Datenkompression:

Teilbandcodierung

H. Fernau

email: fernau@uni-trier.de

WiSe 2008/09

Universität Trier

Teilbandcodierung—Eine Überleitung

Beobachte Beispielfolge $\{x_n\}$:

$$10 \quad 14 \quad 10 \quad 12 \quad 14 \quad 8 \quad 14 \quad 12 \quad 10 \quad 8 \quad 10 \quad 12$$

Anwenden eines DPCM-Schemas $\rightsquigarrow$ codiere Differenzenfolge:

$$10 \quad 4 \quad -4 \quad 2 \quad 2 \quad -6 \quad 6 \quad -2 \quad -2 \quad -2 \quad 2 \quad 2$$

Abgesehen vom ersten Wert liegen alle Differenzen zwischen -6 und $+6$.
Ein m-Bit Gleichquantisierer würde eine Schrittweite $\Delta = 12/2^m$ benutzen $\rightsquigarrow$ maximaler Quantisierfehler von $\Delta/2 = 6/2^m$.

Dezimierung Statt dieser Differenzen kann ohne Informationsverlust auch $z_n = \frac{x_n - x_{n-1}}{2}$ übertragen. Das halbiert den Quantisierfehler.

Betrachte stattdessen $y_n = \frac{x_n + x_{n-1}}{2}$ (Folge der Mittelwerte) $\rightsquigarrow$ Differenzenfolge der y_n:

$$10 \quad 2 \quad 0 \quad -1 \quad 2 \quad -2 \quad 0 \quad 2 \quad -2 \quad -2 \quad 0 \quad 2$$

$\rightsquigarrow$ maximaler Quantisierfehler für einen m-Bit Gleichquantisierer von $2/2^m$.
Also: die Mittelwertfolge ist „glatter" als die Originalwertfolge.

Werden y_n und z_n getrennt übertragen, so lässt sich hieraus durch Addition die eigentlich zu übertragende Folge x_n rekonstruieren.

Beachte überdies, dass sich alle x_n-Werte aufgrund von

$$\begin{aligned} x_{2n-1} &= y_{2n} - z_{2n} \text{ und} \\ x_{2n} &= y_{2n} + z_{2n} \end{aligned}$$

wiedergewinnen lassen lediglich durch Übertragen der „halben" Folgen y_{2n} und z_{2n}, und auch die hierfür zu betrachtende Differenzenfolge $y_{2n} - y_{2n-2}$ verhält sich angenehm.

Grundidee der Teilbandcodierung: eine gegebene Folge wird durch sog. *Filter* in *(Teil-)Bänder* genannte Bestandteile zerlegt, die dann (evtl. mit jeweils angepassten unterschiedlichen Verfahren codiert) getrennt übertragen werden.

Im Beispiel enthielt das Band der Mittelwertsfolge „langfristige Informationen", also *niederfrequente* Anteile, und das Band der Differenzenfolge *hochfrequente* Anteile.
Niederfrequenzbänder lassen sich oft gut mit DPCM-Technik komprimieren.
Ein Filter, der vornehmlich niederfrequente Anteile „durchlässt", wird *Tiefpass(filter)* (engl.: low pass) genannt.
Entsprechend heißt ein Filter, den fast ausschließlich hochfrequente Anteile „passieren", *Hochpass(filter)* (engl.: high pass).
Anstelle von Tief- und Hochpassfiltern spricht man auch von *Glättungs- und Differenzierfiltern*.

Frequenzfilter

Verallgemeinern wir obiges Beispiel:

Wir betrachten Filter, die eine *Eingabefolge* $\{x_n\}$ in eine *Ausgabefolge* $\{y_n\}$ vermöge

$$y_n = \sum_{i=0}^{N-1} a_i x_{n-i} + \sum_{i=1}^{M} b_i y_{n-i}$$

überführen. Auf die spezielle *Impuls-Folge*

$$x_n = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

„antwortet" ein Filter mit der *Impulsantwort* $\{h_n\}$. Sind alle b_i Null, so handelt es sich um einen Filter mit *endlicher Impulsantwort* (FIR: engl.: finite impulse response) mit höchstens N Nicht-Null-Werten $h_0, \ldots, h_{N-1}$, andernfalls um einen mit *unendlicher Impulsantwort* (IIR: engl.: infinite impulse response).*

*Wir beschäftigen uns hier nur mit eindimensionalen Signalen.

