reellen Zahlen, die beim Teilen durch Vielfache von $2\pi$ den selben Rest lassen, sind äquivalent, und der Torus enthält alle möglichen Reste, die dabei auftreten können. Beispiel:

$$0/2\pi = 0 \text{ Rest } 0, \quad 2\pi/2\pi = 1 \text{ Rest } 0, \quad 4\pi/2\pi = 2 \text{ Rest } 0, \ldots$$

D.h., die Zahlen $0$, $2\pi$ und $4\pi$ sind zueinander äquivalent, da sie alle den Rest $0$ lassen. Man sagt, sie befinden sich in einer gemeinsamen *Äquivalenzklasse*. Da es unendlich viele Zahlen gibt, die beim Teilen durch $2\pi$ den Rest $0$ lassen, befinden sich unendlich viele Zahlen in dieser Äquivalenzklasse. Daher sucht man sich einen Stellvertreter (einen sog. *Repräsentanten*), um vernünftig arbeiten zu können. Es bietet sich die einfachste Zahl in der Äquivalenzklasse an, in diesem Fall die $0$, und man schreibt dann häufig $[0]$, wenn alle Zahlen gemeint sind, die zur $0$ äquivalent sind.

Nun wird dem einen oder anderen schon aufgefallen sein, dass der Torus auch unendlich viele Äquivalenzklassen besitzt (da es ja unendlich viele mögliche Reste beim Teilen gibt). Z.B. ist jede Zahl aus dem Intervall $[0, 2\pi)$ ein Repräsentant genau einer Äquivalenzklasse, und zu jeder Äquivalenzklasse kann man einen Repräsentanten in $[0, 2\pi)$ finden. Daher sagt man $\mathbb{T}$ ist *isomorph* zu $[0, 2\pi)$, in Zeichen

$$\mathbb{T} \simeq [0, 2\pi).$$

Wichtig: Das heißt *nicht*, dass $\mathbb{T}$ das Gleiche ist wie $[0, 2\pi)$! $[0, 2\pi)$ ist immer noch ein Intervall, und die Elemente aus $\mathbb{T}$ können zwar mit denen aus $[0, 2\pi)$ identifiziert werden, aber $\mathbb{T}$ hat *mehr Struktur* als $[0, 2\pi)$. Denn: Wenn ich z.B. die Zahl $\pi$ aus dem Torus mit sich selber addiere, dann bekomme ich $0$, denn $\pi + \pi = 2\pi$ und $2\pi/2\pi = 1$ mit Rest $0 \in \mathbb{T}$. Das Ergebnis ist wieder ein Element aus dem Torus! Wenn ich aber $\pi \in [0, 2\pi)$ betrachte, und die selbe Rechnung wiederhole, dann bekomme ich immer noch $2\pi$. Aber im Unterschied zu vorher gilt jetzt $2\pi \notin [0, 2\pi)$!

Übrigens gibt es unendlich viele Intervalle, die isomorph zu $\mathbb{T}$ sind, z.B.

$$[-\pi, \pi), (-2\pi, 0], [-2\pi, 0), [42.5\pi, 44.5\pi), \ldots$$

Jedes Intervall der Länge $2\pi$ ist isomorph zu $\mathbb{T}$! Aber man sucht sich natürlich nur die hübschen Intervalle raus, und das sind im wesentlichen eh nur $[0, 2\pi)$ und $[-\pi, \pi)$.

**Definition 1.3** (Funktionenräume)

1. Wir bezeichnen mit $L(\mathbb{R})$ die Gesamtheit aller reellwertigen Funktionen, d.h. die Menge aller Funktionen, die von $\mathbb{R}$ von $\mathbb{R}$ abbilden:

   $$L(\mathbb{R}) = \{f : \mathbb{R} \to \mathbb{R}\}.$$

   Analog ist $l(\mathbb{Z})$ die Menge aller reellwertigen Folgen, also Funktionen, die von $\mathbb{Z}$ nach $\mathbb{R}$ abbilden:

   $$l(\mathbb{Z}) = \{c : \mathbb{Z} \to \mathbb{R}\}.$$

   Wichtig: Die Indexmenge der Folge kommt aus $\mathbb{Z}$, das Bild einer Folge das aber selbstverständlich weiterhin eine Teilmenge von $\mathbb{R}$ sein. Wir könnten also auch schreiben:

   $$l(\mathbb{Z}) = \{(c_n)_{n\in\mathbb{Z}} \subseteq \mathbb{R}\}.$$

2. Die Menge aller summierbaren Funktionen $L_1(\mathbb{R})$ ist definiert durch

   $$L_1(\mathbb{R}) := \left\{ f \in L(\mathbb{R}) : \left\|f\right\|_1 := \int_{\mathbb{R}} |f(t)|\,\mathrm{d}t < \infty \right\}$$

   und die Menge aller summierbaren Folgen durch

   $$l_1(\mathbb{Z}) := \left\{ c \in l(\mathbb{Z}) : \left\|f\right\|_1 := \sum_{k\in\mathbb{Z}} |c(k)| < \infty \right\}$$

   Bildlich Dargestellt kann man sich dies so vorstellen:

   
   

   Abbildung 1.1.: Hinreichend schnelles Abklingverhalten von $L_1(\mathbb{R})$-Funktionen.

3. Analog wird die Menge der quadratsummierbaren Funktionen $L_2(\mathbb{R})$ definiert durch

   $$L_2(\mathbb{R}) := \left\{ f \in L(\mathbb{R}) : \left\|f\right\|_2 := \sqrt{\int_{\mathbb{R}} |f(t)|^2\,\mathrm{d}t} < \infty \right\}$$

und die Menge der quadratsummierbaren Folgen durch

$$l_2(\mathbb{Z}) := \left\{ c \in l(\mathbb{Z}) \colon \|c\|_2 := \sqrt{\sum_{k\in\mathbb{Z}} |c(k)|^2} < \infty. \right\}$$

$\|\bullet\|_2$ bezeichnet man auch als die »Energie-Norm«.
Bildlich Dargestellt kann man sich dies so vorstellen:


Abbildung 1.2.: Geometrische Interpretation als Mittelwert einer Fläche.

4. Die Menge aller beschränkten Funktionen und Folgen definiert durch

$$L_\infty(\mathbb{R}) := \left\{ f \in L(\mathbb{R}) \colon \left\|f\right\|_\infty := \sup_{t\in\mathbb{R}} |f(t)| < \infty \right\}$$

bzw.

$$l_\infty(\mathbb{Z}) := \left\{ c \in l(\mathbb{Z}) \colon \|c\|_\infty := \sup_{k\in\mathbb{Z}} |c(k)| < \infty \right\}.$$

5. Einer geht noch: Die Menge der Funktionen und Folgen mit *endlichem Träger*, geschrieben als $L_{00}(\mathbb{R})$ bzw. $l_{00}(\mathbb{Z})$. Was ist mit »endlichem Träger« gemeint? Das bedeutet, dass der Bereich, auf dem die Funktion bzw. Folge lebt, nicht unendlich groß sein darf. Formal: Es existiert ein $N \in \mathbb{N}$, sodass

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

**Bemerkung 1.4** ($L_1(\mathbb{R})$-Funktionen und $l_1(\mathbb{Z})$-Folgen) Wir betrachten nur Funktionen (bzw. Folgen, aber das werde ich jetzt nicht mehr dazu sagen), die »brav« sind. Damit ist gemeint, dass die Funktionen ein hinreichend schnelles Abklingverhalten gegen 0 besitzen müssen. Anschaulich gesprochen bewirkt der Betrag ja, dass wir einfach alles, was von der Funktion unterhalb der $x$-Achse liegt, nach oben »umklappen«, sodass es nun positiv ist. Und wenn wir jetzt darüber integrieren, darf nur was Endliches dabei herauskommen. Dies ist eine hinreichende Forderung, damit wir die Fourier-Transformation zu einer Funktion überhaupt vernünftig definieren können.

Wir stellen fest, dass in $l_1(\mathbb{Z})$ *nur* Nullfolgen (Folgen, deren Grenzwert 0 ist) zu finden sind, z.B. $(1/k^2)_{k\in\mathbb{Z}}$. Achtung! Die Folge $(1/k)_{k\in\mathbb{Z}}$ ist zwar auch eine Nullfolge, aber die Reihe dazu konvergiert nicht absolut (wir haben es hier ja mit der harmonischen Reihe zu tun), und ist daher ist die Folge auch nicht in $l_1(\mathbb{Z})$.

**Bemerkung 1.5** ($L_2(\mathbb{R})$-Funktionen und $l_2(\mathbb{Z})$-Folgen) Unterschied zu den »normalen« summierbaren Funktionen ist, dass hier die Zwei-Norm der Funktion kleiner als unendlich sein muss anstatt der Eins-Norm (daher kommt ja auch der Name »quadratsummierbar«). Leider gibt es hierfür keine so schöne geometrische Anschauung, da wir über ganz $\mathbb{R}$ integrieren, aber man kann versuchen, sich das Ganze so vorzustellen: Durch das Quadrieren des Betrags erhält man sozusagen eine Fläche, und durch das Integrieren erzeugen wir ein Volumen, welches dann so wie ein Schlauch an der $x$-Achse entlang wabert. Durch das Wurzelziehen brechen wir das Volumen wieder herunter auf eine Fläche. Und hier endet leider schon die Analogie. Auf dem Torus würde man jetzt noch durch $2\pi$ teilen (weil das die Länge eines Intervalls ist, welches isomorph zum Torus ist), und man könnte sich die Zwei-Norm vorstellen als Mittelwert der Fläche. Da wir aber über ganz $\mathbb{R}$ integrieren und wir schlecht durch $\infty$ teilen können, lass ma das hier bleiben und geben uns mit dem zufrieden, was wir schon haben.

Das führt uns unweigerlich zu einer wichtigen Frage: Warum sollte man also überhaupt den Raum $L_2(\mathbb{R})$ definieren wollen? Die Antwort ist: Weil Mathematiker es immer cool finden, irgendwelche abgefahrenen Konzepte zu verallgemeinern. Außerdem kann man in $L_2(\mathbb{R})$ ein Skalarprodukt von Funktionen definieren, mit dem sich recht schön rechnen lässt, was eben in $L_1(\mathbb{R})$ nicht geht.

Für die mathematisch Interessieren unter uns: $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ liegt dicht[1] in $L_2(\mathbb{R})$, und es gilt:

$$L_1(\mathbb{R}) \not\subset L_2(\mathbb{R}) \quad \text{und} \quad L_2(\mathbb{R}) \not\subset L_1(\mathbb{R}).$$

Das heißt aber nicht, dass der Schnitt der beiden Funktionenräume leer ist! Darin befinden sich nämlich ziemlich coole Funktionen, für die man wieder eine Fourier-Transformierte und sogar ein Skalarprodukt[2] definieren kann.

In $l_2(\mathbb{R})$ befinden sich übrigens auch nur Nullfolgen, da die Reihen zu den Folgen sonst wieder nicht konvergieren würden. Allerdings ist jetzt $(1/k)_{k\in\mathbb{Z}} \in l_2(\mathbb{R})$.

**Beispiel 1.6** (Beschränkte Folge) Betrachten wir als Beispiel einer Folge in $l_\infty(\mathbb{Z})$ die Folge $((-1)^n : n \in \mathbb{Z})$. Die Folge besitzt zwei Häufungspunkte $-1$ und 1, zwischen denen sie immer hin- und herspringt. Das Supremum dieser Folge ist natürlich 1, was kleiner als $\infty$ ist. Wäre diese Folge auch in $l_1(\mathbb{Z})$? Nein, wäre sie nicht, da sie ja nicht mal konvergiert.

**Beispiel 1.7** ($L_{00}(\mathbb{R})$-Funktion) Die Exponentialfunktion exp wird niemals 0, sie hat unendlichen Träger und deshalb keine $L_{00}(\mathbb{R})$-Funktion. Die Rechtecksfunktion $\chi_{[-1,1]}$ hingegen ist nur im Intervall $[-1,1]$ ungleich 0 und daher in $L_{00}(\mathbb{R})$.

**Definition 1.8** (Dirac-Puls) Der Dirac-Puls

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

ist eine lustige Funktion, die nur an der Stelle 0 gleich 1 ist und sonst überall 0 (siehe Abbildung 1.8).

[1] Siehe dazu auch B

[2] Achtung: Ein Skalarprodukt auf Lebesgue-Räumen ist nur auf $L_2$ definiert, für alle anderen Lebesgue-Räume existiert kein Skalarprodukt.

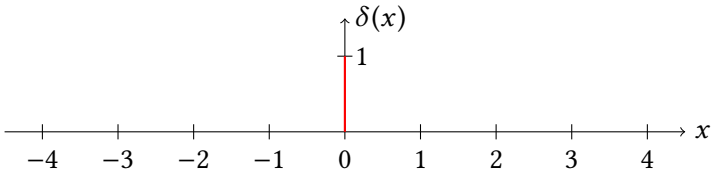


Abbildung 1.3.: Der Dirac-Puls $\delta$.

**Definition 1.9** (Abtastoperator) Der Abtastoperator $S_h : L(\mathbb{R}) \to l(\mathbb{R})$ mit Schrittweite $h$ ist für eine Funktion $f$ definiert als

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

**Bemerkung 1.10** (Abtastoperator) Anstatt die Funktion für alle reellen Zahlen zu betrachten, tasten wir die Funktion nur an abzählbar vielen Stellen im Abstand $h \in \mathbb{R}$ ab. Wir betrachten die Funktion also nur an den Stellen $0, h, -h, 2h, -2h, 3h, -3h \ldots$, was effektiv zu einer Diskretisierung des gegebene kontinuierlichen Signals führt. Diesen Prozess kann man auch als *Downsampling* bezeichnen.

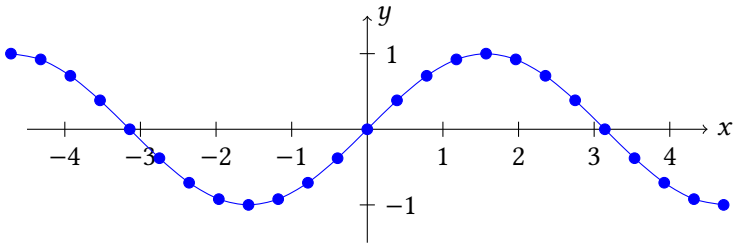


Abbildung 1.4.: Abtastung des Sinus an den hervorgehobenen Punkten.

## 1.2. Fourier

**Definition 1.11** (Fourier-Transformation) Für Funktionen $f \in L_1(\mathbb{R})$ definieren wir mit

$$\widehat{f} : \mathbb{R} \to \mathbb{C}, \qquad \widehat{f}(\xi) := f^\wedge(\xi) := \int_{\mathbb{R}} f(t) e^{-i\xi t} \, \mathrm{d}t, \quad \xi \in \mathbb{R}$$

die Fourier-Transformierte von $f$ und für Folgen $c \in l_1(\mathbb{Z})$

$$\widehat{c} : \mathbb{R} \to \mathbb{C}, \qquad \widehat{c}(\xi) := c^\wedge(\xi) := \sum_{k \in \mathbb{Z}} c(k) e^{-i\xi k}, \quad \xi \in \mathbb{R}$$

die Fourier-Transformierte von $c$.

**Bemerkung 1.12** (Fourier-Transformation)

- Was machen wir hier eigentlich? Schreiben wir einfach mal $\widehat{f}$ als

$$\begin{aligned}\int_{\mathbb{R}} f(t)e^{-i\xi t}\,\mathrm{d}t &= \int_{\mathbb{R}} f(t)(\cos(\xi t) - i\sin(\xi t))\,\mathrm{d}t \\ &= \int_{\mathbb{R}} f(t)\cos(\xi t) - i\int_{\mathbb{R}} f(t)\sin(\xi t)\,\mathrm{d}t\end{aligned}$$

  dann sehen wir, dass wir lediglich versuchen, $f$ auszudrücken als Kombination von Sinus- und Cosinus-Termen. Wir schauen einfach, wo $f$ und der sin bzw. cos eine große Ähnlichkeit zueinander haben (an der Stelle wird das Integral dann groß) und finden so heraus, welchen »Anteil« die Frequenz $\xi$ am Signal $f$ hat. Man kann es auch so sehen: Der Cosinus gibt immer den Gewichtungsfaktor der jeweiligen Frequenz vor, und der Sinus die Phasenverschiebung. Dass wir hier die doofe imaginäre Einheit $i$ mit drin haben, liegt halt einfach daran, dass wir die Identität

$$e^{ix} = \cos(x) + i\sin(x)$$

  ausgenutzt haben, um die Fourier-Transformation besonders elegant zu schreiben. Man hätte auch für Real- und Imaginärteil zwei gesonderte Fourier-Transformationen definieren können. Aber das soll uns hier nicht weiter stören. Außerdem kann man halt mit einer Exponentialfunktion schöner rechnen (z.B. ist die Stammfunktion der Exponentialfunktion wieder die Exponentialfunktion). Das ist eigentlich alles, was dahinter steckt. Will man die imaginäre Einheit ganz wegbekommen, geht man im diskreten Fall einfach über zur *Diskreten Cosinus-Transformation*.

- Wichtig: Mit der Fourier-Transformation finden wir zwar heraus, welche Frequenzen im Singal stecken, aber wir wissen nicht, an welcher Stelle bzw. zu welchem Zeitpunkt die entsprechende Frequenz auftritt! Wir haben keine Lokalität, da wir ja über ganz $\mathbb{R}$ integrieren. Das ist ein wichtiger Unterschied zur *Gabor-Transformation*, wo wir unser einer Fensterfunktion bedienen, um so Frequenzen besser lokalisieren zu können ☺.

- Warum brachen wir $L_1(\mathbb{R})$-Funktionen? Wie vorher schon erwähnt, ist das eine hinreichende Bedingung, dass $\widehat{f}$ überhaupt existiert:

$$\widehat{f} \le |\widehat{f}| = \left|\int_{\mathbb{R}} f(t)e^{-i\xi t}\,\mathrm{d}t\right| \le \int_{\mathbb{R}} |f(t)|\,\underbrace{|e^{-i\xi t}|}_{=1}\,\mathrm{d}t = \int_{\mathbb{R}} |f(t)|\,\mathrm{d}t < \infty.$$

  Das Argument lässt sich analog auf $l_1(\mathbb{Z})$-Folgen übertragen.

- Der Fourier-Transformation einer Folge $c$ sollte an dieser Stelle noch besondere Aufmerksamkeit gewidmet werden. Sie überführt $c$ zum einen in eine *Funktion* $\widehat{c}$ (und nicht wieder in eine Folge) und ist zum anderen auch noch $2\pi$-periodisch:

$$\widehat{c}(\xi + 2\pi) = \sum_{k\in\mathbb{Z}} c(k)e^{-i(\xi+2\pi)k} = \sum_{k\in\mathbb{Z}} c(k)e^{-i\xi k}\underbrace{e^{-i2\pi k}}_{=1} = \sum_{k\in\mathbb{Z}} c(k)e^{-i\xi k} = \widehat{c}(\xi).$$

  Die Fourier-Transformation zu einer Funktion ist *nicht* $2\pi$-periodisch! Diese Tatsache bewirkt z.B., dass die Inverse Fourier-Transformation einer Folge eine völlig andere Struktur besitzt als die einer Funktion, wie wir später sehen werden.

**Definition 1.13** (Translations- und Skalierungsoperator)

1. Der Translationsoperator $\tau_y$ mit $y \in \mathbb{R}$ angewendet auf eine Funktion $f$ ist definiert als

$$\tau_y f := f(\bullet + y).$$

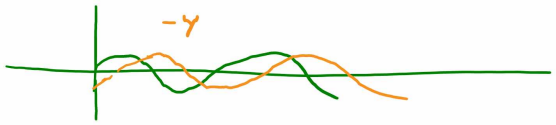


Abbildung 1.5.: Verschiebung der grün gezeichneten Funktion um $y$ Einheiten nach rechts führt zur orange dargestellten Funktion.

