# Medical Image Processing

**6 Fourier and Wavelet Transforms**

Prof. Dr. Marcin Grzegorzek

Research Group for Pattern Recognition
Institute for Vision and Graphics
University of Siegen, Germany

# Table of Contents

# Overview

# Overview

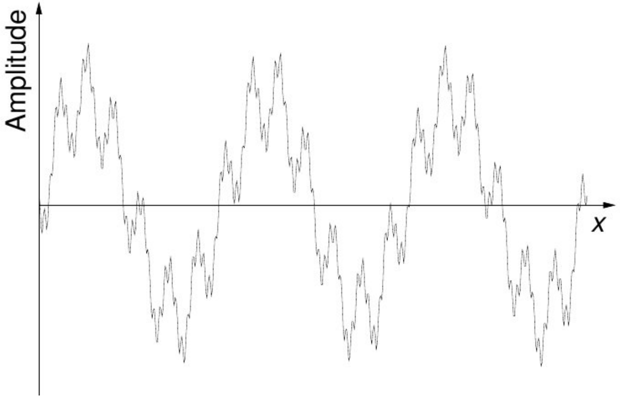
# DFT - Allgemeines



- benannt nach franz. Mathematiker Jean Baptiste Joseph Fourier
- periodische Funktion als Summe von Sinus- und Kosinusfunktionen darstellbar
- Darstellung im Ortsbereich versus Darstellung im Frequenzbereich
- quivalente Darstellungen (ineinander verlustfrei berfhrbar)
- Darstellung im Frequenzbereich
  - niedrige Frequenzen fr groben Funktionsverlauf
  - hohe Frequenzen fr Detailinformationen (etwa abrupte Funktionswertnderungen)

## DFT - Erstes Beispiel

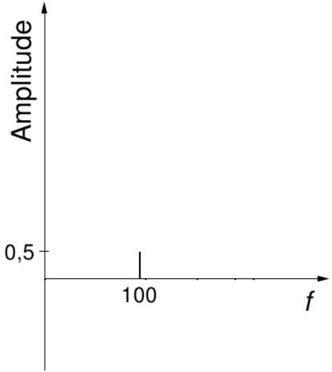
gegeben: $f(x) = 4\sin 3x - 3/2\sin 20x + 1/2\cos 100x$

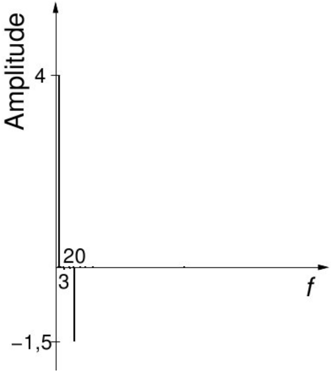


# DFT - Erstes Beispiel



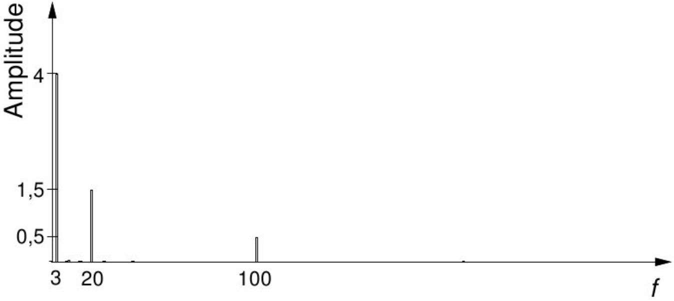
Anteil Kosinus- und Sinusschwingungen:



# DFT - Erstes Beispiel

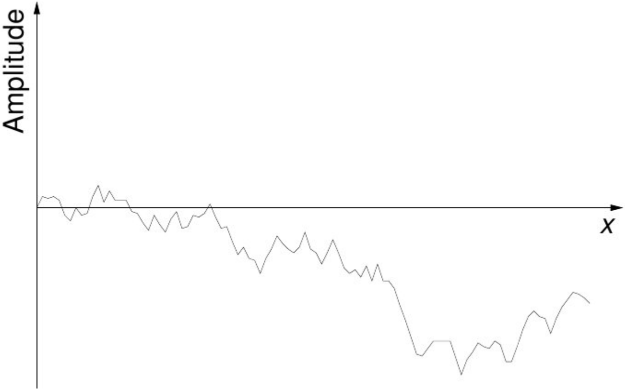
Frequenzspektrum:



# DFT - Zweites Beispiel



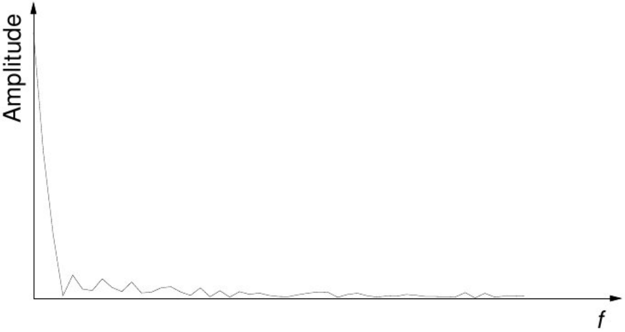
zufällig erzeugte Funktion:



# DFT - Zweites Beispiel



Frequenzspektrum und Kompaktheit:



# DFT - Gedankenexperiment

Darstellung im Orts- und Frequenzbereich:

- Lautsprecher als Tonquelle erzeugt konstantes Gerusch
  $\rightarrow$ Orts- bzw. Zeitbereich
- Eine bestimmte Klaviersaite beginnt zu schwingen, wenn ihre Frequenz im Gerusch enthalten ist. Wenn man die Strke aller Saitenschwingungen aufzeichnet, erhlt man die Darstellung des Signals im Frequenzbereich.
  $\rightarrow$ Frequenzbereich
- Rekonstruktion des Gerusches durch Anregen der entsprechenden Klaviersaiten

# DFT - Anwendungen



## Feature-Normalisierung

- Unterdrckung von Streinflssen innerhalb der Medienobjekte
- Problem: Trennung Nutz- und Strdaten
- Trennung manchmal im Frequenzbereich mglich

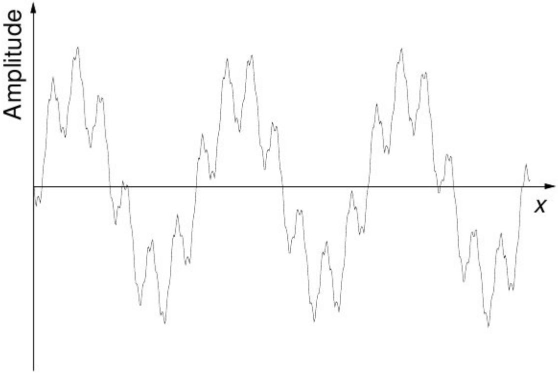
## Feature-Erkennung

- Korrespondenz zwischen zu extrahierenden Eigenschaften und Frequenzen mglich

## Feature-Aufbereitung

- Kompaktheit im Frequenzbereich $\rightarrow$ Minimalitt
- Orthogonalitt der Fourier-Koeffizienten

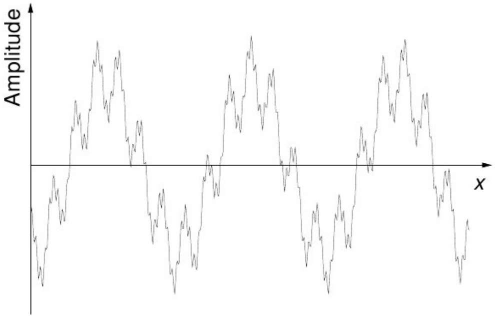
## Beispiel: Entfernung von Frequenz f=100



# DFT - Feature-Erkennung, Translationsinvarianz



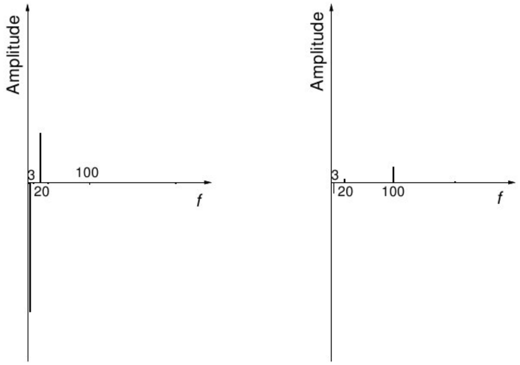
Funktion $f(x) = 4\sin 3x - 3/2\sin 20x + 1/2\cos 100x$ um $\pi/3$ verschoben:



# DFT - Feature-Erkennung, Translationsinvarianz



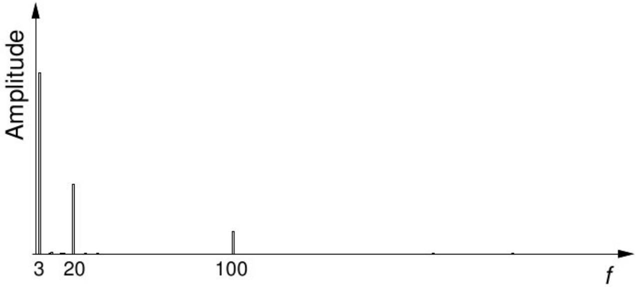
Anteil Kosinus- und Sinusschwingungen:



# DFT - Feature-Erkennung mit Translationsinvarianz

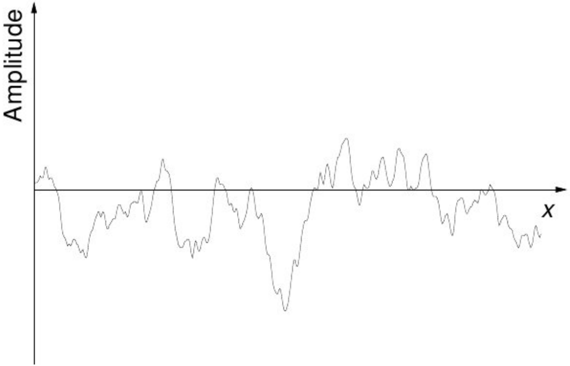


Frequenzspektrum:



# DFT - Feature-Aufbereitung, Minimalitt

Ausgangsfunktion:



# DFT - Feature-Aufbereitung, Minimalitt

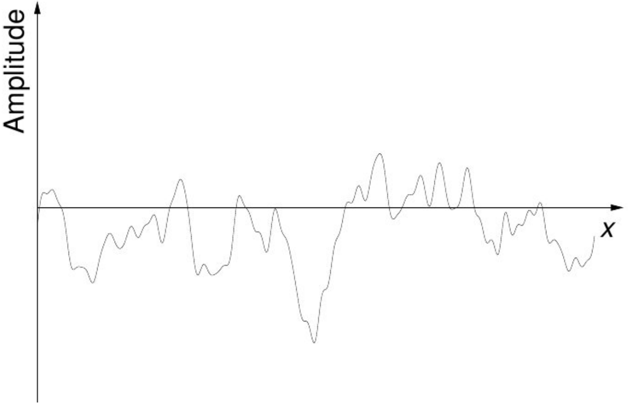


Frequenzspektrum und Abschneiden hoher Frequenzen:

# DFT - Feature-Aufbereitung, Minimalitt



approximierte, d.h. minimierte Funktion:



# DFT - Berechnung

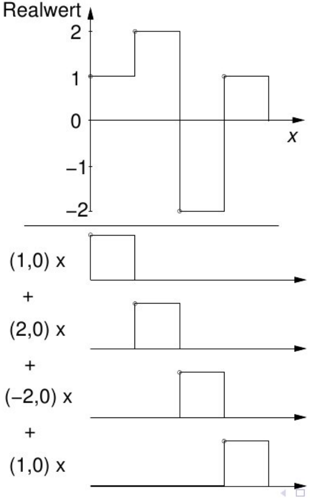
## Erinnerung lineare Algebra und komplexe Zahlen

- diskrete Funktion $f(x)$ als Vektor des komplexen Vektorraums: $f(x) \in D_n^{\mathbb{C}}$
  (Vektorraum $D_n^{\mathbb{C}}$ hat $n$ kanonische Basisvektoren)
- Konjugation einer komplexen Zahl: $\overline{(x, y)} = (x, -y)$
- inneres Produkt für $f, g \in D_n^{\mathcal{C}}$: $\langle f, g \rangle = \sum_{x=0}^{n-1} f(x) * \overline{g(x)}$
- Entwicklungsformel für Orthonormalbasis $(v_0, \ldots, v_{n-1})$:

$$\forall v \in D_n^{\mathcal{C}} : v = \sum_{i=0}^{n-1} \langle v, v_i \rangle v_i$$