Ein Beispiel

Man errechnet als Impulsantwort für den durch die Nicht-Null-Werte $a_0 = 1,25$ und $a_1 = 0,5$ spezifizierten Filter:

$$\begin{aligned} h_0 &= a_0 x_0 + a_1 x_{-1} = 1,25 \\ h_1 &= a_0 x_1 + a_1 x_0 = 0,5 \\ h_n &= 0 \text{ sonst} \end{aligned}$$

Für den durch die Nicht-Null-Werte $a_0 = 1$ und $b_1 = 0,9$ spezifizierten Filter bekommt man als Impulsantwort:

$$\begin{aligned} h_0 &= a_0 x_0 + b_1 h_{-1} = 1(1) + 0,9(0) = 1 \\ h_1 &= a_0 x_1 + b_1 h_0 = 1(0) + 0,9(1) = 0,9 \\ h_2 &= a_0 x_2 + b_1 h_1 = 1(0) + 0,9(0,9) = 0,81 \\ &\vdots \quad\quad \vdots \\ h_n &= (0.9)^n \end{aligned}$$

Hoch- und Tiefpass im Beispiel

Ähnlich erhält man als Impulsantwort h_n bzw. h'_n des „Mittelwertfilters y_n" bzw. des „Differenzfilters z_n" aus dem früheren Beispiel:

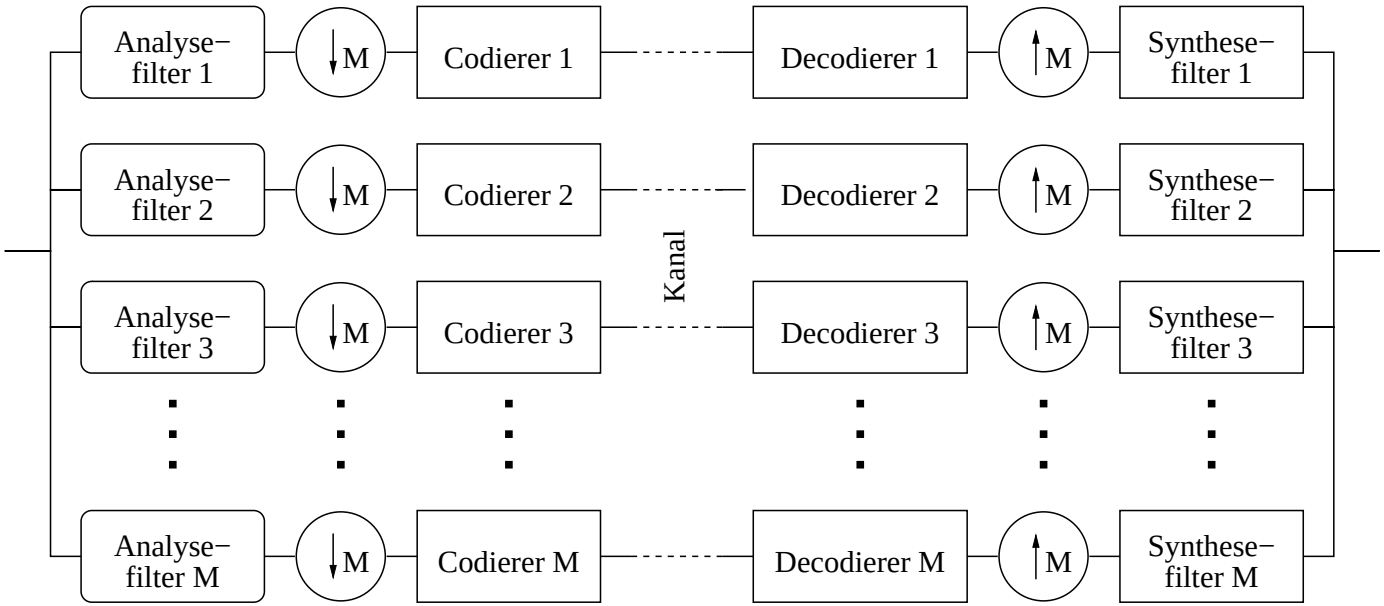
$$h_n = \begin{cases} 0,5 & n = 0 \\ 0,5 & n = 1 \\ 0 & \text{sonst} \end{cases} \quad h'_n = \begin{cases} 0,5 & n = 0 \\ -0,5 & n = 1 \\ 0 & \text{sonst} \end{cases}$$