2. Der Skalierungsoperator $\sigma_h$ mit $h \in \mathbb{R} \setminus \{0\}$ angewendet auf eine Funktion $f$ ist definiert als

$$\sigma_h f := f(h \cdot \bullet).$$

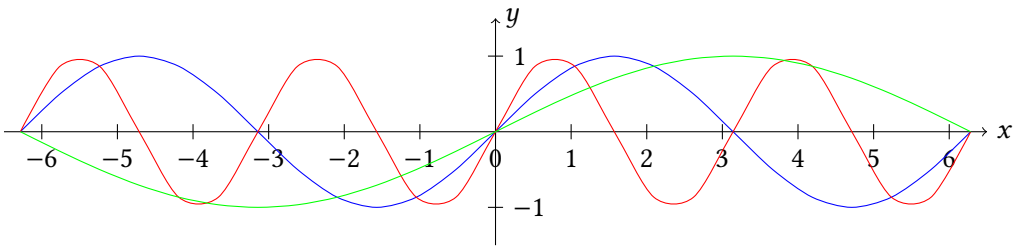


Abbildung 1.6.: Skalierung der Sinus-Funktion (blau) mit dem Faktor 2 führt zu doppelt so schneller Schwingung (roter Graph). Skalierung mit dem Faktor 0.5 bewirkt halb so schnelle Schwingung (grüner Graph).

**Bemerkung 1.14** (Translations- und Skalierungsoperator)

1. der Translationsoperator verschiebt eine Funktion auf der $x$-Achse um $y$ Einheiten nach links oder rechts:

| Wert von $y$ | Effekt auf $f$ |
|---|---|
| $y > 0$ | Verschiebung nach *links* |
| $y = 0$ | Keine Verschiebung |
| $y < 0$ | Verschiebung nach *rechts* |

2. Der Skalierungsoperator streckt oder staucht eine Funktion um den Faktor $h$ und kann sie sogar an der $y$-Achse spiegeln:

| Wert von $h$ | Effekt auf $f$ |
|---|---|
| $h > 1$ | Stauchung |
| $h = 1$ | Kein Effekt |
| $0 < h < 1$ | Streckung |
| $h = 0$ | Um Gottes Willen! Das ist pfui-gack. |
| $-1 < h < 0$ | Streckung und Spiegelung an der $y$-Achse |
| $h = -1$ | Nur Spiegelung an der $y$-Achse |
| $h < -1$ | Stauchung und Spiegelung an der $y$-Achse |

3. Die Definition des Skalierungsoperators ist recht ähnlich zur der des Abtastoperators mit Schrittweite $h$. An dieser Stelle sollte betont werden, dass beide Operatoren nicht miteinander zu verwechseln sind! Der Abtastoperator liefert zu einem kontinuierlichen Signal $f$ ein diskretes Signal $S_h\, f = (f(h \cdot \bullet))_{h\in\mathbb{Z}}$. Das $h$ ist hier variabel. Der Skalierungsoperator überführt ein kontinuierliches Signal $f$ wieder in ein kontinuierliches Signal $\sigma_h\, f = f(h \cdot \bullet)$, welches eben um den Faktor $h$ gestreckt bzw. gestaucht wurde. $h$ ist hier ein fester Wert.

4. Translation und Skalierung sind invertierbar, d.h. man kann ihre Auswirkungen wieder rückgängig machen:

$$\tau_y\, (\tau_{-y}\, f) = \tau_{-y}\, (\tau_y\, f) = f \quad \text{und} \quad \sigma_h\, (\sigma_{1/h}\, f) = \sigma_{1/h}\, (\sigma_h\, f) = f.$$

**Definition 1.15** (Faltung) Seien $f, g \in L(\mathbb{R})$ und $c, d \in l(\mathbb{Z})$. Dann ist die Faltung zweier Funktionen definiert als

$$f * g \coloneqq \int_{\mathbb{R}} f(\bullet - t) \cdot g(t)\, \mathrm{d}t \in L(\mathbb{R})$$

und die Faltung zweier Folgen als

$$c * d \coloneqq \sum_{k\in\mathbb{Z}} c(\bullet - k) \cdot d(k) \in l(\mathbb{Z}).$$

Die Faltung einer Funktion $f$ mit einer Folge $c$ ist definiert durch

$$c * f \coloneqq f * c \coloneqq \sum_{k\in\mathbb{Z}} f(\bullet - k) \cdot d(k) \in L(\mathbb{R}).$$

**Bemerkung 1.16** (Eigenschaften der Faltung) Aus der Linearität des Integrals und den Gruppenoperationen auf $\mathbb{R}$ lassen sich folgende Eigenschaften der Faltung für Funktionen oder Folgen $f, g, h$ herleiten:

- Kommutativität: $f * g = g * f$

- Assoziativität: $(f * g) * h = f * (g * h)$
- Distributivität: $f * (g + h) = f * g + f * h$
- Skalare Multiplikation: $a \cdot (f * g) = (a \cdot f) * g = f * (a \cdot g)$, wobei $a \in \mathbb{R}$ eine beliebige Konstante ist.

**Bemerkung 1.17** (Interpretation der Faltung) Die Faltung kann aufgefasst werden als Produkt zweier Funktionen oder Folgen, welches wieder eine Funktion bzw. Folge liefert. Wie kann man sich die Faltung geometrisch vorstellen? Betrachten wir als Beispiel zwei Funktionen $f$ und $g$ und die Faltung $f * g$. Was dabei passiert, ist Folgendes: Zunächst wird $f$ durch $f(\bullet - t)$ *vertikal* gespiegelt. Wir halten anschließend die Funktion $g$ fest und lassen $f$ einmal komplett von ganz links nach ganz rechts über die $x$-Achse wandern. Dort, wo sich $f$ und $g$ überlagern und eine große »Gemeinsamkeit« miteinander haben, wird auch das Integral groß. Dort, wo beide Funktionen keine große Gemeinsamkeit miteinander haben, wird das Integral klein. Die Faltung ist also eine Methode, um feststellen zu können, wie *lokal* ähnlich (nicht global!) sich zwei Funktionen sind.

Je nachdem, welche Funktion man für $g$ wählt, lassen sich mit der Faltung unterschiedliche interessante andere Funktionen erzeugen. Wikipedia meint, dass eine Faltung $f * g$ einen »gewichteten Mittelwert« von $f$ darstellt, wobei die Gewichtung durch $g$ vorgegeben ist. Diese Argumentation versteht man eigentlich erst, wenn man sich *zyklische Faltungen* anschaut, wo nicht über ganz $\mathbb{R}$ integriert wird, sondern über ein Kompaktum. Dann wird das Integral nämlich noch durch die Länge des Kompaktums dividiert, sodass man tatsächlich eine Art Durschnitt hat. Und hey, das kommt uns doch jetzt irgendwie von der Definition der $L_2(\mathbb{R})$- Funktionen bekannt vor!

**Beispiel 1.18** (Faltung) Falten wir doch einmal die Rechtecksfunktion $\chi_{[-1,1]}$ mit sich selbst. Man definiert

$$\chi_{[-1,1]}(x) = \begin{cases} 1, & x \in [-1,1] \\ 0, & \text{sonst.} \end{cases}$$

Dann ist

$$\begin{aligned}
(\chi_{[-1,1]} * \chi_{[-1,1]})(x) &= \int_{\mathbb{R}} \chi_{[-1,1]}(x-t) \cdot \chi_{[-1,1]}(t)\,\mathrm{d}t = \int_{\mathbb{R}} \chi_{[x-1,x+1]}(t) \cdot \chi_{[-1,1]}(t)\,\mathrm{d}t \\
&= \int_{\mathbb{R}} \chi_{[x-1,x+1]\cap[-1,1]}(t)\,\mathrm{d}t = \int_{[x-1,x+1]\cap[-1,1]} 1\,\mathrm{d}t \\
&= \begin{cases} \int_{-1}^{x+1} 1\,\mathrm{d}t = 2 + x, & -2 \le x \le 0, \\ \int_{x-1}^{1} 1\,\mathrm{d}t = 2 - x, & 0 < x \le 2, \\ 0, & \text{sonst}, \end{cases} \\
&=: \Delta_{[-2,2]}(x).
\end{aligned}$$

Man erhält also die Dreiecksfunktion auf dem Intervall $[-2,2]$. Was bedeutet das? Naja, die $\Delta_{[-2,2]}(x)$ an der Stelle $x$ gibt genau die Fläche an, die zwischen den beiden Rechtecksfunktionen gerade eingeschlossen wird, wenn man eine Rechtecksfunktion um $x$ Einheiten verschiebt. Im

Fall von $x = 0$ überlappen sich beide Rechtecksfunktionen ganz genau, und deren Flächeninhalt ist 2. Dies ist genau der Wert von $\Delta_{[-2,2]}(2)$! Verschiebt man eine Rechtecksfunktion um 1 Einheit nach links oder rechts, dann wird nur noch die Hälfte der Fläche zwischen beiden Rechtecksfunktionen eingeschlossen, also Flächeninhalt 1. Und genau das kommt bei $\Delta_{[-2,2]}(x)$ heraus, wenn man $x = 1$ oder $x = -1$ einsetzt.

Die geometrische Interpretation einer Faltung ist also sehr vielfältig und hängt sehr stark von den beiden Funktionen ab, die miteinander gefaltet werden, siehe Abbildung 1.7.

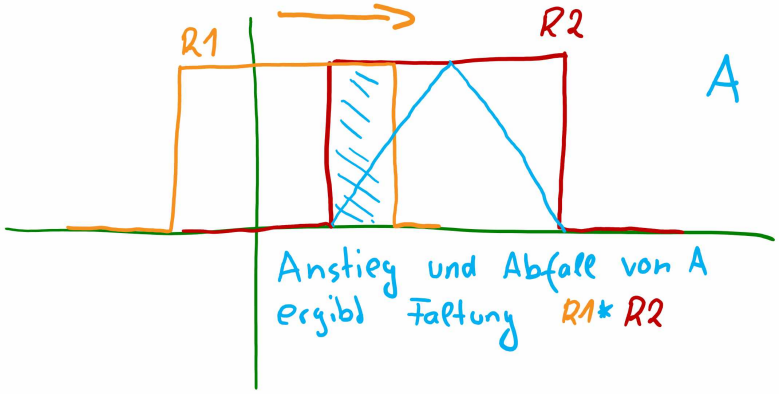


Abbildung 1.7.: Faltung der Rechtecksfunktion (orange und rot) mit sich selbst ergibt die Dreiecksfunktion (blau).

Nun kommen wir zu einem sehr wichtigen Teil der Vorlesung, nämlich zu den Eigenschaften der Fourier-Transformation. Insbesondere wollen wir uns mit deren graphischer Deutung beschäftigen. Faltungen werden in diesem Kontext auch eine sehr wichtige Rolle spielen, wenn wir die Fourier-Transformierte besonders schnell berechnen wollen.

**Bemerkung 1.19** (Eigenschaften der Fourier-Transformation und deren graphische Deutung)

**Linearität** Die Fourier-Transformierte ist linear. Dies folgt sofort aus der Linearität des Integrals. Seien also $f, g \in L_1(\mathbb{R})$ und $a, b \in \mathbb{R}$. Dann gilt:

$$(a \cdot f + b \cdot g)^\wedge = a \cdot \widehat{f} + b \cdot \widehat{g}.$$

Bildliche Vorstellung: Multiplizieren wir eine Funktion $f$ mit einer Konstanten $a$, so ändert sich lediglich die Amplitude von $f$. Die Frequenzen in der Funktion bleiben gleich, es kommen keine neuen Frequenzen hinzu und es fallen keine weg. Der »Anteil« der bereits vorhandenen Frequenzen wird nur anders gewichtet, nämlich mit dem Faktor $a$ versehen. Deshalb gilt $(a \cdot f)^\wedge = a \cdot \widehat{f}$ (siehe hierfür auch Abbildung 1.8, linker Teil).

Was hat es aber mit der Addition zweier Funktionen auf sich? Nun, wenn wir zwei Funktionen addieren, so sollten sich auch die Anteile der Frequenzen addieren. Dies spiegelt sich genau in der Identität $(f + g)^\wedge = \widehat{f} + \widehat{g}$ wider (Abbildung 1.8, rechter Teil).

Was passiert, wenn wir $f = g$ setzen? Dann sollte der Anteil jeder Frequenz doppelt so groß sein wie vorher. Und in der Tat: Wir können jetzt entweder sagen $(f + f)^\wedge = \widehat{f} + \widehat{f} = 2\widehat{f}$ oder $(f + f)^\wedge = (2f)^\wedge = 2\widehat{f}$. In beiden Fällen kommt das gleiche raus.

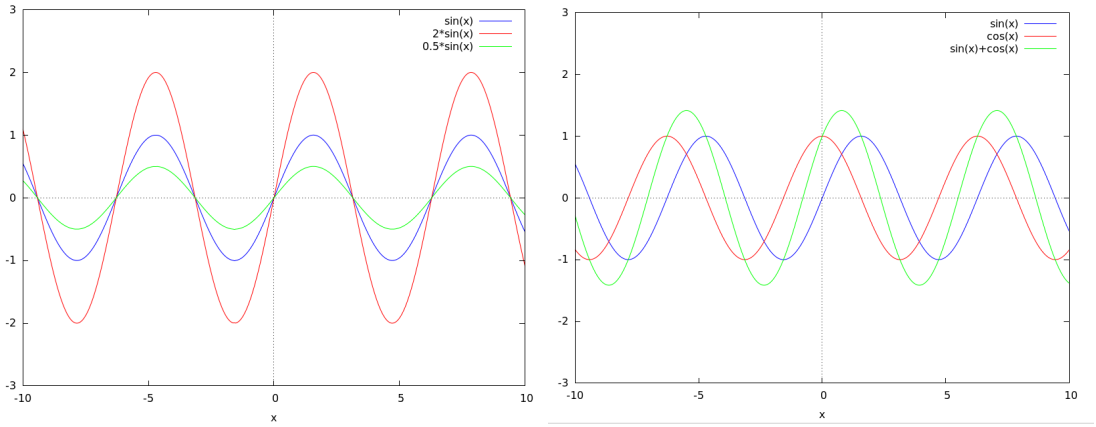


Abbildung 1.8.: Links: Änderung der Amplitude der Sinus-Funktion bei gleichbleibender Frequenz. Rechts: Bei Addition der Sinus- und Cosinus-Funktion addieren sich auch die Frequenzen.

**Translation (Zeitverschiebung)** Für $f \in L_1(\mathbb{R})$ und $y \in \mathbb{R}$ gilt:

$$(\tau_y f)^\wedge(\xi) = \Big(f(\bullet + y)\Big)^\wedge(\xi) = e^{iy\xi}\widehat{f}(\xi), \quad \xi \in \mathbb{R}.$$

Wie bereits oben erwähnt verschiebt eine Translation eine Funktion um den Wert $y$. Verschiebt man nun die ursprüngliche Funktion des Signal, sollte sich an den Anteilen der Frequenzen $\xi$ nichts ändern. Wo kommt aber nun dieser seltsame Faktor $e^{iy\xi}$ her? Dazu müssen wir uns wieder in Erinnerung rufen, dass sich die Fourier-Transformation auch so schreiben lässt: $\int_{\mathbb{R}} f(t)\cos(\xi t)\,\mathrm{d}t - i\int_{\mathbb{R}} f(t)\sin(\xi t)\,\mathrm{d}t$.
→ Es gibt einen Imaginärteil und einen Realteil. Durch die Translation ändert sich nur die Aufteilung zwischen diesen zwei Teilen. Der Gesamtanteil bleibt gleich. Diese Umverteilung geschieht durch den Vorfaktor $e^{iy\xi}$. (Dies bezeichnet man auch als *Frequenz-Modulation*).

Betrachten wir $\widehat{f}$ als Vektor in der komplexen Zahlenebene, so führt der Faktor $e^{iy\xi}$ lediglich dazu, dass die der Vektor um den Winkel $y \cdot \xi$ im Uhrzeigersinn gedreht wird. D.h. an den Anteilen der Frequenzen ändert sich im Absolutbetrag nichts. Schließlich ist ja auch $|e^{iy\xi}| = 1$.

Als veranschaulichendes Beispiel diene hier die Rechtecks-Funktion $\chi_{[-1,1]}$ mit ihrer Fourier-Transformierten $2\sin(x)/x$. Wird die Rechtecks-Funktion um eine Einheit nach links verschoben, so wird die Fourier-Transformierte mit dem Vorfaktor $e^{i\xi}$ versehen, also

$$(\tau_1\chi_{[-1,1]})^\wedge(\xi) = \Big(\chi_{[-1,1]}(\bullet + 1)\Big)^\wedge(\xi) = \widehat{\chi}_{[-2,0]}(\xi) = e^{i\xi} \cdot \frac{2\sin(\xi)}{\xi}.$$

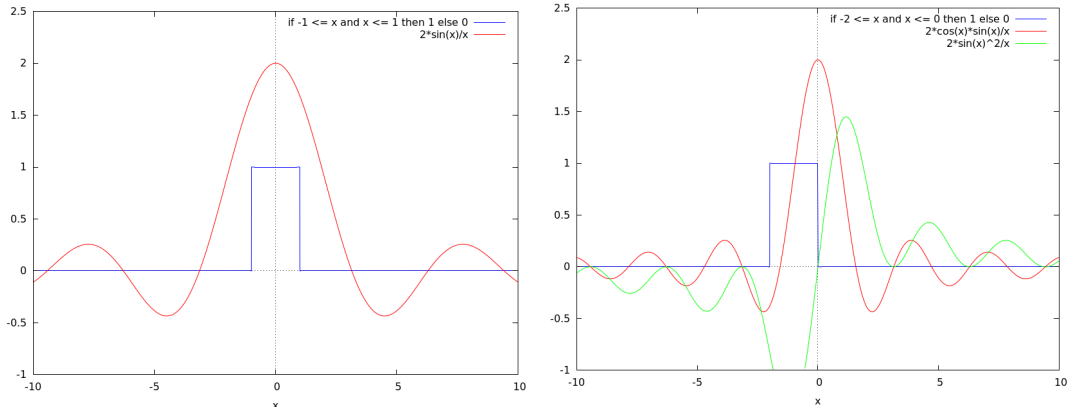


Abbildung 1.9.: Links: Rechtecksfunktion und deren Fourier-Transformierte. Rechts: Die um 1 nach links verschobene Rechtecksfunktion und deren Fourier-Transformierte mit Real- und Imaginärteil separat geplottet (Realteil rot, Imaginärteil grün). Gut sichtbar ist die Phasenverschiebung zwischen Sinus und Cosinus.

Abbildung 1.9 zeigt die Rechtecksfunktion und die verschobene Rechtecksfunktion, sowie deren Fourier-Transformierte. Wir stellen zunächst fest, dass die Rechtecksfunktion eine vollständig reelle Fourier-Transformierte besitzt und die verschobene Rechtecksfunktion eine reell- und komplexwertige Fourier-Transformierte. Dies zeigt ganz deutlich, dass sich die Aufteilung der Frequenzen zwischen Sinus und Cosinus verschoben hat. Die interessante Frage ist aber nun: Hat sich an den Anteilen der Frequenzen im Ganzen etwas geändert? Die Antwort liefert Abbildung 1.10. Im Absolutbetrag sind nämlich beide Fourier-Transformationen identisch. D.h. die Frequenzanteile sind im Ganzen gleich geblieben. Dies hätte man auch schon so sehen können:

$$\left|(\tau_1 \chi_{[-1,1]})^\wedge(\xi)\right| = \left|e^{i\xi} \cdot \widehat{\chi}_{[-1,1]}(\xi)\right| = \left|e^{i\xi}\right| \cdot \left|\widehat{\chi}_{[-1,1]}(\xi)\right| = 1 \cdot \left|\widehat{\chi}_{[-1,1]}(\xi)\right| = \left|\widehat{\chi}_{[-1,1]}(\xi)\right|.$$

Also ganz kurz und knapp: Verschiebung im Zeit-/Ortsbereich führt zu Modulation im Frequenzbereich.