# DFT - Beispiel im Vektorraum $D_4^{\mathbb{C}}$





# DFT - Fourier-Basis



$$\begin{aligned} e_j(x) &= \frac{1}{\sqrt{n}} e^{\frac{i2\pi jx}{n}} \\ &= \frac{1}{\sqrt{n}} \cos\frac{2\pi jx}{n} + \frac{1}{\sqrt{n}} i \sin\frac{2\pi jx}{n} \end{aligned}$$

mit $j = 0, \ldots, n-1$, $x = 0, \ldots, n-1$ und $i = \sqrt{-1}$

Orthonormalität:

$$||e_j(x)|| = \sqrt{\langle e_j(x), e_j(x)\rangle} = 1 \qquad \text{für } j = 0, 1, \ldots, n-1$$

$$\langle e_j(x), e_k(x)\rangle = 0 \qquad \text{für } j, k = 0, 1, \ldots, n-1 \text{ und } j \neq k$$

# DFT - Fourier-Koeffizienten

- aufgrund Orthonormalität der Fourier-Basis Berechnung der Fourier-Koeffizienten mittels innerem Produkt:

$$(re_j, im_j) = \langle f_n(x), e_j(x) \rangle$$

  $\longrightarrow$ Transformation als einfache Multiplikation mit DFT-Matrix möglich
- Transformation entspricht Rotation im komplexen, hochdimensionalen Raum
- Ergebnis: komplexe Fourier-Koeffizienten
  - Realteil für Kosinusamplituden
  - Imaginärteil für Sinusamplituden

# DFT - Transformationsformel



$$
\begin{aligned}
F_n(j) &= \langle f_n(x), e_j(x) \rangle \\
&= \sum_{x=0}^{n-1} f(x) \cdot \overline{e_j(x)} \\
&= \sum_{x=0}^{n-1} f(x) \cdot \overline{\frac{1}{\sqrt{n}} e^{\frac{i2\pi jx}{n}}} \\
&= \frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} f(x) \cdot e^{-\frac{i2\pi jx}{n}} \\
&= \left( \frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} f(x) \cos \frac{2\pi jx}{n}, -\frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} f(x) \sin \frac{2\pi jx}{n} \right)
\end{aligned}
$$

# DFT - Rcktransformation $\mathrm{DFT}^{-1}$



$$
\begin{aligned}
f_n(x) &= \sum_{j=0}^{n-1} F_n(j) \cdot e_j(x) \\
&= \sum_{j=0}^{n-1} F_n(j) \cdot \frac{1}{\sqrt{n}} e^{\frac{i2\pi jx}{n}} \\
&= \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} F_n(j) \cdot e^{\frac{i2\pi jx}{n}} \\
&= \left( \frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} F(x) \cos \frac{2\pi jx}{n}, \frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} F(x) \sin \frac{2\pi jx}{n} \right)
\end{aligned}
$$

# DFT - Polarkoordinaten

komplexe Zahl $(x, y)$ als Polarkoordinaten mit Länge und Winkel

- Winkel (Phase): $\tan \gamma = \frac{y}{x}$
- Länge: $l = \sqrt{x^2 + y^2}$
- Winkel drückt Verschiebung aus (Sinus versus Kosinus)
- Frequenzspektrum berücksichtigt nur Länge

# DFT - Eigenschaften

- Parseval-Theorem: $\langle f_n^1(x), f_n^2(x) \rangle = \langle F_n^1(x), F_n^2(x) \rangle$
  $\longrightarrow$ euklidsche Distanzen sind im Orts- und Frequenzbereich gleich
- Translation im Ortsbereich ändert ausschließlich Phasenwinkel
- Symmetrie der Fourier-Koeffizienten: Werte sind spiegelsymmetrisch
  $\longrightarrow$ n reelle Zahlen reichen zur Darstellung von $F_n(x)$
- Nyquist-Theorem (Abtasttheorem): zur Abbildung bestimmter Frequenz mind. doppelt so viele Abtastwerte erforderlich $\longrightarrow$ Symmetrie

# Overview

# DWT - Allgemeines



- hier Fokus auf Haar-Wavelets (nach Alfred Haar) als einfachstes Wavelet
- „Wavelet“steht für Wellchen, also lokal begrenzte Welle
- vielfältiger Einsatz etwa in Signal- und Bildverarbeitung (etwa JPEG2000)

# DWT - Probleme der DFT

## Probleme mit der Fourier-Transformation

- lokale versus globale Änderung:
  - lokale Änderung im Ortsbereich
    $\longrightarrow$ globale Änderung im Frequenzbereich und umgekehrt
  - Problem: etwa temporäre Störgeräusche aus Audio-Signal entfernen
- Ort und Frequenz als Feature-Wert:
  - beides nicht gemeinsam in einer Darstellung verfügbar
  - Problem etwa bei Erkennung lokal begrenzter Texturen