Der allgemeine Fall

Allgemein legt die Impulsantwort die Filterfunktion in folgender Weise fest:

$$y_n = \sum_{k \geq 0} h_k x_{n-k}$$

Blockdiagramm zur Teilbandcodierung



Grundschritte beim Teilbandcodierverfahren:

Auf der *Codierer*seite wird der einkommende Signalstrom zunächst durch eine Filterbank in M (sich evtl. überlappende) Teilbänder zerlegt;
jeder gefilterte Signalstrom wird dann „dezimiert", d.h., nur jedes M-te Datum wird dem eigentlichen Codierer weiter geleitet. Anstelle von *Dezimierung* spricht man auch von *Unterabtastung*.
Um unterschiedliche Charakteristika verschiedener Bänder desselben Signals auszunutzen, werden die für die Einzelbänder benutzten Codierverfahren unterschiedlich sein; typischerweise sind niederfrequente Bänder DPCM-quantisiert und höherfrequente skalar- oder vektorquantisiert.
Im *Decodierer* müssen umgekehrt die rekonstruierten Signale der verschiedenen Bänder wieder zusammengesetzt werden. Die Synthesefilter sind natürlich im Allgemeinen von den Analysefiltern verschieden.

Ein Beispiel„bild"

10	14	10	12	14	8	14	12
10	12	8	12	10	6	10	12
12	10	8	6	8	10	12	14
8	6	4	6	4	6	8	10
14	12	10	8	6	4	6	8
12	8	12	10	6	6	6	6
12	10	6	6	6	6	6	6
6	6	6	6	6	6	6	6

Ein Beispielbild — gefilterte und dezimierte Zeilen

Dezimierte Tiefpassausgabe				Dezimierte Hochpassausgabe			
5	12	13	11	5	-2	1	3
5	10	11	8	5	-2	1	2
6	9	7	11	6	-1	1	1
4	5	5	7	4	-1	-1	1
7	11	7	5	7	-1	-1	1
6	10	8	6	6	-2	-2	0
6	8	6	6	6	-2	0	0
3	6	6	6	3	0	0	0

LL-Bereich				HL-Bereich			
2,5	6	6,5	5,5	2,5	-1	0,5	1,5
5,5	9,5	9	9,5	5,5	-1,5	1	1,5
5,5	8	6	6	5,5	-1	-1	1
6	9	7	6	6	0	0	0
LH-Bereich				HH-Bereich			
2,5	6	6,5	5,5	2,5	-1	0,5	1,5
0,5	-0,5	-2	1,5	0,5	0,5	0	-0,5
1,5	3	1	-1	1,5	0	0	0
0	1	-1	0	0	0	1	0

Was beobachten wir ?

üblich im Falle von Bildern: zunächst wird zeilenweise, dann spaltenweise gefiltert.
Links und oben wird dabei eine Spalte bzw. Zeile mit Nullen angesetzt. Dagegen bleiben die rechte Spalte und die unterste Zeile unbeachtet.
Leider sind in diesem Mini-Beispiel die Entstellungen an den Rändern noch überdeutlich.
Die Hauptinformation des Bildes ist im niederfrequenten (oben links dargestellten) Teilbild vorhanden und „fast keine" Information im hochfrequenten Teilbild (unten rechts).
„Gute" Filter konzentrieren möglichst viel Energie im niederfrequenten Teilbild.
Um die verschiedenartigen Dezimierungskaskaden zu verdeutlichen, werden die Abkürzungen LL, HL, LH und HH verwendet. (low / high)

Filterbänke sind oft kaskadenartig aus Paaren von Tief- und Hochpassfiltern aufgebaut.
Beliebt sind *Spiegelfilter* (engl.: quadrature mirror filters QMF); ist $\{h_n\}$ die Impulsantwort des Tiefpassfilters, so ist $\{(-1)^n h_{N-1-n}\}$ die Impulsantwort des Hochpassfilters.
Die Filter von Johnston und Smith-Barnwell sind ebenfalls von dieser Gestalt.