Übrigens: Der Realteil der Fourier-Transformierten ist immer achsensymmetrisch zur $y$-Achse, weil der Cosinus achsensymmetrisch zur $y$-Achse ist, und der Imaginärteil ist punktsymmetrisch um den Ursprung, weil der Sinus punktsymmetrisch um den Ursprung ist.

**(Zeit-)Skalierung** Sei $f \in L_1(\mathbb{R})$ und $h \neq 0$. Dann gilt

$$(\sigma_h f)^\wedge(\xi) = f(h \cdot \bullet)^\wedge(\xi) = \frac{1}{|h|} \widehat{f}\left(\frac{\xi}{h}\right), \quad \xi \in \mathbb{R}.$$

Durch den Skalierungsoperator wird eine Funktion im Zeit-/Ortsbereich in Richtung der $x$-Achse gestaucht oder gestreckt (anders ausgedrückt: Die Periodenlänge ändert sich). Damit ändern sich auch die im Signal enthaltenen Frequenzen. Beispiel anhand des Sinus:

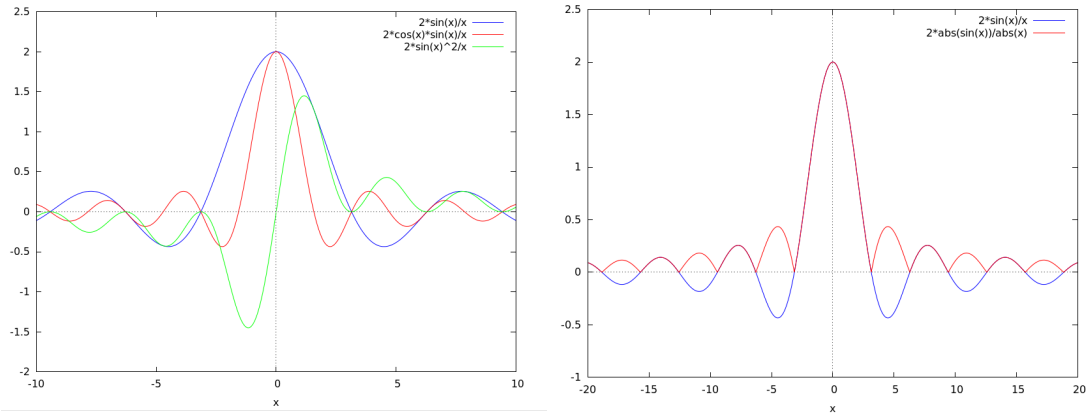


Abbildung 1.10.: Links: Fourier-Transformierte der Rechtecksfunktion (blau) und der verschobenen Rechtecksfunktion (Realteil rot, Imaginärteil grün). Rechts: Fourier-Transformation der Rechtecksfunktion (blau) und der verschobenen Rechtecksfunktion im Absolutbetrag (rot).

Wird die Periodenlänge halbiert, so verdoppelt sich doch die Frequenz, denn der Sinus schwingt jetzt doppelt so schnell. In diesem Kontext ist die Frequenz der Reziprokwert der Periodenlänge.

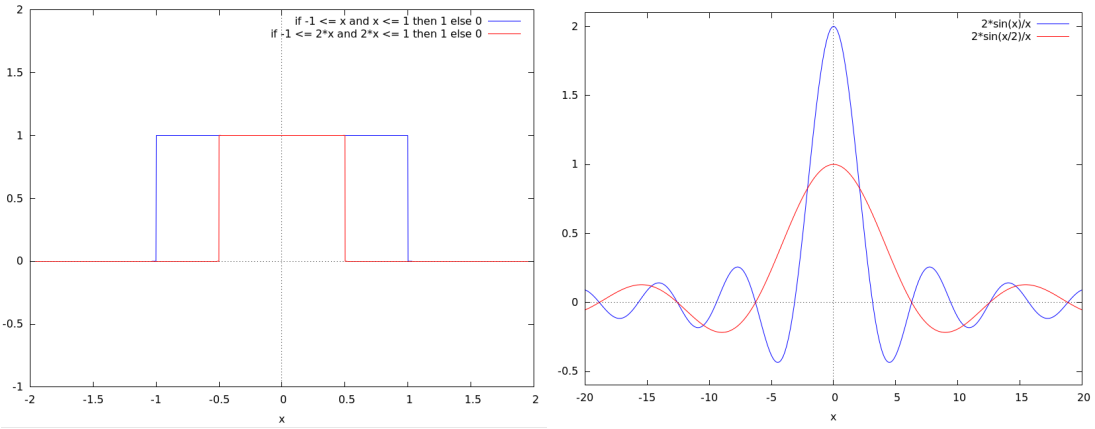


Abbildung 1.11.: Links: Rechtecksfunktion (blau) und gestauchte Rechtecksfunktion (rot). Rechts: Fourier-Transformationen zu den Rechtecksfunktionen (in gleicher Farbe).

Können wir diese Analogie auch auf die Fourier-Transformation übertragen? Betrachten wir ein weiteres Mal die Rechtecksfunktion. Skalieren wir diese mit einem Faktor von $h = 2$ und bilden davon die Fourier-Transformation, so erhalten wir:

$$(\sigma_2\ \chi_{[-1,1]})^\wedge(\xi) = \frac{1}{2}\widehat{\chi}_{[-1,1]}\left(\frac{\xi}{2}\right) = \frac{2}{\xi}\sin\left(\frac{\xi}{2}\right).$$

Abbildung 1.11 zeigt die eben angesprochenen Funktionen. Das $\xi/2$ im Argument von $(\sigma_2\ \chi_{[-1,1]})^\wedge$ führt dazu, dass die Fourier-Transformation auseinander gezogen wird. Durch den Vorfaktor $2/\xi$ verringert sich zudem die Amplitude. Dies macht auch Sinn, denn durch das Zusammenstauchen einer Funktion im Ortsbereich schwingt sie ja schneller, sie wird hochfrequenter. Das entspricht genau dem Auseinanderziehen der Fourier-Transformation. Es wird mehr Anteil in den hochfrequenten Bereich verlagert. Gleichzeitig wird der Anteil der niedrigen Frequenzen (die sich im Frequenzbereich um den Ursprung konzentrieren) geringer, was dem Plattdrücken der Fourier-Transformierten entspricht.

Man stellt fest: Für große Werte von $h$ konzentriert sich die Funktion im Zeitbereich stark um den Ursprung, sie wird gestaucht und hochfrequenter. Im Frequenzbereich führt dies dazu, dass die Fourier-Transformation gestreckt (auseinander gezogen) und platt gedrückt wird. Umgekehrt führen kleine Werte von $h$ dazu, dass die Funktion im Ortsbereich gestreckt wird, also niederfrequenter wird. Die Fourier-Transformation wird um den Ursprung zusammengedrückt und schlägt stärker aus.

**Faltung** Für $f, g \in L_1(\mathbb{R})$ bzw. $c, d \in l_1(\mathbb{Z})$ sind $f * g \in L_1(\mathbb{R})$, $c * d \in l_1(\mathbb{Z})$ und $f * c \in L_1(\mathbb{R})$ und es gilt:

$$(f * g)^\wedge(\xi) = \widehat{f}(\xi)\widehat{g}(\xi), \quad (c * d)^\wedge(\xi) = \widehat{c}(\xi)\widehat{d}(\xi) \quad \text{und} \quad (f * c)^\wedge(\xi) = \widehat{f}(\xi)\widehat{c}(\xi).$$

Eine Faltung im Zeitbereich gibt ja so etwas wie die Übereinstimmung zwischen den beiden Funktionen an, und die Multiplikation im Frequenzbereich kann man sich nun vorstellen als Übereinstimmung der Frequenzen der beiden Funktionen.

*Eine Faltung im Zeitbereich wird also überführt in eine Multiplikation im Frequenzbereich.* Dies ist eine extrem wichtige Eigenschaft der Fourier-Transformation. Dies hat im Wesentlichen zwei Gründe:

1. Die Berechnung einer Faltung zweier Funktionen ist normalerweise viel aufwendiger als die beiden Funktionen einfach punktweise zu multiplizieren (es sind hier weniger Operationen auszuführen).
2. Außerdem weist das Produkt eine höhere numerische Stabilität auf als die Faltung.

Diese Eigenschaften kann man sich zunutze machen, um Filter effizient zu implementieren, und dabei ist von der Schnellen Fourier-Transformation (FFT) noch gar nicht die Rede.

Weil es so schön ist, hier der Faltungsbeweis für den Fall, dass zwei Funktionen $f, g \in L_1(\mathbb{R})$ gegeben sind:

$$\begin{aligned}
(f * g)^\wedge(\xi) &= \int_{\mathbb{R}} (f * g)(x) e^{-i\xi x}\,\mathrm{d}x && \text{(Def. Fourier-Transf.)} \\
&= \int_{\mathbb{R}} \left( \int_{\mathbb{R}} f(t) g(x-t)\,\mathrm{d}t \right) e^{-i\xi x}\,\mathrm{d}x && \text{(Def. Faltung)} \\
&= \int_{\mathbb{R}} \int_{\mathbb{R}} f(t) g(x-t) e^{-i\xi x}\,\mathrm{d}x\,\mathrm{d}t && \text{(Tonelli)} \\
&= \int_{\mathbb{R}} \int_{\mathbb{R}} f(t) g(s) e^{-i\xi(t+s)}\,\mathrm{d}s\,\mathrm{d}t && \text{(Subst. } s := x - t,\ \mathrm{d}s = \mathrm{d}x\text{)}
\end{aligned}$$

$$= \left(\int_{\mathbb{R}} f(t)e^{-i\xi t}\,\mathrm{d}t\right)\left(\int_{\mathbb{R}} g(s)e^{-i\xi s}\,\mathrm{d}s\right) \qquad \text{(Arithmetik \& Tonelli)}$$
$$= \widehat{f}(\xi)\cdot\widehat{g}(\xi). \qquad \text{(Def. Fourier-Transf.)}$$

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

$$\left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)^{\wedge}(\xi) = i\xi\widehat{f}(\xi), \quad \xi \in \mathbb{R}.$$

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

$$\frac{\mathrm{d}\widehat{f}(\xi)}{\mathrm{d}\xi} = (-ixf)^{\wedge}(\xi).$$

Diese Eigenschaften sind vor allem nützlich fürs Rechnen.

**Inverse Fourier-Transformation** Sind $f, \widehat{f} \in L_1(\mathbb{R})$, dann ist

$$f(x) = (\widehat{f})^{\vee}(x) \coloneqq \frac{1}{2\pi}\int_{\mathbb{R}} \widehat{f}(\vartheta)\, e^{ix\vartheta}\,\mathrm{d}\vartheta, \qquad x \in \mathbb{R}.$$

Die Operation $f \mapsto f^{\vee} \coloneqq \frac{1}{2\pi} f^{\wedge}(-\bullet)$ bezeichnet man als inverse Fourier-Transformation.

Sind $c \in l_1(\mathbb{R})$ und $\widehat{c} \in L_1(\mathbb{T})$, dann ist

$$c(k) = (\widehat{c})^{\vee}(k) \coloneqq \frac{1}{2\pi}\int_{\mathbb{T}} \widehat{c}(\xi)e^{ik\xi}\,\mathrm{d}\xi, \qquad k \in \mathbb{Z}.$$

Die Operation $c \mapsto c^{\vee}$ bezeichnet man als inverse Fourier-Transformation.

Dass die Nützlichkeit einer Transformation stark davon abhängt, ob bzw. unter welchen Voraussetzungen diese wieder rückgängig gemacht werden kann, braucht an dieser Stelle nicht weiter betont zu werden. In unserem Fall müssen beide Funktionen $f, \widehat{f} \in L_1(\mathbb{R})$ sein. Setzen wir beispielsweise $f = \chi_{[-1,1]}$, so erhalten wir i.W. die sinc-Funktion als Fourier-Transformierte. Diese ist aber keine $L_1(\mathbb{R})$-Funktion mehr und deshalb ist die Fourier-Transformation hier insbesondere nicht invertierbar! Aber das kommt Gott sei Dank nicht so häufig vor.

Ist $c \in l_1(\mathbb{Z})$, dann gilt auch $\widehat{c} \in L_1(\mathbb{T})$, denn

$$\left\|\widehat{c}\right\|_{\mathbb{T},1} = \int_{\mathbb{T}} |\widehat{c}(t)|\,\mathrm{d}t = \int_{\mathbb{T}} \left|\sum_{k\in\mathbb{Z}} c(k)e^{-ikt}\right| \mathrm{d}t$$
$$\leq \int_{\mathbb{T}} \sum_{k\in\mathbb{Z}} |c(k)|\,\underbrace{|e^{-ikt}|}_{=1}\,\mathrm{d}t = \int_{\mathbb{T}} \underbrace{\sum_{k\in\mathbb{Z}} |c(k)|}_{=\|c\|_1}\,\mathrm{d}t$$
$$= \int_{\mathbb{T}} \|c\|_1\,\mathrm{d}t = \underbrace{\|c\|_1}_{<\infty}\underbrace{\int_{\mathbb{T}} 1\,\mathrm{d}t}_{<\infty}$$

$$< \infty.$$

Mit anderen Worten, die Fourier-Transformation für Folgen ist im Gegensatz zu der für Funktionen *immer* invertierbar.

Dass die Formel für die Inverse Fourier-Transformation den Vorfaktor $1/2\pi$ beinhaltet, liegt daran, dass wir die Formel für die Fourier-Transformation in Definition 1.11 mit keinen Vorfaktor versehen haben. Dies ist aber nur Normierung. Wir hätten auch in Definition 1.11 mit $1/\sqrt{2\pi}$ normieren können, dann wäre dieser Faktor jetzt auch bei der Inversen Fourier-Transformation aufgetaucht und wir hätten eine schöne Isometrie.

**Beispiel 1.20** (Inverse Fourier-Transformation) Wir berechnen die Fourier-Transformation der Gauß-Funktion

$$f(x) = \exp(-x^2), \quad x \in \mathbb{R}.$$

Der Nachweis $f \in L_1(\mathbb{R})$ sei dabei dem Leser zur Übung überlassen. Es sei aber soviel gesagt, dass

$$\int_{\mathbb{R}} f(x)\,\mathrm{d}x = \sqrt{\pi}.$$

Es gilt

$$\begin{aligned}\widehat{f}(\xi) &= \int_{\mathbb{R}} f(x)\exp(-i\xi x)\,\mathrm{d}x = \int_{\mathbb{R}} \exp(-x^2)\exp(-i\xi x)\,\mathrm{d}x = \int_{\mathbb{R}} \exp(-(x^2+i\xi x))\,\mathrm{d}x \\ &= \int_{\mathbb{R}} \exp(-(x^2+i\xi x-\xi^2/4)-\xi^2/4)\,\mathrm{d}x = \int_{\mathbb{R}} \exp(-(x+i\xi/2)^2)\exp(-\xi^2/4)\,\mathrm{d}x.\end{aligned}$$

Die Substitution $u := x + i\xi/2$, $\mathrm{d}u = \mathrm{d}x$ liefert weiter

$$\exp(-\xi^2/4)\underbrace{\int_{\mathbb{R}} \exp(-u^2)\,\mathrm{d}u}_{=\sqrt{\pi}} = \sqrt{\pi}\exp(-\xi^2/4).$$

Wir haben also gezeigt, dass $\widehat{f}(\xi) = \sqrt{\pi}\exp(-\xi^2/4)$. Die Fourier-Transformierte der Gauß-Funktion ist also i.W. bis auf Normierung die Gauß-Funktion selbst. Man kann sich leicht überlegen, dass auch $\widehat{f}(\xi) \in L_1(\mathbb{R})$. Damit können wir die Inverse Fourier-Transformation berechnen. Dabei sollte wieder die ursprüngliche Funktion $f$ herauskommen. Also:

$$\begin{aligned}\left(\widehat{f}\right)^{\vee}(x) &= \frac{1}{2\pi}\int_{\mathbb{R}} \widehat{f}(\xi)\exp(i\xi x)\,\mathrm{d}\xi = \frac{1}{2\pi}\int_{\mathbb{R}} \sqrt{\pi}\exp(-\xi^2/4)\exp(i\xi x)\,\mathrm{d}\xi \\ &= \frac{1}{2\sqrt{\pi}}\int_{\mathbb{R}} \exp(-\xi^2/4+i\xi x)\,\mathrm{d}\xi = \frac{1}{2\sqrt{\pi}}\int_{\mathbb{R}} \exp(-(\xi^2/4-i\xi x-x^2)-x^2)\,\mathrm{d}\xi \\ &= \frac{1}{2\sqrt{\pi}}\int_{\mathbb{R}} \exp(-(\xi/2-ix)^2)\exp(-x^2)\,\mathrm{d}\xi \\ &= \frac{1}{2\sqrt{\pi}}\exp(-x^2)\int_{\mathbb{R}} \exp(-(\xi/2-ix)^2)\,\mathrm{d}\xi.\end{aligned}$$

Die Variablensubstitution $u := \xi/2 - ix$, $\mathrm{d}\xi = 2\,\mathrm{d}u$ liefert daraus schließlich

$$\frac{1}{2\sqrt{\pi}} \exp(-x^2) \underbrace{\int_{\mathbb{R}} 2\exp(-u^2)\,\mathrm{d}u}_{=\sqrt{\pi}} = \frac{1}{\sqrt{\pi}} \sqrt{\pi} \exp(-x^2) = \exp(-x^2) = f(x),$$

wie behauptet.

**Bemerkung 1.21** (Zusammenhang zwischen Symmetrie einer Funktion und deren Fourier-Transformierter) Ist $f \in L_1(\mathbb{R})$ eine symmetrische Funktion, d.h. $f(x) = f(-x)$ für alle $x \in \mathbb{R}$, dann ist auch $\widehat{f}$ symmetrisch und reell. Denn es gilt:

$$\widehat{f}(\xi) = \int_{\mathbb{R}} f(x) e^{-i\xi x}\,\mathrm{d}x = \int_{\mathbb{R}} f(x)\cos(\xi x)\,\mathrm{d}x - i\int_{\mathbb{R}} f(x)\sin(\xi x)\,\mathrm{d}x.$$

Für das rechte Integral gilt

$$\begin{aligned}
\int_{\mathbb{R}} f(x)\sin(\xi x)\,\mathrm{d}x &= \int_{-\infty}^{0} f(x)\sin(\xi x)\,\mathrm{d}x + \int_0^{\infty} f(x)\sin(\xi x)\,\mathrm{d}x \\
&= -\int_{\infty}^{0} f(-t)\sin(-\xi t)\,\mathrm{d}t + \int_0^{\infty} f(x)\sin(\xi x)\,\mathrm{d}x \\
&\qquad\qquad \text{(Subst. } t := -x,\ \mathrm{d}t = -\mathrm{d}x) \\
&= \int_0^{\infty} f(t)\sin(-\xi t)\,\mathrm{d}t + \int_0^{\infty} f(x)\sin(\xi x)\,\mathrm{d}x \qquad (f(x) = f(-x)) \\
&= -\int_0^{\infty} f(x)\sin(\xi x)\,\mathrm{d}x + \int_0^{\infty} f(x)\sin(\xi x)\,\mathrm{d}x \qquad (\sin(-x) = -\sin(x)) \\
&= 0,
\end{aligned}$$

und damit

$$\widehat{f}(\xi) = \int_{\mathbb{R}} f(x)\cos(\xi x)\,\mathrm{d}x \in \mathbb{R}, \qquad \widehat{f}(-\xi) = \widehat{f}(\xi),$$

wie behauptet.