# DWT - Allgemeine Idee

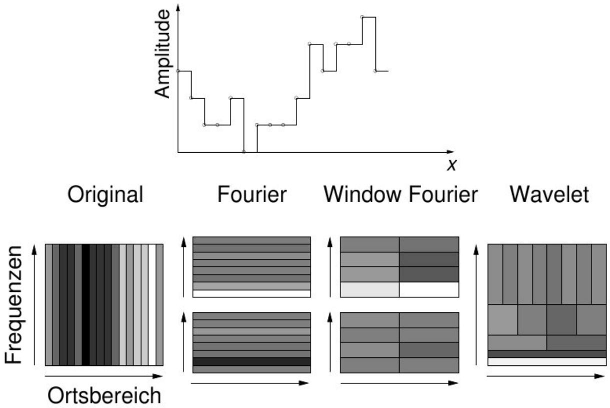


- gemeinsame Darstellung von Frequenz und Ort
- Ansatz für Fourier-Transformation: Window-Fourier-Transformation
  - Zerlegung Ausgangssignal in disjunkte Intervalle (Fenster) konstanter Breite
  - Fourier-Transformation isoliert auf einzelnen Intervallen
  - Problem: statische Intervallbreite
- Wavelet-Transformation: Frequenzen bei unterschiedlicher Ortsauflösung
  $\longrightarrow$ Multi-Resolution-Analyse
- Einschränkung der Frequenzen durch Nyquist-Abtasttheorem
  $\longrightarrow$ je größer Ortsauflösung, desto geringer Frequenzauflösung und umgekehrt

# DWT - Graphische Darstellung



Ausgangssignal:



# DWT - Wavelet-Basisfunktionen



- Support (Funktionswert ungleich Null) lokal begrenzt $\longrightarrow$ Wellchen
- Generierung von Basisfunktionen aus „Mutter-Wavelet“ durch Verschiebung und Skalierung
- Existenz diverser Mutter-Wavelets (hier nur Haar-Mutter-Wavelet)

# DWT - Haar-Wavelet



## Funktionsprinzip (stark vereinfacht)

- Ausgangspunkt: diskrete Funktion mit $2^n$ Funktionswerten
- schrittweises und iteriertes Berechnen der Summen (Skalierungswerte) und Differenzen (Detailkoeffizienten)
- Abbildung der Ausgangsfunktion auf $2^n - 1$ Detailkoeffizienten und einen Skalierungswert (Gesamtsumme)
- Ausgangsfunktion kann verlustfrei rekonstruiert werden

# DWT - Haar-Wavelet

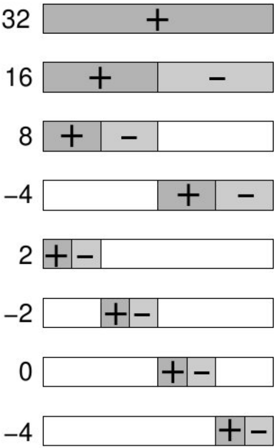
**Beispiel**

Ausgangsfunktion: $[\,9 \;\; 7 \;\; 3 \;\; 5 \;\; 1 \;\; 1 \;\; 1 \;\; 5\,]$

| Auflösungsstufe | Skalierungswerte | Detailkoeffizienten |
|---|---|---|
| 1 | $[\,9 \;\; 7 \;\; 3 \;\; 5 \;\; 1 \;\; 1 \;\; 1 \;\; 5\,]$ | |
| 2 | $[\,16 \;\; 8 \;\; 2 \;\; 6\,]$ | $[\,2 \;\; -2 \;\; 0 \;\; -4\,]$ |
| 4 | $[\,24 \;\; 8\,]$ | $[\,8 \;\; -4\,]$ |
| 8 | $[\,32\,]$ | $[\,16\,]$ |

Ergebnis: $[\,32 \;\; 16 \;\; 8 \;\; -4 \;\; 2 \;\; -2 \;\; 0 \;\; -4\,]$

## Support der einzelnen Wavelet-Basisfunktionen



# DWT - Anwendungen



### Feature-Normalisierung

- Störfrequenzen lassen sich lokal begrenzt entfernen
- Mutter-Wavelet kann an Störsignal angepasst werden
  $\longrightarrow$ aufwändige Analyse erforderlich
- Beispiel: Entfernen von Knackgeräuschen aus Audio-Signal