Die **Quantisierung der einzelnen Teilbänder** wird beeinflusst durch Wahl der Filterbank.
Sollen pro Datum höchstens R Bits übertragen werden, und sei $\sigma^2_{y_k}$ die Eingabesignalvarianz des k-ten Quantisierers ($1 \leq k \leq M$), so arbeitet folgendes Verfahren gut für die Verteilung der zur Verfügung stehenden R Bits auf die Teilbänder:

1. Berechne $S_k = \sigma^2_{y_k}$ und setze $R_k = 0$ für $1 \leq k \leq M$.

2. Wähle ein k', so dass $S_{k'}$ maximal für $1 \leq k' \leq M$.

3. Inkrementiere $R_{k'}$; dividiere $S_{k'}$ durch zwei.

4. Dekrementiere R. Falls $R > 0$, gehe zu Schritt 2.

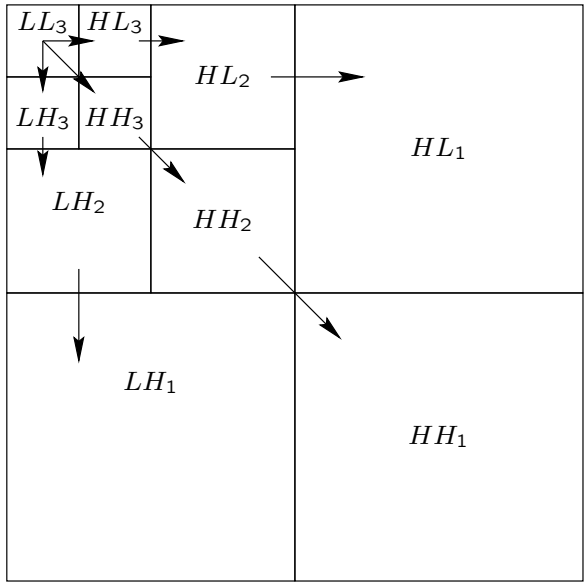
Nach Durchlauf des Verfahrens sollten R_k Bits für das k-te Band genommen werden.

Shapiros EZW-Algorithmus

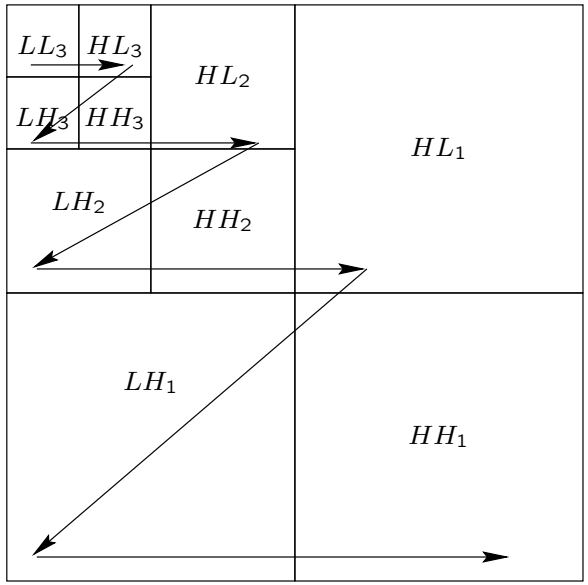
Eine andere, adaptive Idee zur Bitverteilung bei der Bildkompression mit Teilbandverfahren ist im *EZW-Algorithmus** enthalten. Bekanntermaßen wird ein Bild zeilen- und spaltenweise mit einem Tief- bzw. Hochpassfilter in vier Bandbereiche zerlegt.
Für den Low-Low-Bereich, in dem sich ja die Energie bei geeigneter Filterwahl konzentrieren sollte, kann man nun diese Zerlegung wiederholen, was bei dreimaliger Wiederholung zu folgender Aufteilung des „gefilterten und dezimierten" Bildes führt.

*EZW steht für engl.: „Embedded Zerotree Wavelet"; im Ggs. zum Namen ist das Grundprinzip des Vorgehens nicht auf Wavelets beschränkt.

Abkömmlingsbaum à la Shapiro



Abtasten von Filterkoeffizienten à la Shapiro



Ein dreimal wiederholt gefiltertes und dezimiertes Beispiel-„Bild"

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

Das Beispiel durchgerechnet...