Auf gleiche Weise lässt sich zeigen, dass die Fourier-Transformation zu einer antisymmetrischen Funktion rein imaginär ist.

**Satz 1.22** (Riemann-Lebesgue-Lemma) *Ist $f \in L_1(\mathbb{R})$, so ist*

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

**Bemerkung 1.23** (Riemann-Lebesgue-Lemma) Die Kernessenz von Lemma 1.22 ist, dass die Fourier-Transformation $\widehat{f}$ einer Funktion $f$ im Unendlichen verschwindet. Der Anteil der sehr großen Frequenzen nimmt immer weiter ab, je größer die Frequenzen werden. Aber Achtung: Der Satz macht keine Aussage darüber, wie schnell das Abklingverhalten von $\widehat{f}$ ist! Daher gilt im allgemeinen *nicht*, dass $\widehat{f} \in L_1(\mathbb{R})$, wenn $f \in L_1(\mathbb{R})$. Das haben wir ja bereits gesehen anhand der charakteristischen Funktion.

**Definition 1.24** (Fourier-Koeffizienten und Fourier-Reihe) Zu $f \in L_1(\mathbb{T})$ sind die *Fourier-Koeffizienten* definiert als

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

und die *Fourier-Reihe* zu $f$ als

$$\mathcal{F} f := \sum_{k \in \mathbb{Z}} f_k e^{ik\bullet}$$

**Bemerkung 1.25** (Fourier-Koeffizienten und Fourier-Reihe)

- Die Fourier-Reihe $\mathcal{F} f$ ist $2\pi$-periodisch und unter bestimmten Voraussetzungen die Fourier-Transformation der Folge der Fourier-Koeffizienten: $\mathcal{F} f = \widehat{f_k}$.
- Es ist aber im allgemeinen nicht garantiert, dass die Fourier-Reihe von $f$ konvergiert, oder dass sie im Falle der Konvergenz gegen $\widehat{f}$ konvergiert.

**Satz 1.26** (Poisson'sche Summenformel) *Für $f \in L_1(\mathbb{R})$ ist*

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

**Satz 1.27** (Parseval/Plancherel) *Für $f, g \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ ist*

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

*Insbesondere gilt mit $f = g$*

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

**Bemerkung 1.28** (Parseval/Plancherel) Aus dem vorherigen Satz lassen sich gleich mehrere interessante Schlüsse ziehen:

- Die Fourier-Transformation ist bis auf Normierung eine Isometrie auf $L_2$. Hätten wir in Definition 1.11 die Normierung $1/\sqrt{2\pi}$ verwendet, so könnten wir nun schreiben:

  $$\left\| f \right\|_2 = \left\| \widehat{f} \right\|_2$$

  und wir hätten die perfekte Symmetrie. Dies ließe auch die Interpretation zu, dass die Fourier-Transformation energieerhaltend ist (die Zwei-Norm wird ja auch als Energienorm bezeichnet). Da lacht der Physiker ☺.

- Der Satz von Parseval/Plancherel liefert ein Skalarprodukt von Funktionen $f, g \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$:

  $$\langle f, g \rangle = \int_{\mathbb{R}} f(t) \, g(t) \, \mathrm{d}t = \frac{1}{2\pi} \int_{\mathbb{R}} \widehat{f}(\vartheta) \, \overline{\widehat{g}(\vartheta)} \, \mathrm{d}\vartheta.$$

  Das bedeutet, wir dürfen uns insbesondere der schönen Rechenregeln für Skalarprodukte bedienen (Linearität, Symmetrie usw.).

- Wie wir bereits gesehen haben, gibt es Funktionen in $L_1(\mathbb{R})$, für die die Inversionsformel der Fourier-Transformation nicht gilt. Daher wäre es schön, die Fourier-Transformation auf andere Funktionenräume wie den $L_2(\mathbb{R})$ zu verallgemeinern. Dabei können wir ausnutzen, dass $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ dicht in $L_2(\mathbb{R})$ liegt. Wir definieren zu $f \in L_2(\mathbb{R})$ eine Folge

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

  welche für $n \to \infty$ in der Norm $\|\bullet\|_2$ gegen $f$ konvergiert. $\widehat{f_n}$ ist dann eine Cauchy-Folge, die wegen der Dichtheit von $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ in $L_2(\mathbb{R})$ gegen eine Funktion in $L_2(\mathbb{R})$ konvergiert. Diese Funktion nennen wir $\widehat{f}$:

$$\widehat{f} := \lim_{n\to\infty} \widehat{f_n} = \lim_{n\to\infty} \left(\chi_{[-n,n]} \cdot f\right)^\wedge = \lim_{n\to\infty} \int_{-n}^{n} f(x) \exp(-i\xi x)\, \mathrm{d}x, \qquad f \in L_2(\mathbb{R}).$$

**TODO:** Sollen wir mal ein Beispiel rechnen zu einer FT in $L_2$?

## 1.3. Der Abtastsatz

### Definition 1.29

1. Eine Funktion $f \in L_1(\mathbb{R})$ heißt *bandbeschränkt* mit *Bandbreite* $T$ oder kurz $T$*-bandbeschränkt*, wenn
$$\widehat{f}(\xi) = 0, \qquad \xi \notin [-T, T].$$
   Oder anders formuliert: Eine Funktion $f \in L_1(\mathbb{R})$ ist bandbeschränkt, wenn $\widehat{f}$ endlichen Träger besitzt.

2. Der *Sinus Cardinalis* oder sinc-Funktion ist definiert als
$$\operatorname{sinc}(x) := \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \in \mathbb{R} \setminus \{0\}, \\ 1, & x = 0. \end{cases}$$

**Bemerkung 1.30** (Bandbeschränktheit) Ist eine Funktion $T$-bandbeschränkt, dann ist sie auch gleichzeitig $T + n$-bandbeschränkt für jedes $n > 0$. D.h. es gibt eigentlich gar nicht *die* Bandbreite einer Funktion.

**Bemerkung 1.31** (Sinus Cardinalis)

- Der Name stammt daher, dass die Funktion an allen ganzzahligen Stellen entweder den Wert 0 oder 1 annimmt. Abbildung 1.12 zeigt den Verlauf der sinc-Funktion.

- Fun fact: Es gilt
$$\int_{\mathbb{R}} \operatorname{sinc}(x)\, \mathrm{d}x = 1 \quad \text{und damit} \quad \int_{\mathbb{R}} \operatorname{sinc}(x/\pi)\, \mathrm{d}x = \pi.$$

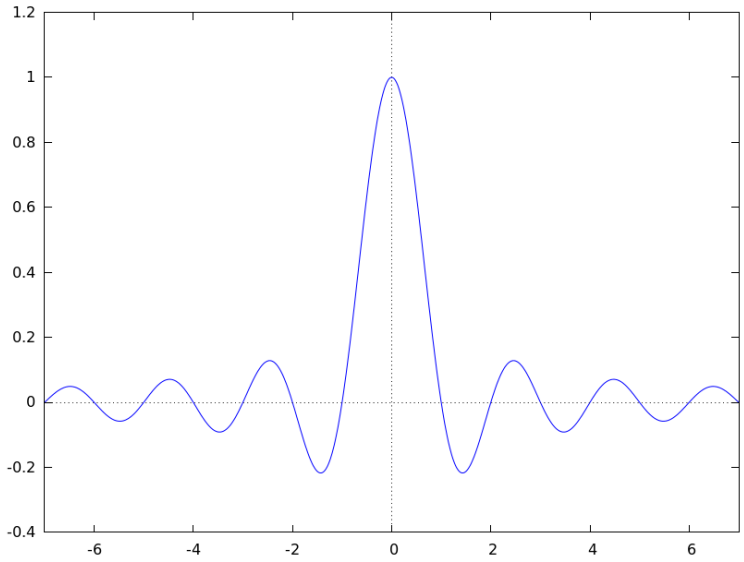


Abbildung 1.12.: Plot der sinc-Funktion auf dem Intervall $[-7, 7]$.

- Die sinc-Funktion ist i.W. bis auf Normierung die Fourier-Transformierte der Rechtecksfunktion. Damit ist der sinc der ideale Tiefpassfilter (aus Sicht des Frequenzbereichs).
- Es kann aber gezeigt werden, dass sinc $\notin L_1(\mathbb{R})$, weswegen der ideale Tiefpassfilter nur näherungsweise implementiert werden kann. Es gilt aber sinc $\in L_2(\mathbb{R})$, was u.a. ein Grund dafür ist, die Fourier-Transformation auf dem $L_2(\mathbb{R})$ einzuführen. Für Beweis siehe Anhang C.

**Satz 1.32** (Abtastsatz von Shannon) *Ist $f \in L_1(\mathbb{R})$ eine $T$-bandbeschränkte Funktion und ist $h < h^* = \pi/T$, dann ist*

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

**Bemerkung 1.33** (Abtastsatz von Shannon) Ein paar Anmerkungen dazu:

- Der Abtastsatz liefert uns eine Formel, wie man das ursprüngliche Signal $f$ aus einer Abtastung $S_h\, f$ mithilfe der sinc-Funktion[3] rekonstruieren kann. Allerdings besitzt der sinc unendlichen Träger und fällt nur linear ab, sodass man in der Praxis wohl lieber mit anderen Funktionen arbeiten sollte.
- Voraussetzung dafür, dass die Rekonstruktion wie gewünscht funktioniert, ist, dass…
  - $f$ nicht zu große Frequenzen enthalten darf (das ist die Forderung der $T$-Bandbeschränktheit). Sonst könnte es potentiell unendlich hohe Frequenzen geben und dann müsste die Schrittweite für die Abtastung unendlich nahe bei der 0 liegen, was nicht wirklich Sinn macht.

[3] Der sinc ist schließlich der ideale Interpolationskern für ganzzahlige Stützstellen. Sagt zumindest Wikipedia.

– die Schrittweite $h$ an den Parameter $T$ angepasst wird, um eine ausreichend große Zahl an Abtastpunkten zu erreichen. Sonst hat man zu wenig Information, um $f$ eindeutig rekonstruieren zu können (siehe Abbildung 1.13). Die unerwünschten Effekte, die aus einer Unterabtastung resultieren, bezeichnet man als *Aliasing*. Bei Bildern können dadurch z.B. neue Strukturen auftreten, die im Original gar nicht enthalten sind.

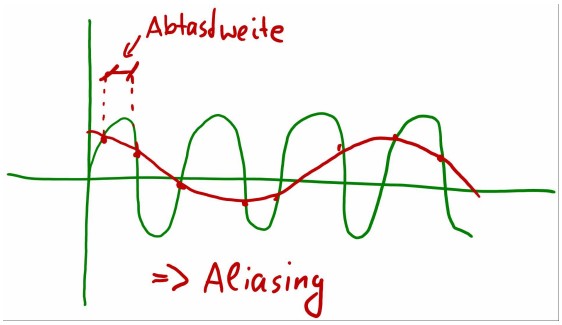


Abbildung 1.13.: Aus der gegebenen Menge an äquidistanten Abtastpunkten kann die hochfrequente grüne Funktion nicht rekonstruiert werden, da auch eine zweite Funktion (rot) gefunden werden kann, welche den Punkten genau entspricht. Die Abtastweite wurde zu groß gewählt.

Die Kopplung von Bandbreite und Abtastfrequenz ist also wirklich essentiell.

- Die Abtastfrequenz $1/h^*$ wird auch als *Nyquist-Frequenz* bezeichnet. Da $f$ laut Voraussetzung bandbeschränkt ist, können wir $\widehat{f}$ periodisieren. Jede periodische Funktion lässt sich *ohne Informationsverlust* als $2\pi$-periodische Funktion darstellen, indem man geeignet skaliert (in unserem Fall mit $h$). Auf den passenden Wert von $h$ kommt man durch eine sehr einfache Rechnung:

$$h[-T,T] \subseteq [-\pi,\pi] \quad \Leftrightarrow \quad [-hT,hT] \subseteq [-\pi,\pi] \quad \Leftrightarrow \quad hT \leq \pi \quad \Leftrightarrow \quad h \leq \pi/T.$$

Darüber hinaus kann jede $2\pi$-periodische Funktion $f$ der Gestalt $f(x) = \sum_{k\in\mathbb{Z}} \gamma_k e^{ikx}$ *exakt* als Fourier-Reihe dargestellt werden. Der Clou beim Beweis und auch im Verständnis des Abtastsatzes ist nun folgende Formel: Wegen der $\pi$-Bandbeschränktheit von $(\sigma_h\, f)$ ist

$$\begin{aligned}(\sigma_h\, f)^\wedge(\theta) &= \sum_{k\in\mathbb{Z}} (\sigma_h\, f)^\wedge(\theta + 2\pi k) = \sum_{k\in\mathbb{Z}} \frac{\widehat{f}\left(\frac{\theta+2\pi k}{h}\right)}{h} \\ &= \sum_{k\in\mathbb{Z}} f(hk)e^{-ik\theta} = (S_h\, f)^\wedge(\theta), \qquad \theta \in \mathbb{T}.\end{aligned}$$

Sie verknüpft unsere $2\pi$-Periodisierung von $\widehat{f}$ mit der Fourier-Transformation der Abtastfolge, $(S_h\, f)^\wedge$, welche genau eine Fourier-Reihe ist. Abbildung 1.14 erklärt den Fall, wenn $h$ ungeeignet gewählt ist.

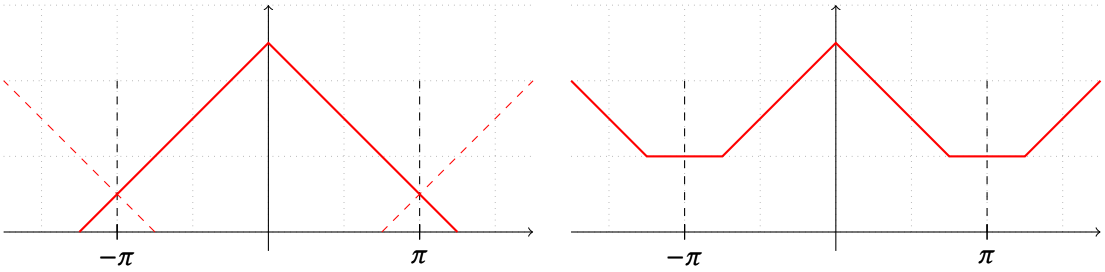


Abbildung 1.14.: Links: Der Träger der skalierten Fourier-Transformation (fetter roter Graph) von $f$ ragt über das Intervall $[-\pi, \pi]$ hinaus. Rot gestrichelt dargestellt ist die $2\pi$- Periodisierung der skalierten Fourier-Transformation. Rechts: Da Frequenzen modulo $2\pi$ betrachtet werden, summieren sich Frequenzen auf, die eigentlich nichts miteinander zu tun haben, und verfälschen damit unser Ergebnis. Das Resultat ist der rot eingezeichnete Graph.

- Wählt man die Abtastfrequenz $k$-mal so hoch wie nötig, so spricht man auch von $k$-fachem Oversampling.

## 1.4. Filter

**Definition 1.34** (Filter) Ein Filter $F$ ist eine Abbildung von einem Signalraum in einen anderen, oder anders formuliert, ein Operator, der eine Folge $c \in l(\mathbb{Z})$ in $Fc \in l(\mathbb{Z})$ abbildet.

**Definition 1.35** (Impulsantwort) Die *Impulsantwort* eines Filters $F$ ist definiert als

$$f(k) := (F\delta)(k), \qquad k \in \mathbb{Z}.$$

**Definition 1.36** (Transferfunktion) Die *Transferfunktion* $\widehat{f}$ eines Filters $F$ bezeichnet die Fourier-Transformiertes der Impulsantwort $f$:

$$\widehat{f} = \widehat{F\delta}.$$

**Bemerkung 1.37** (Filtertypen) Es gibt verschiedene Kategorien von Filtern:

**Energieerhaltender Filter** In diesem Fall ist $F : l_2(\mathbb{Z}) \to l_2(\mathbb{Z})$ so, dass

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

**Linearer Filter** Der Filter ist ein linearer Operator, d.h.

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

**Zeitinvarianter Filter** Der Operator ist *stationär*, d.h. es ist egal, ob

- das Signal zuerst in der Zeit verschoben wird und dann Operator darauf angewendet wird,

- oder zuerst das Signal gefiltert und das Resultat in der Zeit verschoben wird.

Formal:

$$F(c(\bullet + k)) = (Fc)(\bullet + k), \qquad c \in l(\mathbb{Z}), \quad k \in \mathbb{Z}.$$

Oder kürzer über die Kommutativität

$$F\tau_k = \tau_k F.$$

**Kausaler Filter** Das Ergebnis des Filters zum Zeitpunkt $k$ hängt nur von den Eingaben in der Vergangenheit (also $c(j)$ mit $j \leq k$) ab. Der Filter kann nicht in die Zukunft sehen. Kausalität lässt sich auch auffassen als Anforderung an die Impulsantwort des Filters: Für alle $k > 0$ muss $f(k) = (F\delta)(k) = 0$ gelten. Einen Filter mit der Eigenschaft $f(k) = 0$ für alle $k < 0$ bezeichnet man übrigens als *antikausal*. Einen Filter, der weder kausal noch antikausal ist, bezeichnet man als *nicht-kausal*. Man beachte, dass nur ein nicht-kausaler Filter eine reellwertige Transferfunktion besitzen kann!

**Bemerkung 1.38** (Lineare und zeitinvariante Filter)

- Lineare und zeitinvariante Filter (sog. *LTI-Filter*) können immer realisiert werden als *Faltungen mit der Impulsantwort*. Sei also $c \in l(\mathbb{R})$ ein gegebenes diskretes Signal, dann gilt

  $$Fc = c * f,$$

  wie man mit ein wenig Rechnerei nachweisen kann: Unser gegebenes Signal lässt sich auch schreiben als

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

  und unter Verwendung der Linearität und Zeitinvarianz ergibt sich schließlich

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

  Das bedeutet, ein LTI-Filter lässt sich vollständig durch seine Impulsantwort charakterisieren.

- Das Tolle daran ist: Dank der Faltung lassen sich LTI-Filter effizient am Rechner über eine Multiplikation der Transferfunktion mit der Fourier-Transformierten des Signals implementieren. Konkret:

  $$Fc = \left((Fc)^\wedge\right)^\vee = \left((f * c)^\wedge\right)^\vee = \left(\widehat{f} \cdot \widehat{c}\right)^\vee$$

  ist sehr schnell! Denn eine Multiplikation ist wesentlich schneller und vor allem auch numerisch stabiler durchzuführen als eine Faltung. Weiteren Zeitgewinn kann man durch Vorberechnen der Fourier-Transformation der Impulsantwort herausschlagen. Und nicht zuletzt ist die Darstellung im Frequenzbereich nützlich (Filterdesign funktioniert so).

Von jetzt an soll die Bezeichnung *digitaler Filter* immer für einen LTI-Filter stehen.

**Definition 1.39** (Weitere Filtertypen) Ein digitaler Filter $F$ heißt

1. *FIR-Filter* (Finite Impulse Response, Filter mit endlicher Impulsantwort), wenn die Impulsantwort endlichen Träger hat, d.h. $F\delta \in l_{00}(\mathbb{Z})$.

2. *IIR-Filter* (Infinite Impulse Response, Filter mit unendlicher Impulsantwort), wenn die Impulsantwort keinen endlichen Träger hat.

**Beispiel 1.40** (FIR-Filter in der Praxis) In der Praxis sind kausale, digitale FIR-Filter weit verbreitet. Sei $F$ solch ein Filter mit Impulsantwort $f$, wobei $\operatorname{supp}(f) \subseteq [0, N]$, und sei das zu verarbeitende Signal mit $c$ bezeichnet. Dann lässt sich der Filter angewandt auf das Signal darstellen als

$$\begin{aligned} Fc = f * c &= \sum_{k\in\mathbb{Z}} f(k)\, c(\bullet - k) = \sum_{k=0}^{N} f(k)\, \tau_{-k}\, c \\ &= f(0)\, c + f(1)\, \tau_{-1}\, c + f(2)\, \tau_{-2}\, c + \ldots + f(N)\, \tau_{-N}\, c. \end{aligned}$$

Um $F$ in Hardware implementieren zu können, werden also drei Bausteine benötigt: Addierer, Multiplizierer und Verzögerer (welcher das Verschieben des Signals in der Zeit, also den $\tau$-Operator, modelliert).

- Die Verzögerer werden dazu benutzt, um nacheinander die Werte $c(\bullet-1), c(\bullet-2), \ldots, c(\bullet-N)$ einzulesen. Für den ersten Wert $c$ braucht man keinen Verzögerer.
- Diese Werte werden jeweils durch einen Multiplizierer mit den jeweiligen Werten $f(0), f(1), \ldots, f(N)$ multipliziert.
- Zum Schluss werden die Ergebnisse summiert nach folgendem Muster:
  - $s_1 = f(0)\, c + f(1)\, \tau_{-1}$.
  - $s_2 = s_1 + f(2)\, \tau_{-2}$.
  - $s_3 = s_2 + f(3)\, \tau_{-3}$ usw.
  - $Fc = s_{N-1} + f(N)\, \tau_{-N}$.

**TODO:** Hier gehört noch das Bild hin (händisch gezeichnet aufm Tablet wär ganz gut, weil ich will das Zeug nicht tikZen)

Die Quintessenz bei diesem Beispiel ist: Wegen der Verzögerer besitzt jeder Filter in der Praxis eine gewisse Laufzeit.

**Bemerkung 1.41** (Vorgehen beim Filterdesign) Das Design eines Filters erfolgt normalerweise im Frequenzbereich. Das heißt, man stellt Anforderungen an die Transferfunktion (oder halt die Fourier-Transformierte des Filters), z.B. ob hohe oder tiefe Frequenzen durchgelassen werden sollen usw.

- Eine reelle Transferfunktion erhält man nur dann, wenn man auf Kausalität des Filters verzichtet. Sonst kann der Filter nicht symmetrisch sein. Die Transferfunktion eines kausalen Filters ist häufig komplexwertig. Dabei liefert der Realteil die Gewichtung einer Frequenz und der entsprechende Imaginärteil legt die Phasenverschiebung fest.
- Alle Frequenzen des Filters müssen im Intervall $[-\pi, \pi]$ liegen. Dies ist eine Folgerung aus dem Abtastsatz.

Es wäre jetzt natürlich schön, wenn wir jeden Filter als FIR-Filter realisieren könnten. Eine endliche Impulsantwort ist ja eine Grundvoraussetzung dafür, dass man den Filter effizient implementieren kann (sonst müsste man bei der Faltung eine unendliche Summe ausrechnen). Man könnte zwar prinzipiell jeden IIR-Filter künstlich in einen FIR-Filter umwandeln, indem man die Transferfunktion in einem gewissen Bereich bewusst auf 0 setzt. Dabei muss man aber aufpassen, dass man nicht zu hart abschneidet, da der Filter an diesen Stellen sonst unangenehme Artefakte liefern kann. Wie dem auch sei, mit solchen Problemen möchte man sich normalerweise erst gar nicht herumschlagen. Man möchte lieber gleich einen FIR-Filter haben. Ein Problem beim Filterdesign besteht nun darin, dass sich aber viele Filter gar nicht als FIR-Filter realisieren lassen. Wir werden daher im Folgenden zunächst erkunden, welche Eigenschaften die Transferfunktion eines FIR-Filters erfüllen muss, und anhand dessen später ein Verfahren kennen lernen, mit dem sich ein gewünschter Filter näherungsweise als FIR-Filter darstellen lässt.

**Definition 1.42** (Trigonometrisches Polynom) Eine unendlich oft stetig differenzierbare Funktion $f \in C^\infty(\mathbb{T})$ der Form

$$f(x) = \sum_{|k| \leq n} f_k e^{ikx} = \sum_{|k| \leq n} f_k \left(e^{ix}\right)^k$$

mit Koeffizienten $f_k \in \mathbb{C}$ und Argument $x \in \mathbb{T}$ bezeichnet man als trigonometrisches Polynom der Ordnung $n$.

**Bemerkung 1.43**

- Wie bereits in der Übung bewiesen, lässt sich jedes trigonometrische Polynom der Ordnung $n$ auch darstellen als

  $$f(x) = a_0 + \sum_{k=1}^{n} \left(a_k \cos^k(x) + b_k \sin^k(x)\right),$$

  wobei die Koeffizienten $a_0, \ldots, a_n, b_1, \ldots, b_n \in \mathbb{C}$ geeignet gewählt sind.
- Wir können direkt folgern, dass die Transferfunktion $\widehat{f}$ zu einem FIR-Filter $F$ ein trigonometrisches Polynom sein muss. Denn die Impulsantwort $f$ hat endlichen Träger, sagen wir $\operatorname{supp}(f) \subseteq [-N, N]$ und dann gilt ja

  $$\widehat{f}(x) = \sum_{k \in \mathbb{Z}} f(k) e^{-ikx} = \sum_{k=-N}^{N} f(k) e^{-ikx} = \sum_{|k| \leq N} f(k) e^{ikx},$$

was genau der Definition eines trigonometrischen Polynoms entspricht, wenn man $f(k) = f_k$ setzt.

**Bemerkung 1.44** (Best-Approximation einer Transferfunktion eines FIR-Filters) Da die Transferfunktion $\widehat{f}$ eines FIR-Filters immer ein trigonometrisches Polynom ist, und die trigonometrischen Polynome dicht in $L_2(\mathbb{T})$ liegen, ist es auch immer möglich, $\widehat{f}$ durch die $n$-te Partialsumme einer Fourier-Reihe zu approximieren und durch die Fourier-Reihe *exakt* darzustellen.

**Beispiel 1.45** (Approximation des Sägezahn-Filters) Anstatt das gleiche Beispiel wie im Skript abzudrucken (idealer Tiefpassfilter), betrachten wir nun eine Variation der Sägezahnfunktion als Transferfunktion. Diese sei definiert durch

$$g\colon \mathbb{R} \to \mathbb{R}, \quad g(x) = \begin{cases} \dfrac{\pi - x}{2}, & x \in \mathbb{T} \setminus \{0\} \\ 0, & x = 0. \end{cases}$$

Die Funktion ist offensichtlich $2\pi$-periodisch und *kein* trigonometrisches Polynom. Abbildung 1.15 zeigt den Verlauf der Funktion auf dem Intervall $[-\pi, \pi]$. Wir wollen also ein tri-

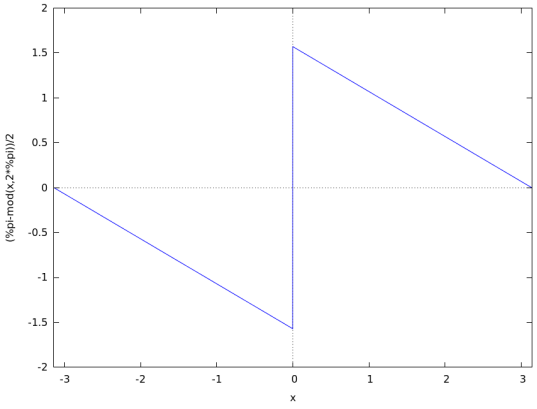


Abbildung 1.15.: Die Sägezahnfunktion $g$ auf dem Intervall $[-\pi, \pi]$.

gonometrisches Polynom finden, welches $g$ besonders gut annähert. Dazu verwenden wir Fourier-Reihen. Hierfür müssen zuerst die sog. Fourier-Koeffizienten berechnet werden. Der erste Fourier-Koeffizient ist gegeben durch

$$\begin{aligned} g_0 &= \frac{1}{2\pi} \int_{\mathbb{T}} g(x) e^{-i \cdot 0 \cdot x} \, \mathrm{d}x = \frac{1}{2\pi} \int_0^{2\pi} \frac{\pi - x}{2} \, \mathrm{d}x = \frac{1}{2\pi} \left( \frac{\pi}{2} \int_0^{2\pi} \mathrm{d}x - \int_0^{2\pi} \frac{x}{2} \, \mathrm{d}x \right) \\ &= \frac{1}{2\pi} \left( \frac{\pi}{2} \cdot 2\pi - \frac{4\pi^2}{4} \right) = \frac{1}{2\pi} \left( \pi^2 - \pi^2 \right) = 0. \end{aligned}$$

Für $k \in \mathbb{Z} \setminus \{0\}$ hingegen ergibt sich mit partieller Integration

$$g_k = \frac{1}{2\pi} \int_{\mathbb{T}} g(x) e^{-ikx} \, \mathrm{d}x = \frac{1}{2\pi} \int_0^{2\pi} \frac{\pi - x}{2} e^{-ikx} \, \mathrm{d}x$$

$$= \frac{1}{4\pi}\left(\pi \underbrace{\int_0^{2\pi} e^{-ikx}\,\mathrm{d}x}_{=0} - \int_0^{2\pi} xe^{-ikx}\,\mathrm{d}x\right) = -\frac{1}{4\pi}\left(x\frac{e^{-ikx}}{-ik}\Big|_0^{2\pi} - \underbrace{\int_0^{2\pi} \frac{e^{-ikx}}{-ik}\,\mathrm{d}x}_{=0}\right)$$

$$= \frac{1}{4\pi ik}\left(xe^{-ikx}\Big|_0^{2\pi}\right) = \frac{1}{4\pi ik}\left(2\pi \underbrace{e^{-2\pi ik}}_{=1} - 0\right) = \frac{1}{2ik}.$$

Damit ist die Fourier-Reihe von $g$

$$\mathcal{F}g = \sum_{k\in\mathbb{Z}} g_k e^{ikx} = \sum_{k=1}^{\infty} \frac{e^{ikx} - e^{-ikx}}{2ik} = \sum_{k=1}^{\infty} \frac{\sin(kx)}{k}$$

und deren $n$-te Partialsumme definiert durch

$$\mathcal{F}_n g := \sum_{k=1}^{n} \frac{\sin(kx)}{k}.$$

Diese $n$-te Partialsumme verwenden wir für unseren näherungsweisen FIR-Filter. Abbildung 1.16 zeigt dies exemplarisch für die Werte $n = 5, 10, 100$. Man erkennt sehr schön, dass die Approxima-

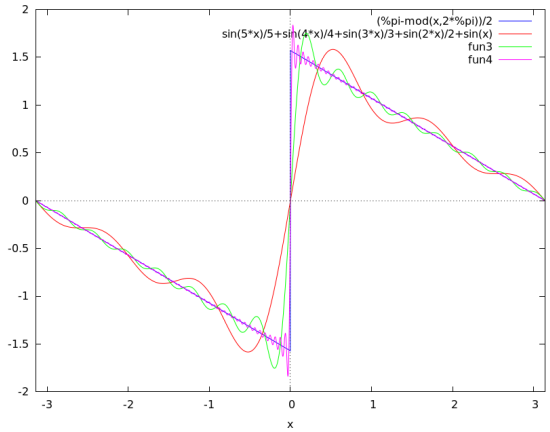


Abbildung 1.16.: Approximation von $g$ (blau) durch die Partialsummen $\mathcal{F}_n g$ für $n = 5, 10, 100$ (rot, grün, lila).

tionsqualität für größeres $n$ immer besser wird. Allerdings fallen die Überschießer in der Nähe von $x = 0$ (hier macht die Funktion zwei Sprünge) unangenehm auf. Diese werden mit zunehmendem $n$ zwar immer schmaler, aber eben nicht kleiner. Diese Beobachtung bezeichnet man auch als *Gibbs-Phänomen*. Dies macht Fourier-Reihen zur Approximation von Bandpassfiltern nicht immer zur besten Wahl.

Man kann versuchen, die problematischen Überschwinger und Oszillationen zu glätten. Dafür definieren wir uns die *Fejér'schen Mittel*:

$$\mathcal{G}_n := \frac{1}{n+1}\sum_{k=0}^{n} \mathcal{F}_k, \quad n \in \mathbb{N}.$$

Anstatt nur eine Partialsumme $\mathcal{F}_n$ betrachten wir also das arithmetische Mittel aller Partialsummen $\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_n$. Die Fejér'schen Mittel sind für Werte von $n = 5, 10, 100$ in Abbildung 1.17 dargestellt. Man sieht recht schön, dass diese zwar eine größere Abweichung von $g$ aufweisen

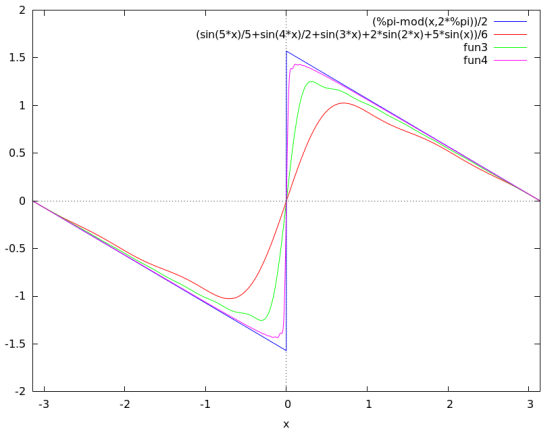


Abbildung 1.17.: Approximation von $g$ (blau) durch die Fejér'schen Mittel $\mathcal{G}_n g$ für $n = 5, 10, 100$ (rot, grün, lila).

als die $n$-ten Partialsummen der Fourier-Reihe von $g$. Dafür vermeiden sie aber das Gibbs-Phänomen.

## 1.5. Filter für Bilder

**Definition 1.46** (Bild) Ein Bild ist ein *zweidimensionales* Signal, also eine Funktion von $\mathbb{R}^2 \to \mathbb{R}$ bzw. von $\mathbb{Z}^2 \to \mathbb{R}$.

**Definition 1.47** (Signalklassen) Mit $l(\mathbb{Z}^2)$ bezeichnen wir die Menge aller Signale der Form

$$c = \left(c(\alpha) : \alpha \in \mathbb{Z}^2\right) = \left(c(\alpha_1, \alpha_2) : \alpha_1, \alpha_2 \in \mathbb{Z}\right).$$

Den Index $\alpha$ nennt man auch *Multiindex*.

Auch für Bilder lassen sich die Normen

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

für den Fall, dass $c \in l(\mathbb{R}^2)$, bzw.

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

falls $f : \mathbb{R}^2 \to \mathbb{R}$, definieren. Damit sind insbesondere die Signalräume wie $L_1(\mathbb{R})$, $l_1(\mathbb{Z})$, usw. für Bilder wohldefiniert.

**Bemerkung 1.48** (Formeln für Bilder) Der ganze Kram, den wir in den vorherigen Abschnitten für eindimensionale Signale definiert haben, lässt sich nun sehr leicht auf das zweidimensionale übertragen (meistens muss man bei den Formeln einfach nur $\mathbb{R}$ durch $\mathbb{R}^2$ ersetzen und hier und da eine Variable transponieren). Seien im Folgenden $f, g$ zwei Bilder (kontinuierlich modelliert).

**Fourier-Transformation** Für ein $\xi \in \mathbb{R}^2$ ist

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

**Faltung** Die Faltung von $f$ und $g$ ist definiert als

$$f * g = \int_{\mathbb{R}^2} f(x) g(\bullet - x) \,\mathrm{d}x.$$

Auch hier gilt die sehr nützliche Identität

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

**Inverse Fourier-Transformation**

$$f(x) = \left(\widehat{f}\right)^\vee(x) = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} \widehat{f}(t) e^{it^\top x} \,\mathrm{d}t = \frac{1}{2\pi} \int_{\mathbb{R}} \frac{1}{2\pi} \int_{\mathbb{R}} \widehat{f}(t) e^{it_1 x_1} \,\mathrm{d}t_1 \; e^{it_2 x_2} \,\mathrm{d}t_2.$$

**Parseval/Plancherel**

$$\int_{\mathbb{R}^2} f(t) g(t) \,\mathrm{d}t = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} \widehat{f}(\xi) \overline{\widehat{g}(\xi)} \,\mathrm{d}\xi$$

und insbesondere mit $f = g$

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

**Poisson'sche Summenformel**

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

Für die Beweise sei auf die entsprechenden Aufgaben auf den Übungsblättern verwiesen.

**Bemerkung 1.49** (LTI-Filter für Bilder)

- Verwenden wir das Kronecker-Delta als

$$\delta(k) = \delta_{(0,0),(k_1,k_2)} = \delta_{0,k_1} \delta_{0,k_2} = \begin{cases} 1, & k_1, k_2 = 0, \\ 0, & \text{sonst}, \end{cases} \qquad k = (k_1, k_2) \in \mathbb{R}^2,$$

dann können wir auch für Filter $F$, die auf zweidimensionalen Signalen operieren, eine Impulsantwort definieren:

$$f = F\delta \in l(\mathbb{Z}^2).$$

- Dementsprechend kann man, falls $F$ ein LTI-Filter ist, die Filterung eines Bildes $c$ wieder schreiben als Faltung der Impulsantwort von $F$ mit dem Bild:

$$Fc = f * c = \sum_{\alpha \in \mathbb{Z}^2} f(\bullet - \alpha) c(\alpha) = \sum_{\alpha_1 \in \mathbb{Z}} \sum_{\alpha_2 \in \mathbb{Z}} f(\bullet - (\alpha_1, \alpha_2))\ c(\alpha_1, \alpha_2).$$

- Diese Formel kann man noch weiter vereinfachen, wenn $f$ ein Tensorprodukt ist, d.h. falls

$$f = f_1 \otimes f_2, \quad f(\alpha) = f_1(\alpha_1) \cdot f_2(\alpha_2)$$

für $f_1, f_2 \in l(\mathbb{Z})$ und $\alpha = (\alpha_1, \alpha_2) \in \mathbb{R}^2$. Denn dann gilt weiter

$$\begin{aligned} f * c &= \sum_{\alpha_1 \in \mathbb{Z}} \sum_{\alpha_2 \in \mathbb{Z}} f(\bullet - (\alpha_1, \alpha_2))\ c(\alpha_1, \alpha_2) \\ &= \sum_{\alpha_1 \in \mathbb{Z}} \sum_{\alpha_2 \in \mathbb{Z}} f_1(\bullet_1 - \alpha_1)\ f_2(\bullet_2 - \alpha_2)\ c(\alpha_1, \alpha_2) \\ &= \sum_{\alpha_1 \in \mathbb{Z}} f_1(\bullet_1 - \alpha_1) \sum_{\alpha_2 \in \mathbb{Z}} f_2(\bullet_2 - \alpha_2)\ c(\alpha_1, \alpha_2) \\ &= \sum_{\alpha_1 \in \mathbb{Z}} f_1(\bullet_1 - \alpha_1) \left( f_2 * c(\alpha_1, \bullet) \right). \end{aligned}$$

Das bedeutet: Ist die Impulsantwort des LTI-Filters ein Tensorprodukt, dann erfolgt die Filterung eines Bildes folgendermaßen:

1. Zuerst werden die Zeilen $c(\alpha_1, \bullet)$ des Bildes für jedes feste $\alpha_1$ mit $f_2$ gefiltert. Diese Operation reduziert das Bild auf einen einzigen Ergebnisvektor, welchen wir als Spalte auffassen.
2. Diese Spalte wird anschließend mit $f_1$ gefiltert. Das Resultat ist dann gerade $f * c$.