Betrachte die im Beispiel gegebenen gefilterten und dezimierten Werte.
Der betragsmäßig größte Wert ist 63.
Als initialen Schwellwert T_0 wählen wir daher dessen gerundete Hälfte, d. i. $T_0 = 32$.
Im ersten Hauptdurchgang wird nun das Bild abgetastet.
Für Teilbilder wird rekursiv dieselbe Abtastreihenfolge gewählt.
In den Hauptdurchgängen wird ein Strom von Symbolen aus { POS, NEG, IZ, ZTR } erzeugt; diese Zeichen sind natürlich mit je zwei Bit codierbar.
POS: der abgetastete Wert ist größer als der aktuelle Schwellwert T_k,
NEG: er ist kleiner als $-T_k$,
IZ bzw. ZTR: er ist im Intervall $[-T_k, T_k]$.
Werte, für die POS oder NEG erzeugt wird, heißen auch *signifikant*, die übrigen *insignifikant*.
ZTR (engl.: zerotree) bedeutet überdies, dass alle „Abkömmlinge" im Sinne des in obigem Bild mit Pfeilen dargestellten Baumes (der im übrigen wieder rekursiv fortgesetzt zu denken ist, s. u. Beispiel) im Intervall $[-T_k, T_k]$ liegen und so einen *Nullbaum* bilden; dahingegen bezeichnet IZ eine *vereinzelte Null* (engl.: isolated zero).

Ergebnisse des ersten Hauptdurchgangs

Der erste Hauptdurchgang ist in der Tabelle unten aufgelistet.
erster Wert 63 $>$ Schwellwert 32 $\leadsto$ POS.
$-34 < -32 \leadsto$ NEG.
$-31 \in [-32, 32]$ (im darunterhängenden Teilbaum liegt aber noch ein „signifikanter Wert'' 47) $\leadsto$ IZ Eintrag im HH_3-Band sowie alle seine Abkömmlinge (in den Bändern HH_2 und HH_1) insignifikant $\leadsto$ bei (3) ZTR
($\leadsto$ es werden keine weiteren Symbole mehr für Einträge aus dem HH_2- und HH_1-Band erzeugt in diesem Hauptdurchgang.)

Jetzt wird das Teilband HL_2 abgetastet.
Bemerke bei (4): die Abkömmlinge des Eintrags 10 sind $\{-12, 7, 6, 1\}$, was durch rekursive Forsetzung des dargestellten Abkömmlingsbaums zu sehen ist, und diese sind insignifikant $\leadsto$ ZTR.
Bemerke bei (5) (innerhalb der Analyse des LH_2-Bands): 14 selbst liegt unterhalb des Schwellwerts, das Abkömmlingsteilband $\{-1, 47, -3, 2\}$ enthält jedoch mit 47 einen signifikanten Wert $\leadsto$ IZ.
Beachte bei (6): das HH_2-Band wurde bereits berücksichtigt $\leadsto$ mach weiter mit Codierung des HL_1-Bands.

Kommentar	Teilband	Wert	Symbol	rekonstruierter Wert
(1)	LL_3	63	POS	48
	HL_3	-34	NEG	-48
(2)	LH_3	-31	IZ	0
(3)	HH_3	23	ZTR	0
	HL_2	49	POS	48
(4)	HL_2	10	ZTR	0
	HL_2	14	ZTR	0
	HL_2	-13	ZTR	0
	LH_2	15	ZTR	0
(5)	LH_2	14	IZ	0
	LH_2	-9	ZTR	0
	LH_2	-7	ZTR	0
(6)	HL_1	7	Z	0
	HL_1	13	Z	0
	HL_1	3	Z	0
	HL_1	4	Z	0
	LH_1	-1	Z	0
(7)	LH_1	47	POS	48
	LH_1	-3	Z	0
	LH_1	-2	Z	0

Die erste Verfeinerung

Koeffizienten-größe	Symbol	Rekonstruktions-größe
63	1	56
34	0	40
49	1	56
47	0	40

für die vier signifikanten Werte (aus dem ersten Hauptdurchgang) wird angegeben, ob sie (dem Betrage nach) im Intervall $[32, 48)$ oder in $[48, 64)$ liegen.