Als grafische Veranschaulichung sei auf Abbildung 1.18 verwiesen. Man kann sich leicht überlegen, dass die Filterung der Spalten mit $f_1$ zuerst und anschließende Filterung der Ergebniszeile mit $f_2$ zum obigen Vorgehen äquivalent ist. Wenn man das Bild um 90° dreht, sollte sich ja schließlich nichts ändern, oder?

Das ist zwar eine schöne Anschauung, aber die Darstellung von $f$ als Tensorprodukt ist ein echtes Killer-Feature: Sie ermöglicht nämlich eine noch effizientere Berechnung der Filterung! Wir werden dies am Beispiel des Binomialfilters weiter unten kurz besprechen.

- Auch die Transferfunktion $\widehat{f}$ lässt sich besonders elegant schreiben, falls $f = f_1 \otimes f_2$:

$$\begin{aligned} \widehat{f}(\xi) &= \sum_{\alpha \in \mathbb{Z}^2} f(\alpha) e^{-i\alpha^\top \xi} = \sum_{\alpha_1, \alpha_2 \in \mathbb{Z}} f(\alpha_1, \alpha_2) e^{-i(\alpha_1 \xi_1 + \alpha_2 \xi_2)} \\ &= \sum_{\alpha_1 \in \mathbb{Z}} \sum_{\alpha_2 \in \mathbb{Z}} f_1(\alpha_1) f_2(\alpha_2) e^{-i\alpha_1 \xi_1} e^{-i\alpha_2 \xi_2} \\ &= \left( \sum_{\alpha_1 \in \mathbb{Z}} f_1(\alpha_1) e^{-i\alpha_1 \xi_1} \right) \left( \sum_{\alpha_2 \in \mathbb{Z}} f_2(\alpha_2) e^{-i\alpha_2 \xi_2} \right) \end{aligned}$$

$$
\begin{array}{ccccc}
\ddots & \vdots & \vdots & \ddots & \vdots \\
\cdots & c(0,0) & c(0,1) & \cdots & \to f_2 * c(0,\bullet) = \quad c'(0) \\
\cdots & c(1,0) & c(1,1) & \cdots & \to f_2 * c(1,\bullet) = \quad c'(1) \\
\ddots & \vdots & \vdots & \ddots & \vdots \\
 & & & & \downarrow \\
 & & & & f_1 * c' \\
 & & & & \downarrow \\
 & & & & f * c
\end{array}
$$

Abbildung 1.18.: Schematische Darstellung der Funktionsweise eines FIR-Filters für Bilder, wenn dessen Impulsantwort eine Tensorproduktstruktur aufweist. Zuerst wird das Bild horizontal gefiltert und so zu einer Spalte reduziert, welche dann vertikal gefiltert wird, um so das Endergebnis zu erhalten.

$$= \widehat{f_1}(\xi_1) \cdot \widehat{f_2}(\xi_2).$$

Und damit wird die Anwendung des Filters besonders einfach:

$$Fc = \left((Fc)^\wedge\right)^\vee = \left((f * c)^\wedge\right)^\vee = \left(\widehat{f} \cdot \widehat{c}\right)^\vee = \left(\widehat{f_1} \cdot \widehat{f_2} \cdot \widehat{c}\right)^\vee.$$

Die beiden Transferfunktionen $\widehat{f_1}$ und $\widehat{f_2}$ kann man bereits vor dem Anwenden des Filters berechnen, sodass wir nur noch eine schnelle Methode benötigen, um $\widehat{c}$ zu bestimmen. Diese werden wir im nächsten Kapitel mit der *Schnellen Fourier-Transformation (FFT)* kennenlernen.

**Beispiel 1.50** (Filter in der Bildverarbeitung) Wir betrachten im Folgenden ein paar Beispiele für bekannte Filter in der Bildverarbeitung. Dabei ist es meistens so, dass die Filter für Signale der Form $\phi\colon \mathbb{R}^2 \to \mathbb{R}$, also kontinuierliche Daten, konzipiert sind und später erst diskretisiert werden.

**Mittelwertfilter** Der Mittelwertfilter $F$ mit Impulsantwort

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

filtert das Signal $\phi$ mittels

$$\phi(x) \mapsto (f * \phi)(x) = \frac{1}{|\Omega|}\int_{\mathbb{R}} \chi_\Omega(t)\phi(x-t)\,\mathrm{d}t = \frac{1}{|\Omega|}\int_{\Omega} \phi(x-t)\,\mathrm{d}t = \frac{1}{|\Omega|}\int_{x+\Omega} \phi(t)\,\mathrm{d}t.$$

Das heißt, er ordnet $\phi(x)$ den Mittelwert der Menge $x + \Omega$ zu. Um den Filter zu diskretisieren, setzt man einfach $f = S_h\chi_\Omega$ und normalisiert dann, indem man durch die Anzahl der in $\Omega$ enthaltenen Abtastpunkte teilt.

Ein großer Vorteil des Mittelwertfilters ist, dass er Rauschen unterdrücken kann. Dazu modelliert man das gemessene Signal als $\phi(x) = \psi(x) + \epsilon(x)$, wobei $\psi$ die eigentlichen Daten sind und $\epsilon$ das (häufig als mittelwertfrei angenommene) Rauschen. Insgesamt ergibt sich so mit diesem Modell die Filterung

$$F\phi(x) = \frac{1}{|\Omega|}\int_{x+\Omega} \psi(t)\,\mathrm{d}t + \frac{1}{|\Omega|}\int_{x+\Omega} \epsilon(t)\,\mathrm{d}t.$$

Feine Strukturen (hohe Frequenzen) gehen so verloren, grobe Strukturen (kleine Frequenzen) bleiben erhalten. So gesehen verhält sich ein Mittelwertfilter wie ein Tiefpassfilter.

Das Problem beim Design eines Mittelwertfilters ist die korrekte Wahl von $\Omega$. Denn der Filter angewandt auf die eigentlichen Daten $\psi$ sollte diese weitestgehend unberührt lassen, d.h. $F\psi \sim \psi$. Dafür muss $\Omega$ eine möglichst kleine Menge sein. Gleichzeitig soll aber auch das Rauschen möglichst gut ausgemittelt werden, und dafür muss $\Omega$ groß und »symmetrisch« genug sein.

Eine andere Möglichkeit, einen Mittelwertfilter zu realisieren, ist

$$f(x_1, x_2) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x_1^2 + x_2^2}{2\sigma^2}\right), \qquad \sigma > 0$$

zu setzten. So steht sogar noch der freie Parameter $\sigma$ (die Standardabweichung) zur Verfügung, um das Verhalten des Filters zu spezialisieren. Für die Diskretisierung tastet man $f\chi_{[-N,N]^2}$ für ein $N \in \mathbb{N}$ ab. Die charakteristische Funktion wird benötigt, um einen FIR-Filter zu erhalten, denn der Gaußkern besitzt ja unendlichen Träger. Diese Diskretisierung führt uns sogleich zum nächsten Filter.

**Binomialfilter** Die Binomalfilter sind spezielle Mittelwertfilter, denen wir in einem eigenen Punkt Aufmerksamkeit widmen wollen, und das diskrete Gegenstück zum oben erwähnten Gaußkern. Deren Impulsantwort ist gegeben durch

$$f(j,k) = \frac{1}{2^{m+n}}\binom{m}{j}\binom{n}{k} = \left(\frac{1}{2^m}\binom{m}{j}\right)\left(\frac{1}{2^n}\binom{n}{k}\right), \qquad 0,\dots,m, \quad 0,\dots,n.$$

Wie man sieht, besitzt die Impulsantwort Tensorproduktstruktur. $f$ lässt sich auch als Matrix darstellen, wenn man $j$ als Zeile und $k$ als Spalte auffasst. Geschickt ist es, $m$ und $n$ als gerade zahlen zu wählen, denn dann lässt sich die Matrix sogar zentrieren. Um etwas Terminologie aus der Informatik zu verwenden: Man erhält so einen $m \cdot n$-Punkt-Stencil, der sukzessive über das Bild geschoben wird (mathemat. Faltung). Beispielsweise erhält man für $m, n = 2$

$$\frac{1}{4}\begin{pmatrix}1\\ \mathbf{2}\\ 1\end{pmatrix} * \frac{1}{4}\begin{pmatrix}1 & \mathbf{2} & 1\end{pmatrix} = \frac{1}{16}\begin{pmatrix}1 & 2 & 1\\ 2 & \mathbf{4} & 2\\ 1 & 2 & 1\end{pmatrix}$$

und für $m, n = 4$

$$\frac{1}{16}\begin{pmatrix}1\\4\\\mathbf{6}\\4\\1\end{pmatrix} * \frac{1}{16}\begin{pmatrix}1 & 4 & \mathbf{6} & 4 & 1\end{pmatrix} = \frac{1}{256}\begin{pmatrix}1 & 4 & 6 & 4 & 1\\4 & 16 & 24 & 16 & 4\\6 & 24 & \mathbf{36} & 24 & 6\\4 & 16 & 24 & 16 & 4\\1 & 4 & 6 & 4 & 1\end{pmatrix}.$$

Der fett gedruckte Wert bezeichnet dabei jeweils die aktuelle Zelle des Stencils. Je größer nun die Werte für $m$ und $n$ gewählt, werden, desto mehr benachbarte Zellen werden für die Berechnung hinzugezogen und umso stärker wird das Bild dadurch geglättet.

Mittlerweile sollte auch aufgefallen sein, dass die Implementierung des Filters über die Faltung des $m$-Punkt- mit dem $n$-Punkt-Stencils um Größenordnungen effizienter ist als der $m \cdot n$-Punkt-Stencil. Denn im ersten Fall werden lediglich $O(m+n)$ Speicherzugriffe benötigt (linearer Anstieg), wohingegen im zweiten Fall $O(m \cdot n)$ Speicherzugriffe fällig sind (quadratischer Anstieg). Auch die Zahl der auszuführenden Operationen (Addition, Multiplikation) wird so verringert. Dieser Performance-Unterschied wird mit wachsenden $n, m$ immer deutlicher. Genau hierin liegt der Vorteil, wenn man es schafft, die Impulsantwort als Tensorprodukt darzustellen!

Übrigens ist $2^{m+n}$ gerade die Summe aller Einträge in der Matrix, denn es gilt

$$\begin{aligned}\sum_{j=0}^{m}\sum_{k=0}^{n}\binom{m}{j}\binom{n}{k} &= \left(\sum_{j=0}^{m}\binom{m}{j}\right)\left(\sum_{k=0}^{n}\binom{n}{k}\right) = \left(\sum_{j=0}^{m}\binom{m}{j}1^m 1^{m-j}\right)\left(\sum_{k=0}^{n}\binom{n}{k}1^n 1^{n-k}\right)\\ &= (1+1)^m(1+1)^n = 2^m 2^n = 2^{m+n},\end{aligned}$$

sodass wir es wirklich mit einem Mittelwertfilter zu tun haben!

Abbildung 1.19 veranschaulicht die Wirkung des Binomialfilters an einem kleinen Beispiel.

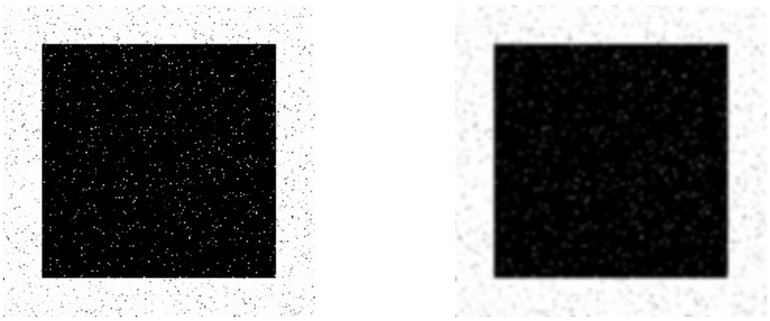

Abbildung 1.19.: Links: Ein ziemlich verrauschtes Bild eines Quadrats. Rechts: Anwendung des Binomialfilters auf das Bild. Es wurde die oben gezeigte Faltungs-Matrix für $m, n = 4$ verwendet. Man erkennt, dass das Bild zwar deutlich entrauscht wurde, dafür sind die scharfen Kanten um Einiges verschwommener.

**Gradientenfilter** Der Gradientenfilter verwendet, wie der Name schon erahnen lässt, den Gradienten unseres Signals $\phi$. Allgemein ist der Gradient einer Funktion $\phi$ definiert als

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

Anschaulich: Wir bilden die partiellen Ableitungen zu $\phi$ und stecken diese in einen Vektor (den Gradienten). Dieser zeigt dann in diejenige Richtung, in der $\phi$ an der Stelle $(x,y)$ am stärksten steigt oder fällt. So können wir Konturen und Kanten im Bild erkennen. Denn eine Kontur findet sich ja dort, wo der (Farb-)Unterschied zwischen benachbarten Pixeln groß ist und damit die Steigung groß ist.

Den Gradienten für ein diskretes Signal $c$ können wir definieren über den Differenzen-Quotienten als

$$\nabla c(x,y) = \begin{pmatrix} \nabla_x c(x,y) \\ \nabla_y c(x,y) \end{pmatrix} = \begin{pmatrix} c(x+1,y) - c(x,y) \\ c(x,y+1) - c(x,y) \end{pmatrix}.$$

Die Impulsantwort lässt sich darstellen als Matrix von Vektoren

$$\big(\nabla\delta(x,y)\big)_{-1\le x,y\le 1} = \begin{pmatrix} \begin{pmatrix}0\\0\end{pmatrix} & \begin{pmatrix}1\\0\end{pmatrix} & \begin{pmatrix}0\\0\end{pmatrix} \\ \begin{pmatrix}0\\1\end{pmatrix} & \begin{pmatrix}-\mathbf{1}\\-\mathbf{1}\end{pmatrix} & \begin{pmatrix}0\\0\end{pmatrix} \\ \begin{pmatrix}0\\0\end{pmatrix} & \begin{pmatrix}0\\0\end{pmatrix} & \begin{pmatrix}0\\0\end{pmatrix} \end{pmatrix},$$

beziehungsweise

$$\big(\nabla_x\delta(x,y)\big)_{-1\le x,y\le 1} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -\mathbf{1} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \big(\nabla_y\delta(x,y)\big)_{-1\le x,y\le 1} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & -\mathbf{1} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Leider sind solche Gradientenverfahren inhärent empfindlich gegen Rauschen. Und je genauer man das Signal abtastet, umso mehr wird das Rauschen durch den Gradientenfilter noch verstärkt. Daher kombiniert man den Gradienten mit einem Mittelwertfilter, z.B. den Mittelungsoperator

$$m \coloneqq \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & \mathbf{1} & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Die Faltungen $m * \nabla_x$ und $m * \nabla_y$ ergeben dann die Matrizen

$$m * \nabla_x = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & \mathbf{0} & 0 \\ -1 & -1 & -1 \end{pmatrix}, \qquad m * \nabla_y = \frac{1}{9}\begin{pmatrix} 1 & 0 & 0 & -1 \\ 1 & \mathbf{0} & 0 & -1 \\ 1 & 0 & 0 & -1 \end{pmatrix}.$$

Die linke Matrix erkennt dabei Kanten, die in $x$-Richtung verlaufen, und die rechte Matrix entsprechend Kanten in $y$-Richtung. Siehe hierfür Abbildung 1.20. Um Kanten in beide

Abbildung 1.20.: Wirkung des Gradientenfilters. Links: Unser Testbild, welches nur aus einem schwarzen Quadrat besteht. Mitte: Anwendung von $m * \nabla_x$ findet alle Kanten in $x$- Richtung. Rechts: Anwendung von $m * \nabla_y$ findet alle Kanten in $y$-Richtung. Bedingt durch die unterschiedliche Gewichtung (Einträge 1 und $-1$ in den Matrizen) werden die Kanten entweder weiß oder schwarz gezeichnet.

Richtungen gleichzeitig zu finden, kann man beispielsweise beide Matrizen über die 1-Norm des geglätteten Gradienten kombinieren:

$$\|m * \nabla c\|_1 = |m * \nabla_x c| + |m * \nabla_y c|.$$

Oder man verwendet den Laplace-Filter, welchen wir als Nächstes behandeln.

**Laplace-Filter** Der Laplace-Filter verwendet den Laplace-Operator

$$\Delta\phi(x,y) = \left\langle \nabla\phi(x,y), \nabla\phi(x,y) \right\rangle = \frac{\partial^2\phi}{\partial x^2}(x,y) + \frac{\partial^2\phi}{\partial y^2}(x,y),$$

welcher das Skalarprodukt des Gradienten eines kontinuierlichen Signals $\phi$ mit sich selbst, also die Summe der zweiten Partiellen Ableitungen bildet. Die Idee dabei ist, dass die zweiten Ableitungen lokale Extrema angeben. Am Beispiel von Bildern sind dies genau diejenigen Stellen, an denen es die gravierendsten (Farb-)Unterschiede zwischen benachbarten Pixeln gibt. Und dies sind häufig Kanten.

Diskretisierungen des Laplace-Operators sind beispielsweise

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & \mathbf{-4} & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad \begin{pmatrix} 1 & 1 & 1 \\ 1 & \mathbf{-8} & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Wie die Gradientenfilter auch ist der Laplace-Operator sehr rauschempfindlich, sodass es sich wieder empfiehlt, diesen mit einem Glättungsfilter zu kombinieren. Gut geeignet ist z.B. der Medianfilter.

**Medianfilter** Der Medianfilter

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

sortiert alle Werte $c(j+\Omega)$ der Größe nach und wählt daraus den Wert in mittlerer Position aus. Da er Bilder entrauscht, Kanten erhält und von wenigen Ausreißern unberührt arbeitet, wird er häufig zur in Kombination mit anderen Filtern, z.B. zur Kantendetektion, eingesetzt. Allerdings ist der Medianfilter kein linearer Operator und besitzt wegen des Sortierens einen vergleichsweise hohen Rechenaufwand von mindestens $O(n \log n)$.

## 1.6. Die Schnelle Fourier-Transformation (FFT)

Wir kommen nun zum Heiligen Gral in der Bildverarbeitung, nämlich der sogenannten *Schnellen Fourier-Transformation (FFT)*. Da es sich dabei im um eine schnelle Implementierung der Diskreten Fourier-Transformation (DFT) handelt, wollen wir uns diese zunächst etwas genauer ansehen.

### 1.6.1. Die Diskrete Fourier-Transformation (DFT)

Wir werden nun kurz die Formel der Diskreten Fourier-Transformation herleiten. Die (kontinuierliche) Fourier-Transformation einer Folge $c \in l_1(\mathbb{Z})$ ist eine $2\pi$-periodische Funktion $\widehat{c} \in L_1(\mathbb{T})$. Die einfachste Möglichkeit $\widehat{c}$ zu diskretisieren, ist $\widehat{c}$ einfach an allen $2\pi/n$-ten Stellen für ein vorher gewähltes $n \in \mathbb{N}$ abzutasten. So erhalten wir

$$\text{DFT}_n := \widehat{c}_n := S_{2\pi/n}\, \widehat{c} = S_{2\pi/n} \sum_{k\in\mathbb{Z}} c(k) e^{-ik\bullet} = \sum_{k\in\mathbb{Z}} c(k)\, S_{2\pi/n}\, e^{-ik\bullet} = \sum_{k\in\mathbb{Z}} c(k) e^{-2\pi ik\bullet/n}.$$

**Definition 1.51** (Diskrete Fourier-Transformation) Zu einer Folge $c \in l_1(\mathbb{Z})$ bezeichnet man

$$\widehat{c}_n := \sum_{k\in\mathbb{Z}} c(k) e^{-2\pi ik\bullet/n}, \qquad n \in \mathbb{N}$$

als die *Diskrete Fourier-Transformation* (DFT) der Ordnung $n$ von $c$.

**Bemerkung 1.52**

- Die DFT $\widehat{c}_n$ ist wegen

$$\begin{aligned}\widehat{c}_n(\bullet + n) &= \sum_{k\in\mathbb{Z}} c(k) e^{-2\pi ik(\bullet+n)/n} = \sum_{k\in\mathbb{Z}} c(k) e^{-2\pi ik(\bullet/n+1)} = \sum_{k\in\mathbb{Z}} c(k) e^{-2\pi ik\bullet/n} \underbrace{e^{-2\pi ik}}_{=1} \\ &= \sum_{k\in\mathbb{Z}} c(k) e^{-2\pi ik\bullet/n} = \widehat{c}_n(\bullet)\end{aligned}$$

$n$-periodisch. Das heißt, sie ist durch die Werte

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

eindeutig festgelegt.

- Für $m$-periodische oder $m$-periodisierte Folgen $c \in l(\mathbb{Z}_m)$ lässt sich die DFT der Ordnung $n$ auch als Matrix darstellen:

$$\widehat{c}_n = V_{n,m} c, \qquad V_{n,m} := \left(e^{-2\pi ijk/n} : j \in \mathbb{Z}_n, k \in \mathbb{Z}_m\right) \in \mathbb{C}^{n\times m}.$$

  Die Matrix $V_{n,m}$ bezeichnen wir auch als *Transformationsmatrix*.

- Ist $m = n$, so schreiben wir statt $V_{n,n}$ einfach $V_n$. In diesem Fall definieren wir die primitive $n$-te Einheitswurzel

$$\omega := e^{-2\pi i/n}, \qquad \omega^n = e^{-2\pi i} = 1.$$

  Dann gilt

$$V_n = \left(\omega^{jk} : j,k \in \mathbb{Z}_n\right) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-2} & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-2)} & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \omega^{n-2} & \omega^{2(n-2)} & \cdots & \omega^4 & \omega^2 \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^2 & \omega \end{pmatrix}.$$

  Das heißt, die Matrix ist symmetrisch und besitzt vollen Rang, weshalb sie insbesondere immer invertierbar ist. Sie definiert einen Basiswechsel vom Ort-/Zeitbereich in den Frequenzbereich.

**Satz 1.53** (Inverse DFT) *Für ein $n \in \mathbb{N}$ ist die Inverse DFT zu $V_n$ gegeben durch*

$$V_n^{-1} = \frac{1}{n}\left(e^{2\pi ijk/n} : j \in \mathbb{Z}_n, k \in Z_m\right) = \frac{1}{n}\left(\omega^{-jk} : j,k \in \mathbb{Z}_n\right).$$

**Beispiel 1.54** (Berechnung der Transformationsmatrix) Für $n = 4$ ist

$$V_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}, \qquad V_4^{-1} = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & 1 & i \\ 1 & 1 & -1 & 1 \\ 1 & i & 1 & -i \end{pmatrix},$$

das heißt

$$V_4\, V_4^{-1} = \frac{1}{4}\begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

**Beispiel 1.55** (Die DFT in Aktion) Tasten wir einmal den Cosinus an 512 äquidistanten Punkten im Intervall $[0, 2\pi)$ ab, d.h. wir berechnen $S_{2\pi/512}\cos$, und bestimmen wir zu dieser Diskretisierung die DFT der Ordnung 512. Vorüberlegungen:

- Der Cosinus ist eine gerade Funktion. D.h. die DFT des Cosinus ist vollständig reellwertig.

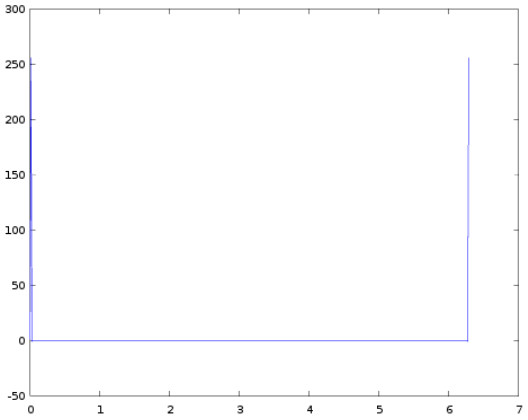


Abbildung 1.21.: DFT der Ordnung 512 zu $S_{2\pi/512}$ cos.

- Der Cosinus lässt sich auch schreiben als

$$\cos(x) = \frac{e^{ix} + e^{-ix}}{2} = \frac{e^{\mathbf{1}\cdot ix}}{2} + \frac{e^{\mathbf{-1}\cdot ix}}{2},$$

  d.h. es treten nur die beiden Frequenzen 1 und $-1 \simeq 511$ im Signal auf, welche beide den gleichen relativen Anteil am Signal haben. Da wir eine DFT der Ordnung 512 verwenden, können wir 512 verschiedene Frequenzen unterscheiden. Der absolute Anteil der beiden oben erwähnten Frequenzen sollte also jeweils $512/2 = 256$ sein. Oder anders gesagt: Im Funktionsplot der DFT muss es an den Stellen $\frac{2\pi}{512}$ und $\frac{2\pi}{512} \cdot 511$ einen Ausschlag von jeweils 256 geben. Und genau das ist auch der Fall, wie Abbildung 1.21 zeigt. Alle anderen Frequenzen haben keinen Anteil am Signal, was auch in der Abbildung zu sehen ist.

Wir wollen nun ein paar wichtige Eigenschaften der Diskreten Fourier-Transformation in Anlehnung an Bemerkung 1.19 zusammenstellen. Dazu definieren wir uns noch schnell den Begriff der Zyklischen Faltung.

**Definition 1.56** (Zyklische Faltung) Zu zwei periodischen Folgen $c, d \in l(\mathbb{Z}_n)$ ist die *zyklische Faltung* $c * d = c *_n d \in l(\mathbb{Z}_n)$ definiert als

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

Dabei ist der Ausdruck $j - k$ den Rechenregeln in $\mathbb{Z}_n$, also modulo $n$, zu verstehen.

**Satz 1.57** (Eigenschaften der DFT) *Für $n \in \mathbb{N}$ und $c, d \in l(\mathbb{Z}_n)$ gilt:*

1. $\mathrm{DFT}_n : l(\mathbb{Z}_n) \to l(\mathbb{Z}_n)$ *ist eine invertierbare lineare Abbildung mit*

$$\|c\|_2 = \frac{1}{\sqrt{n}} \left\| \widehat{c_n} \right\|_2 .$$

*Dabei sind die $p$-Normen definiert als*

$$\|c\|_p := \left( \sum_{k \in \mathbb{Z}_n} |c(k)|^p \right)^{1/p}$$

*bzw.*

$$\|c\|_\infty := \max_{k \in \mathbb{Z}_n} |c(k)|.$$

2. *Es gilt*

$$(c *_n d)^\wedge_n = \widehat{c_n} \odot \widehat{d_n}.$$

*Die Multiplikation $\odot$ auf der rechten Seite ist dabei als punktweises Produkt von Vektoren zu verstehen: $\left(\widehat{c_n} \odot \widehat{d_n}\right)(j) = \widehat{c_n}(j) \cdot \widehat{d_n}(j)$ (sog.* Hadamard-Produkt*).*

**Bemerkung 1.58** (DFT im Zweidimensionalen) Für zweidimensionale Signale $c \in l(\mathbb{Z}_n^2)$ (also bspw. gekachelte Bilder) gilt

$$\begin{aligned}
\widehat{c}(\beta) &= \sum_{\alpha \in \mathbb{Z}_n^2} c(\alpha) e^{-2\pi i \alpha^\top \beta / n} = \sum_{\alpha_1 \in \mathbb{Z}_n} \sum_{\alpha_2 \in \mathbb{Z}_n} c(\alpha_1, \alpha_2) e^{-2\pi i \alpha_1 \beta_1 / n} e^{-2\pi i \alpha_2 \beta_2 / n} \\
&= \sum_{\alpha_1 \in \mathbb{Z}_n} e^{-2\pi i \alpha_1 \beta_1 / n} \underbrace{\sum_{\alpha_2 \in \mathbb{Z}_n} c(\alpha_1, \alpha_2) e^{-2\pi i \alpha_2 \beta_2 / n}}_{= (c(\alpha_1, \bullet))^\wedge (\beta_2)} \\
&= \sum_{\alpha_1 \in \mathbb{Z}_n} (c(\alpha_1, \bullet))^\wedge (\beta_2) e^{-2\pi i \alpha_1 \beta_1 / n}.
\end{aligned}$$

Diese Formel liefert uns eine Berechnungsmethode für zweidimensionale DFTs:

1. Zuerst bildet man die eindimensionale DFT für jede Zeile der Matrix $c$. Dies liefert uns einen Ergebnisvektor mit den Einträgen $(c(0, \bullet))^\wedge$ bis $(c(n-1, \bullet))^\wedge$.

2. Diesen Ergebnisvektor müssen wir nun nochmals mit der eindimensionalen DFT transformieren. Das Ergebnis dieses Vorgangs ist dann die Fourier-Transformation $\widehat{c}$ unserer Matrix.

Schematische Darstellung:

$$\begin{array}{ccccc}
c(0,0) & \cdots & c(0,n-1) & \to & (c(0,\bullet))^\wedge \\
c(1,0) & \cdots & c(1,n-1) & \to & (c(1,\bullet))^\wedge \\
\vdots & \ddots & \vdots & & \vdots \\
c(n-1,0) & \cdots & c(n-1,n-1) & \to & (c(n-1,\bullet))^\wedge \\
 & & & & \downarrow \\
 & & & & \widehat{c}
\end{array}$$

Das heißt, wir benötigen insgesamt $n + 1$ eindimensionale DFTs, und deswegen hängt die Performance der zweidimensionalen DFT signifikant von der eindimensionalen DFT ab.

### 1.6.2. Diskret vs. Diskretisiert

Dieser Teil thematisiert die Probleme, die auftreten, wenn man versucht, aus der DFT eines abgetasteten kontinuierlichen Signals $f$ die Fourier-Transformation von $f$ zu rekonstruieren. Als Beispiel wurde im Skript die DFT der sinc-Funktion angeführt. Diese führt nämlich nicht zur Rechtecksfunktion, wie man eigentlich erwarten würde, sondern zu zwei »Teufelshörnern«. Um die Bildung von solchen Artefakten beim Rekonstruktionsvorgang zu vermeiden, kann man versuchen, $f$ mittels eines sog. *Quasi-Interpolanten* zu approximieren. Häufig verwendete Quasi-Interpolanten sind die kardinalen Splines. Interessant sind eigentlich nur diese beiden Punkte:

1. Die Schrittweite $h$ der Abtastung von $f$ bestimmt, welche Frequenzen von $f$ wirklich in der DFT der Abtastung von $f$ codiert sind. Je feiner die Abtastung, desto mehr Frequenzen werden codiert.

2. Die Frequenzauflösung, also die Zahl der Frequenzen, die sich in der DFT unterscheiden lassen, hängt hingegen von der Ordnung $n$ der DFT ab. Je größer $n$ gewählt wird, umso feiner die Frequenzauflösung. Und je kleiner $n$, umso mehr nicht zusammengehörige Frequenzen werden zu einer Frequenz zusammengefasst.

Aber das hätte man sich auch so irgendwie denken können…

### 1.6.3. Die Schnelle Fourier-Transformation

Wie wir jetzt wissen, lässt sich die Diskrete Fourier-Transformation naiv als Matrix-Vektor-Multiplikation implementieren. Dies ist aber nicht sonderlich effizient, da im Falle einer DFT der Ordnung $n$ somit $O(n^2)$ Rechenoperationen (sprich Multiplikationen und Additionen) durchzuführen sind. Aufgrund der enormen Bedeutung der DFT in den modernen Wissenschaften und deren weit verbreitetem Einsatz in laufzeitkritischen Anwendungen hätte man gerne eine effizientere Möglichkeit, die DFT zu berechnen. Und genau das leistet die *Schnelle Fourier-Transformation* (FFT). Sie verringert die Kosten der Berechnung auf $O(n \log(n))$.

Es gibt eigentlich gar nicht »die eine« FFT, sondern mehrere Varianten der FFT, welche alle abhängig von der Dimension oder der Struktur des Eingabevektors ihre Vor- und Nachteile haben. Die allgemeinste und geläufigste Variante der FFT ist das Verfahren nach James Cooley und John Tukey. Dieses wollen wir im Folgenden studieren.

Hinter der Cooley-Tukey-FFT steckt ein einfacher Divide-and-Conquer-Algorithmus. Angenommen, unser zu transformierendes Signal der Länge $n$ ist $c \in l(\mathbb{Z}_n)$, wobei $n = 2m$ eine gerade Zahl ist. Dann können wir die Fourier-Transformation von $c$ für ein $j \in \mathbb{Z}_n$ auch schreiben als

$$\widehat{c_n}(j) = (c(2\bullet))^\wedge_m(j) + \omega^j \ (c(2 \bullet +1))^\wedge_m(j).$$

Das bedeutet: Wir halbieren den Eingabevektor der Länge $n$ in zwei kleinere Vektoren der Länge $m$, indem wir jeweils nur gerade bzw. ungerade Indizes betrachten. Danach berechnen wir für diese zwei Vektoren jeweils die DFT der Ordnung $m$. Die resultierenden Ergebnisvektoren setzen wir anschließend wieder geschickt zusammen, um so das eigentliche Ergebnis der DFT von $c$ zu erhalten.

Wenden wir dieses Verfahren rekursiv so lange an, bis wir irgendwann nur noch (Teil-)Vektoren der Länge 2 zu verarbeiten haben (deren DFT wir direkt berechnen können), ergibt sich nach dem Master-Theorem eine Komplexität von $O(n\log(n))$. Dies ist typisch für solche Baumalgorithmen. Denn die Tiefe des Berechnungsbaumes ist ja gerade in $O(\log n)$ und die Anzahl der Blätter ist in $O(n)$. Da wir aus den Blättern des Baumes die Gesamtlösung zusammensetzen müssen, ergibt sich somit die oben erwähnte Gesamtkomplexität von $O(n\log(n))$.

Die Form der Cooley-Tukey-FFT, die wir gerade betrachtet haben, ist die sog. *Radix-2-FFT*, da wir die Eingabevektoren immer in zwei Teile zerlegt haben. Darüber hinaus gibt es die Möglichkeit, die Vektoren jeweils in $p$ Teile zu zerlegen, für den Fall, dass $n$ keine gerade Zahl sein sollte. Dies führt dann zu der allgemeinen *Radix-p-FFT*. Hier sind also mehr DFTs für kürzere Segmente zu berechnen, was aber in der $O$-Notation immer noch eine Laufzeitkomplexität von $O(n\log(n))$ ergibt.

Wie würde die Cooley-Tukey-FFT zu einer $\mathrm{DFT}_4$ aussehen? Nun, wenn wir die DFT wieder als Matrix-Vektor-Multiplikation auffassen, dann stellt sich heraus, dass sich die Transformationsmatrix $V_4$ mit der oben beschriebenen Berechnungsvorschrift in vier dünn besetzte Matrizen faktorisieren lässt:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & 1 \\ 1 & \cdot & -1 & \cdot \\ \cdot & 1 & \cdot & -1 \end{pmatrix}}_{=\mathrm{DFT}_2\otimes I_2 := M_4} \underbrace{\begin{pmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & i \end{pmatrix}}_{:=M_3} \underbrace{\begin{pmatrix} 1 & 1 & \cdot & \cdot \\ 1 & -1 & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & 1 & -1 \end{pmatrix}}_{=I_2\otimes\mathrm{DFT}_2 := M_2} \underbrace{\begin{pmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \end{pmatrix}}_{:=M_1}.$$

Und damit gilt für ein Eingabesignal $x \in l(\mathbb{Z}_4)$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = V_4 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = M_4M_3M_2M_1 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}.$$

Diese Berechnung ist in Abbildung 1.22 schematisch dargestellt. Zur Erklärung: Die Matrix $M_1$

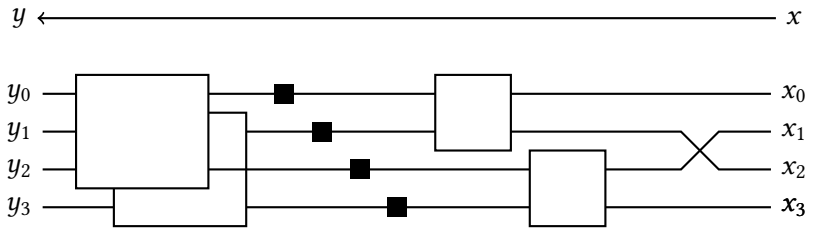


Abbildung 1.22.: Datenflussdiagramm zur Cooley-Tukey-FFT der Ordnung 4. Man beachte, dass die Daten von rechts nach links fließen. Die weißen Kästchen symbolisieren eine DFT der Ordnung 2, die schwarzen Kästchen eine Skalierung der Daten durch $\omega^j$.

bewirkt die Aufteilung des Eingabevektors in gerade und ungerade Indizes. Die Matrix $M_2$ führt anschließend jeweils eine $\mathrm{DFT}_2$ auf den Teilvektoren durch. Danach erfolgt eine Skalierung der Daten durch $M_3$ (dies entspricht dem Korrekturterm $\omega^j$). Abhängig von der Wahl von $j$ müssen die beiden Vektoren nun wieder zu einem großen Vektor zusammengesetzt werden, was durch $M_4$ geschieht. Unter der Annahme, dass Multiplikationen mit 1 und $-1$ sowie Additionen mit 0 wegoptimiert werden können, benötigt dieser ganze Vorgang insgesamt nur drei Additionen und vier Multiplikationen, wohingegen beim naiven Ansatz über die Transformationsmatrix $V_4$ zwölf Additionen und vier Multiplikationen auszuführen sind.

Neben der Cooley-Tukey-FFT gibt es u.a. noch die

- *Prime-factor FFT* (für den Fall, dass $n = km$ und $\gcd(k, m) = 1$),
- *Rader FFT* (falls $n$ eine Primzahl ist),
- *Bluestein FFT*.

### 1.6.4. Fourier und Bilder

Dank der FFT lassen sich digitale Filter schnell und effizient am Rechner implementieren. Dies kann man sich z.B. bei der Bildkomprimierung zunutze machen. Die Idee dabei ist, dass man gewisse hochfrequente Anteile eines Bildes (feine Texturen, Details usw.) zu einem gewissen Grad aus dem Bild entfernen kann, ohne dass dies vom menschlichen wahrgenommen wird. So muss weniger Information im Bild gespeichert werden.

Das gegebene Bild wird hierzu in gleichgroße Blöcke (z.B. der Größe $8\times8$ zur Verarbeitung mit Grafikkarten) aufgeteilt, welche nun alle separat Fourier-transformiert werden. Dieser Vorgang lässt sich hervorragend parallelisieren, da zwischen den Blöcken keinerlei Abhängigkeiten bestehen. Anschließend wird auf jeden dieser Blöcke ein Tiefpassfilter (bspw. eine diskretisierte Variante eines Binomalfilters) angewendet, um so nur noch die groben Informationen des Bildes zu behalten. Auch dieser Vorgang lässt sich über die FFT ausgesprochen schnell durchführen. Abhängig von den Eigenschaften des verwendeten Tiefpassfilters und der gewählten Blockgröße wird das Bild nun mehr oder weniger stark komprimiert worden sein.