Der nächste Hauptdurchgang

Es werden nur die bislang für insignifikant gehaltenen Werte berücksichtigt; die bislang signifikanten Werte werden hingegen in diesem Durchgang als Nullen (im Sinne der Nullbaumkonstruktion) angesehen.
Als Schwellwert nehmen wir $T_1 = T_0/2 = 16$.
Im zweiten Verfeinerungsdurchgang hingegen werden *alle* aus den bisherigen (beiden) Hauptdurchgängen gewonnenen Werte verfeinert, was also zu einer Intervalleinteilung $[16, 24)$, $[24, 32)$, $[32, 40)$, $[40, 48)$, $[48, 56)$ und $[56, 64)$ führt.

Insbesondere wird so hoffentlich die Bedeutung des Wortes „embedded'' deutlich: Jeder hochauflösendere Code enthält sämtliche niederauflösenden Codes eines Bildes als Präfix; m. a. W., EZW eignet sich sehr gut zur fortschreitenden Bildübertragung.

Filter aus Transformationen

Wie kann man *systematisch* ein Eingabesignal in Frequenzbereiche zerlegen? Eine Möglichkeit ist recht allgemein bekannt: die *Fourier-Entwicklung bzw. -Transformation*. Jede reelle (oder komplexe) Funktion f mit Periode T lässt sich darstellen als:

$$
\begin{aligned}
f(t) &= \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(n\frac{2\pi}{T}t\right) + \sum_{n=1}^{\infty} b_n \sin\left(n\frac{2\pi}{T}t\right) \text{ mit} \\
a_n &= \frac{1}{T}\int_0^T f(t)\cos\left(n\frac{2\pi}{T}t\right)dt, \\
b_n &= \frac{1}{T}\int_0^T f(t)\sin\left(n\frac{2\pi}{T}t\right)dt; \qquad \text{alternativ:} \\
f(t) &= \sum_{-\infty}^{\infty} c_n e^{in\frac{2\pi}{T}t} \text{ mit} \\
c_n &= \frac{1}{T}\int_0^T f(t)e^{-in\frac{2\pi}{T}t}dt
\end{aligned}
$$

Hierbei ist, wie üblich, $i = \sqrt{-1}$.

So erhält man nicht nur eine „Sinusentwicklung" von periodischen Funktionen, sondern auch eine „Sinus-Approximation", indem man die Reihenentwicklung nach endlich vielen Gliedern abbricht* Diese ist wichtig, da —elektrophysikalisch bedingt— „physikalische Filter" keine „reinen" Bandpassfilter sind. Wir können diese immer nur durch Schwingungsfunktionen annähern.

Eine lediglich auf einem Intervall $[0, T]$ definierte Funktion lässt sich leicht „periodisch fortsetzen".†

*Für sog. quadratisch integrierbare Funktionen ist Konvergenz garantiert.

†Betrachtet man den Fall „$T \to \infty$", so erhält man die Fouriertransformierte von f.

Diskrete Fourier-Transformation DFT

Da wir zumeist mit „diskreten Funktionen", sprich Folgen $\{x_n\}$ der Länge N beschäftigen, ist die *diskrete Fourier-Transformation* wichtig:

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{\frac{-i2\pi kn}{N}}$$

$$x_n = \sum_{k=1}^{N-1} c_k e^{\frac{i2\pi kn}{N}}$$

In der folgenden Vorlesung werden wir hierauf zurückkommen.

Wavelets

Ein flexibleres und modernes Instrument zur Signalzerlegung liefern *Wavelets*. Bei ihnen steht sowohl der Zeit- als auch der Frequenzbereich parametrisiert zur Verfügung.

Sehr beliebt sind die sog. *Haar-Wavelets*. Aus der einfachen *Urfunktion* (engl.: mother wavelet)

$$\psi_{0,0}(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \\ 0 & \text{sonst} \end{cases} \quad \text{wird durch}$$

$$\begin{aligned} \psi_{j,k}(x) &= \psi_{0,0}(2^j x - k) \\ &= \begin{cases} 1 & k2^{-j} \leq x < (k+\frac{1}{2})2^{-j} \\ -1 & (k+\frac{1}{2})2^{-j} \leq x < (k+1)2^{-j} \\ 0 & \text{sonst} \end{cases} \end{aligned}$$

eine Schar von Funktionen. j kann als *Streckparameter* (*Frequenzparameter*) und k als *Verschiebeparameter* (*Zeitparameter*) gedeutet werden.

Wavelets — ein einfaches Beispiel

Um eine Idee von der Arbeitsweise von Wavelets zu erhalten, betrachten wir exemplarisch nun als Urfunktion

$$\phi_{0,0}(x) \;=\; \begin{cases} 1 & 0 \le x < 1 \\ 0 & \text{sonst} \end{cases}$$

sowie die erzeugte Schar

$$\phi_{j,k}(x) \;=\; \phi_{0,0}(2^j x - k)\,.$$

Angenommen, wir wollten eine Funktion $f : [0, N] \to \mathbb{R}_+$ approximieren. In erster Näherung setzen wir

$$\begin{aligned} \phi_f^0(t) &= \sum_{k=0}^{N-1} c_{0,k}\phi_{0,k} \quad \text{mit} \\ c_{0,k} &= \int_k^{k+1} f(t)[\phi_{0,k}(t)]dt \end{aligned}$$

Durch Wahl eines höheren Frequenzparameters können wir die Genauigkeit der Approximation steigern:

$$\begin{aligned} \phi_f^j(t) &= \sum_{k=0}^{2^j N-1} c_{j,k}\phi_{j,k} \quad \text{mit} \\ c_{j,k} &= 2^j \int_{k/2^j}^{(k+1)/2^j} f(t)dt\,. \end{aligned}$$

Offensichtlich gilt

$$c_{j-1,k} = 1/2(c_{j,2k} + c_{j,2k+1})\,. \qquad (1)$$

Unser Ziel ist es, Funktionen in Bestandteile (mit evtl. unterschiedlichen Charakteristika) zu zerlegen. Wenn nun ϕ_f^1 die Funktion f genügend approximiert, kann man ϕ_f^0 als niederfrequenten Anteil betrachten und muss dann die Differenz $\phi_f^1 - \phi_f^0$ diskutieren. Es gilt wegen Gleichung (1):

$$\phi_f^1(t) - \phi_f^0(t)$$
$$= \begin{cases} c_{0,k} - c_{1,2k} = -1/2 c_{1,2k} + 1/2 c_{1,2k+1} & k \leq t < k + 1/2 \\ c_{0,k} - c_{1,2k+1} = 1/2 c_{1,2k} - 1/2 c_{1,2k+1} & k + 1/2 \leq t < k + 1 \end{cases}$$

M. a. W., die Differenz lässt sich leicht mit Haar-Wavelets ausdrücken, nämlich:

$$\phi_f^1(t) - \phi_f^0(t) = \underbrace{(-c_{1,2k} + c_{1,2k+1})}_{b_{0,k}} \psi_{0,k}(t) \,.$$

Die $2N$-Punkte Folge $c_{1,k}$ kann so in zwei N-Punkt-Folgen $c_{0,k}$ und $b_{0,k}$ zerlegt werden;
die zweite Folge kann als Faktoren von Wavelets interpretiert werden.

Allgemein lässt sich jede Funktionenschar $\phi_{j,k}$, die gewissen Skalier-, Darstellungs- und Integrabilitätsbedingungen genügt, dafür benutzen, eine zugehörige Wavelet-Familie zu definieren. Aus der Darstellungsbeziehung

$$\phi_{0,0}(t) = \sum h_n \phi_{1,n}(t)$$

gewinnt man die Impulsantwort h_n des Glättungsfilters, denn $\phi_{0,0}$ kann als stetige Form des Impulses gesehen werden, und die Darstellung

$$\psi_{0,0}(t) = \sum (-1)^{N-n-1} h_n \phi_{1,n}(t)$$

des zugehörigen Ur-Wavelets liefert die Impulsantwort des Differenzierfilters. Beliebt sind insbesondere die so zu erhaltenen *Daubechies- und Coiflet-Filter*. Wavelets liefern Spiegelfilter.

Wichtig: die Aufspaltung des Ursignals in Glätte- und Differenzanteil kann man weitertreiben, indem der Glätteanteil rekursiv weiter aufgespalten wird.
Die Differenzanteile m-ter Stufe lassen sich dann durch die Wavelets $\psi_{m,k}$ darstellen.
Die so mögliche rekursive *Multiresolutionsanalyse* ist einer der wesentlichen Vorteile Wavelet-basierter Zeit/Frequenz-Analyse gegenüber dem (älteren) Fourier-Ansatz.