Um das resultierende Bild betrachten zu können, muss die Inverse Fourier-Transformation auf jedem der Blöcke ausgeführt werden, da die Bildinformationen ja im Frequenzbereich vorliegen und erst noch in der Ortsbereich überführt werden müssen. Je nach Kompressionsrate des Bildes kann verstärkt Artefaktbildung (Blöcke, Ringe usw.) aufgetreten sein, da das hier beschriebene Verfahren nicht verlustfrei arbeitet.

Dies ist auch die grobe Idee hinter dem Kompressionsverfahren, welches bei JPEG zum Einsatz kommt. Im Gegensatz zur Fourier-Transformation wird bei JPEG auf die *Diskrete Cosinus-Transformation* zurückgegriffen. Diese liefert nur reellwertige Ergebnisse und eignet sich daher wesentlich besser für die Verarbeitung mit modernen Rechnern. Die bereits komprimierten Daten des Bildes können bei JPEG z.B. durch das sog. *Huffman-Encoding* nochmals komprimiert werden, und zwar verlustfrei.

**Definition 1.59** (Diskrete Cosinus-Transformation (DCT)) Die DCT bildet einen Vektor $f \in l(\mathbb{Z}_n)$ auf eine Linearkombination an Cosinus-Termen ab. Diese Operation ist insbesondere invertierbar und linear. Von den vier Typen der DCT ist die gebräuchlichste Variante definiert durch

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

Auf Basis der FFT lässt sich auch die DCT effizient in $O(n \log(n))$ berechnen.

# 2. Transformationen

## 2.1. Die Hough-Transformation

Die Hough-Transformation wird verwendet, um Geraden in Bildern zu erkennen. Jede Gerade in einem Bild lässt sich eindeutig durch zwei Parameter codieren:

- Die Richtung der Gerade. Wir suchen dazu nach einem Normalenvektor auf dem Einheitskreis, welcher senkrecht auf die Gerade steht. Jeden Normalenvektor kann man schreiben als
$$v_\theta = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix},$$
d.h. $\theta$ gibt den Winkel zwischen der $x$-Achse und dem Vektor $v_\theta$ an. Da es zu jeder Gerade offensichtlich zwei Normalenvektoren gibt, beschränken wir uns auf $\theta \in [-\pi/2, \pi/2)$, um eine eindeutige Darstellung zu erhalten.
- Der Abstand $c \in \mathbb{R}$ der Gerade zum Ursprung. Damit ist die Länge der Strecke gemeint, die durch den Ursprung geht und die Gerade senkrecht schneidet.

Die Parametrisierung aller Geraden durch einen Punkt $x$ des Bildes ist dann gegeben durch

$$H(x) = \left\{(\theta, c) : v_\theta^\top x = c\right\} \subset \left[-\frac{\pi}{2}, \frac{\pi}{2}\right) =: \mathbb{H}.$$

Wir betrachten nun eine Menge $X \subset \mathbb{R}^2$ von Punkten (z.B. ein Bild) und bilden zu jedem der Punkte $x$ in $X$ die Menge $H(x)$. Wir zählen nun, wie oft ein Paar $(\theta, c)$ insgesamt in den Mengen $H(x)$ aufgetaucht ist, d.h.

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

Nimmt $n_X(\theta, c)$ einen großen Wert an, dann bedeutet dies, dass wir viele Punkte gefunden haben, die in einer Linie liegen, sprich eine Gerade bilden. Uns interessiert dabei nicht, welchen Farb- oder Helligkeitswert die Bildpunkte haben, sondern einfach nur, ob dieser Bildpunkt vorhanden ist oder nicht. Dazu müssen wir das Bild zuerst binarisieren.

**Definition 2.1**

- Ein binarisiertes Bild ist ein Bild, das nur die Werte 0 oder 1 annimmt (Pixel nicht vorhanden oder vorhanden).
- Die Hough-Transformation eines endlichen binarisierten Bildes mit Pixeln $X = \{(x_j, y_j) \in \mathbb{R}^2 : j = 1, \dots, N\}$ ist
$$(H(X))(\theta, c) = n_X(\theta, c), \qquad (\theta, c) \in \mathbb{H}.$$

Da die meisten Geraden keinen Punkt treffen werden und $H(X)$ damit fast überall 0 ist, zerlegt man $\mathbb{H}$ normalerweise in kleinere Bereiche $\mathbb{H}_{jk}$, von denen man glaubt, dass sie viele Geraden enthalten werden. Dabei ist $\mathbb{H}_{jk} := \Theta_j \times C_k$ mit $j,k = 1,\ldots,M,M'$. $\Theta_j$ ist eine Partition von $[-\pi/2,\pi/2)$ und $C_k$ ist eine Zerlegung eines (hinreichend großen Teilbereichs) von $\mathbb{R}$. Damit lässt sich festlegen, in welchen Bereichen des Bildes man nach welchen Geraden suchen will.

**Bemerkung 2.2** (Pseudo-Algorithmus zur Diskreten Hough-Transformation)

1. Gegeben ist ein (beliebiger) Punkt $p = (x,y)$ des binarisierten Bildes. Wir wollen feststellen, wie viele Geraden durch $p$ verlaufen.
2. Initialisiere die Matrix $\mathbf{H}$ mit $\mathbf{0}_{M\times M'}$. In dieser Matrix wird die Häufigkeit codiert, mit der man für ein gegebenes Paar $(\theta,c)$ einen Punkt des Bildes getroffen hat.
3. Für $j = 1,\ldots,M$:
   a) Wähle einen Mittelpunkt $\theta_j$ der Partition $\Theta_j$.
   b) Bestimme einen Index $k$ mit $1 \le k \le M'$ so, dass
   $$x\cos(\theta_j) + y\cos(\theta_j) \in C_k.$$
   D.h., bestimme die Teilbereiche $C_k$ von $\mathbb{R}$, durch die die Gerade verläuft.
   c) Erhöhe die Anzahl für das entsprechende Paar $(\theta_j,k)$ in der Matrix $\mathbf{H}$ um 1: $H_{jk} \leftarrow H_{jk} + 1$.
4. Ergebnis: $\mathbf{H}$ ist die diskrete Hough-Transformation.

Die Matrix $\mathbf{H}$ kann man nun wiederum als Bild darstellen, indem man die Einträge der Matrix (die ja Häufigkeiten sind) einfach als Helligkeitswert interpretiert. Die $x$-Achse des Bildes gibt den Winkel $\theta$ der Gerade an, und die $y$-Achse den Abstand $c$ der Geraden zum Nullpunkt. Ist das Bild an einer Stelle $(\theta,c)$ hell, so wissen wir, dass die durch $(\theta,c)$ codierte Gerade mit hoher Wahrscheinlichkeit eine Gerade in unserem Ursprungsbild darstellt. Eine andere Möglichkeit, die Ergebnisse der Hough-Transformation zu veranschaulichen, wäre ein Histogramm zu erstellen, bei dem nach rechts z.B. alle Möglichen Paare an Winkel und Abstand aufgetragen werden und nach oben die Zahl der Punkte, durch die diese Geraden gehen.

**Bemerkung 2.3** (Vor- und Nachteile der Hough-Transformation)

- Es werden Kanten erkannt, die in den meisten Fällen eigentlich gar keine Kanten sind, z.B. die Ränder des Bildes. Dieser Fehler ist aber leicht zu korrigieren.
- Die Kanten werden nicht lokalisiert. Wir wissen zwar, durch welche Punkte eine Kante im Originalbild verläuft. Wir wissen aber nicht, wo eine Kante tatsächlich beginnt und wo sie endet. Dies kann auch ein Vorteil sein, da man so z.B. verdeckte Kanten wieder rekonstruieren kann.
- Bevor die Hough-Transformation auf ein Bild angewendet werden kann, muss dieses binarisiert werden. D.h. es muss festgelegt werden, welche Pixel des Bildes als potentielle Kanten betrachtet werden und welche nicht. Dies geschieht meistens durch Schwellwerte, welche aber durch Willkür oder Ausprobieren zustande kommen.

- Mithilfe der Hough-Transformation lassen sich nicht nur Geraden finden, sondern theoretisch alle Kurven, die man irgendwie parametrisieren kann. Z.B. kann man Kreise eindeutig durch einen Mittelpunkt und Radius darstellen und dies ermöglicht es, mit der Hough-Transformation Kreise im Bild zu finden. Das Problem dabei ist nur, dass mit steigender Zahl an Freiheitsgraden der Aufwand der Transformation *exponentiell* steigt. Je komplexer also die Formen sind, nach denen man sucht, desto länger dauert die Transformation. In solchen Fällen hilft es dann nur, einen oder mehrere der Freiheitsgrade zu eliminieren, indem man z.B. für Kreise einen festen Radius oder Mittelpunkt einstellt.

Summa summarum: Die Hough-Transformation ist vor allem dann nützlich, wenn man unterbrochene Kanten im Bild finden will oder das Bild nur durch wenige Kanten dominiert wird.

# Appendices

# A. Satz von Tonelli

Der Satz von Fubini-Tonelli ist eine Verallgemeinerung des Satzes von Fubini.

Die Aussage hier ist, dass wir für integrierbare Funktionen (was also für Funktionen aus Signalräumen immer gilt) die Reihenfolge der Integralauswertungen vertauschen können.

Formal:

Sei $f(x,y)$ eine reelle[1] meßbare[2] Funktion.

Falls eines der Integrale

$$\int_{\Omega_2}\int_{\Omega_1} |f(x,y)|\mathrm{d}\mu_1(x)\mathrm{d}\mu_2(y),$$
$$\int_{\Omega_1}\int_{\Omega_2} |f(x,y)|\mathrm{d}\mu_2(y)\mathrm{d}\mu_1(x)$$

existiert, dann existiert auch das andere und $f(x,y)$ ist bezüglich des Produktmaßes integrierbar und es gilt:

$$\int_{\Omega_1\times\Omega_2} f\mathrm{d}(\mu_1\otimes\mu_2) = \int_{\Omega_2}\int_{\Omega_1} f(x,y)\mathrm{d}\mu_1(x)\mathrm{d}\mu_2(y) = \int_{\Omega_1}\int_{\Omega_2} f(x,y)\mathrm{d}\mu_2(y)\mathrm{d}\mu_1(x)$$

# B. Dichtheit von $L_p$-Räumen

Wir zeigen nun, dass speziell für unsere Zwecke gilt, dass $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ dicht in $L_2(\mathbb{R})$ liegt.

*Beweis.* Sei $f \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$.

Dann ist $f$ auch aus $L_1(\mathbb{R})$ (da es im Schnitt liegt). Das Lemma von *Riemann-Lebesgue* liefert uns, dass $f$ dann im Unendlichen verschwindet. Bei integrablen Funktionen (also Funktionen aus unseren Signalräumen) ist die Fourier-Transformation außerdem stetig. D.h., $f$ ist eine $C_0$-Funktion.

Es gilt, dass $C_0$ dicht in $L_2$ und dicht in $L_1$ liegt[3]. Da dies für beide Räume gilt, so liegt $C_0$ auch dicht in $L_1 \cap L_2$.

Mit den Ergebnissen, dass $C_0 \subset L_2$ dicht, $C_0 \subset L_1 \cap L_2$ dicht und $L_1 \cap L_2 \subset L_2$ ist, so folgt, dass $L_1 \cap L_2$ dicht in $L_2$ liegt. □

# C. Sinus Cardinalis

## C.1. Integrierbarkeit

Es wird gezeigt, dass die sinc-Funktion nicht integrierbar ist.

[1] Da der Körper $\mathbb{C}$ ein zweidimensionaler $\mathbb{R}$-Vektorraum ist (reelle und komplexe Ebene), gilt dies auch für komplexwertige Funktionen.

[2] Dafür braucht man natürlich geeignete $\sigma$-endliche Maßräume, die wir hier mal als gegeben hinnehmen, da unsere Signalräume die geforderten Eigenschaften erfüllen müssten. Es ist mir aber deutlich zu anstrengend dies nachzuweisen, und es interessiert anscheinend sowieso niemanden.

[3] Dies ist ein Resultat, welches ich im Netz gefunden habe und hier einfach so verwende.

*Beweis.* Wir stellen zunächst fest, dass der sinc symmetrisch um die $y$-Achse ist. Außerdem besitzt er seine Nullstellen nur an den ganzzahligen Werten. Dies rechtfertigt

$$\|\operatorname{sinc}\|_1 = \int_{\mathbb{R}} |\operatorname{sinc}(x)| \,\mathrm{d}x = 2\int_0^\infty |\operatorname{sinc}(x)| \,\mathrm{d}x = 2\sum_{k=0}^{\infty} \int_k^{k+1} |\operatorname{sinc}(x)| \,\mathrm{d}x.$$

Wir schätzen das Integral

$$\begin{aligned}\int_k^{k+1} |\operatorname{sinc}(x)| \,\mathrm{d}x &= \int_k^{k+1} \left|\frac{\sin(\pi x)}{\pi x}\right| \mathrm{d}x = \int_k^{k+1} \frac{|\sin(\pi x)|}{\pi x} \,\mathrm{d}x \\ &\geq \int_k^{k+1} \frac{|\sin(\pi x)|}{\pi(k+1)} \,\mathrm{d}x\end{aligned}$$

nach unten ab und erhalten mit der Substitution $t \coloneqq \pi x$, $\mathrm{d}t = \pi \,\mathrm{d}x$

$$\begin{aligned}\int_k^{k+1} \frac{|\sin(\pi x)|}{\pi(k+1)} \,\mathrm{d}x &= \frac{1}{\pi^2(k+1)} \int_{k\pi}^{(k+1)\pi} |\sin(t)| \,\mathrm{d}t \\ &= \frac{1}{\pi^2(k+1)} \Big(-\cos(x)\operatorname{sgn}(\sin(x))\Big)\Big|_{t=k\pi}^{(k+1)\pi} \\ &= \frac{1}{\pi^2(k+1)} \underbrace{(\cos(k\pi) + \cos((k+1)\pi))}_{=1+1=2} \\ &= \frac{2}{\pi^2(k+1)}.\end{aligned}$$

Die Reihe

$$\sum_{k=0}^{\infty} \frac{2}{\pi^2(k+1)} = \frac{2}{\pi^2} \sum_{k=1}^{\infty} \frac{1}{k} \not< \infty$$

divergiert und damit ist auch $\|\operatorname{sinc}\|_1 \not< \infty$, also $\operatorname{sinc} \notin L_1(\mathbb{R})$. □

Auf ähnliche Weise lässt sich übrigens zeigen, dass $\operatorname{sinc} \in L_2(\mathbb{R})$.

## C.2. Flächeninhalt

Gesucht ist der Wert des Integrals

$$\int_{\mathbb{R}} \operatorname{sinc}(x) \,\mathrm{d}x,$$

also der Flächeninhalt, der zwischen der $x$-Achse und der sinc-Funktion eingeschlossen wird. Dazu machen wir uns zuerst klar, dass für beliebige $x \in \mathbb{R} \setminus \{0\}$ gilt

$$\int_0^\infty e^{-xt} \,\mathrm{d}t = \left(-\frac{e^{-xt}}{x}\right)\Big|_{t=0}^{\infty} = 0 - \left(-\frac{1}{x}\right) = \frac{1}{x}.$$

Danach verwenden wir den Satz von Fubini, um zu erhalten:

$$\int_{\mathbb{R}} \frac{\sin(x)}{x} \,\mathrm{d}x = 2\int_0^\infty \frac{\sin(x)}{x} \,\mathrm{d}x = 2\int_0^\infty \int_0^\infty e^{-xt} \sin(x) \,\mathrm{d}x \,\mathrm{d}t.$$

Partielle Integration liefert uns

$$\begin{aligned}
\int_0^\infty e^{-xt}\sin(x)\,\mathrm{d}x &= \left(-\frac{e^{-xt}}{t}\sin(x)\right)\Big|_{x=0}^{\infty} - \int_0^\infty -\frac{e^{-xt}}{t}\cos(x)\,\mathrm{d}x \\
&= 0 + 0 + \int_0^\infty \frac{e^{-xt}}{t}\cdot\frac{e^{ix}+e^{-ix}}{2}\,\mathrm{d}x \\
&= \frac{1}{2t}\left(\int_0^\infty e^{-x(t-i)}\,\mathrm{d}x + \int_0^\infty e^{-x(t+i)}\,\mathrm{d}x\right) \\
&= \frac{1}{2t}\left(-\frac{e^{-x(t-i)}}{t-i}\Big|_{x=0}^{\infty} + \frac{e^{-x(t-i)}}{t+i}\Big|_{x=0}^{\infty}\right) \\
&= \frac{1}{2t}\left(0 + \frac{1}{t-i} + 0 + \frac{1}{t+i}\right) = \frac{1}{2t}\cdot\frac{2t}{t^2+1} \\
&= \frac{1}{t^2+1}.
\end{aligned}$$

Folglich ist

$$\int_{\mathbb{R}} \frac{\sin(x)}{x}\,\mathrm{d}x = 2\int_0^\infty \frac{1}{t^2+1}\,\mathrm{d}t = 2\arctan(t)\Big|_{t=0}^{\infty} = 2\left(\frac{\pi}{2}-0\right) = \pi.$$

Schließlich erhält man

$$\int_{\mathbb{R}} \operatorname{sinc}(x)\,\mathrm{d}x = \int_{\mathbb{R}} \frac{\sin(\pi x)}{\pi x}\,\mathrm{d}x = \frac{1}{\pi}\int_{\mathbb{R}} \frac{\sin(t)}{t}\,\mathrm{d}t = \frac{1}{\pi}\cdot\pi = 1$$

mit Substitution von $t \coloneqq \pi x$, $\mathrm{d}t = \pi\,\mathrm{d}x$.

# D. Faltungseigenschaften

Wir zeigen die folgenden Faltungseigenschaften nur für Funktionen, die Beweise lassen sich ganz analog im voll- bzw semidiskreten Fall führen.

Seien also im Folgenden $f, g, h \in L_1(\mathbb{R})$.

## D.1. Kommutativität

Es gilt:

$$(f * g) = \int_{\mathbb{R}} f(\cdot - t)g(t)\mathrm{dt} = \int_{\mathbb{R}} f(t)g(\cdot - t)\mathrm{dt} = (g * f) \tag{1}$$

Damit ist die Kommutativität auch schon bewiesen.

## D.2. Assoziativität

Durch Umformen erreichen wir, dass gilt:

$$\begin{aligned}(f * (g * h))(x) &= \left(f * \int_{\mathbb{R}} g(\cdot - u)h(u)\mathrm{du}\right)(x) \\ &= \int_{\mathbb{R}} f(x-t) \int_{\mathbb{R}} g(t-u)h(u)\mathrm{dudt} \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} f(x-t)g(t-u)h(u)\mathrm{dudt} \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} f(x-t-u)g(t)h(u)\mathrm{dudt} \\ &= \left(\int_{\mathbb{R}} f(\cdot - t)g(t)\mathrm{dt} * h\right)(x) \\ &= ((f * g) * h)(x)\end{aligned} \tag{2}$$

## D.3. Distributivität

Die Distributivität folgt sehr schnell durch Umformen:

$$\begin{aligned}(f * (g + h)) &= \int_{\mathbb{R}} f(\cdot - t)((g + h)(t))\mathrm{dt} \\ &= \int_{\mathbb{R}} f(\cdot - t)g(t) + f(\cdot - t)h(t)\mathrm{dt} \\ &= \int_{\mathbb{R}} f(\cdot - t)g(t)\mathrm{dt} + \int_{\mathbb{R}} f(\cdot - t)h(t) \\ &= f * g + f * h\end{aligned} \tag{3}$$