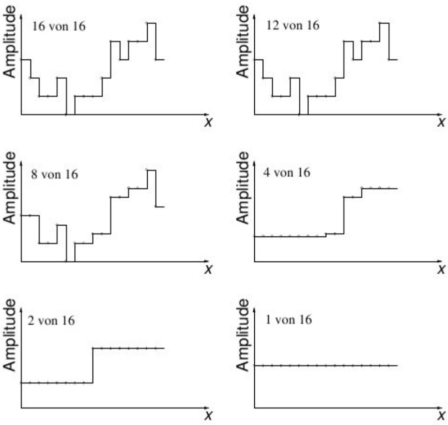
# DWT - Anwendungen



## Feature-Erkennung/-Aufbereitung

- Anwendung für lokale Frequenzanalyse, etwa für Textur-Feature
- Invarianzen
  - können an Orts- & Frequenzinformationen geknüpft sein
  - Verschiebungsinvarianz durch unsortierte Koeffizienten
  - Invarianz bzgl. Skalierung (Verdopplung/-Halbierung der Ortsauflösung) durch Nichtbeachtung der Auflösungsstufen
- Haar-Wavelet: geringe Berechnungskomplexität: $O(n)$
- Kompaktheit und Orthogonalität der Koeffizienten
- lokale Beschränkung bei Modifikation der Wavelet-Koeffizienten

## Anwendung zur verlustbehafteten Komprimierung



# DWT - Berechnung



- Ausgangspunkt: diskrete Funktion $f_n(x) \in D_n^{\mathbb{R}}$
- Berechnung der Detailkoeffizienten $\Psi^j$ und Skalierungswerte $\Phi^j$ in verschiedenen Auflösungsstufen $j = 1, 2, 4, 8, 16, 32, \ldots$
- orthonormale Basisvektoren: $\Psi(x)$ und $\Phi(x)$

# DWT - Skalierungsbasisvektoren



i-ter Skalierungsbasisvektor $\Phi_i^j(x)$ der Auflösungsstufe $j$ des Vektorraums $D_n^{\mathbb{R}}$:

$$\Phi_i^j(x) = 1/\sqrt{j} \cdot \Phi(x/j - i) \qquad \text{für } i = 0, \ldots, n/j - 1 \text{ mit}$$

$$\Phi(x) = \begin{cases} 1 & \text{für } 0 \leq x < 1 \\ 0 & \text{sonst.} \end{cases}$$

es gelten:

$$||\Phi_i^j(x)|| = \sqrt{\langle \Phi_i^j(x), \Phi_i^j(x) \rangle} = 1 \qquad \text{für } i = 0, \ldots, n/j - 1.$$

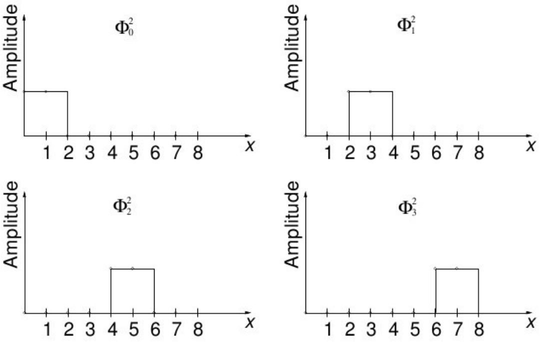
$$\langle \Phi_i^j(x), \Phi_k^j(x) \rangle = 0 \qquad \text{für } i, k = 0, \ldots, n/j - 1 \text{ und } i \neq k.$$

# DWT - Skalierungsbasisvektoren

Fourier Transform

Wavelet Transform

## Skalierungsvektoren der Stufe j=2



# DWT - Detailbasisvektoren



i-ter Detailbasisvektor $\Phi_i^j(x)$ der Auflösungsstufe $j$ des Vektorraums $D_n^{\mathbb{R}}$:

$$\Psi_i^j(x) = 1/\sqrt{j} \cdot \Psi(x/j - i) \qquad \text{für } i = 0, \ldots, n/j - 1 \text{ mit}$$

$$\Psi(x) = \begin{cases} 1 & \text{für } 0 \leq x < 1/2 \\ -1 & \text{für } 1/2 \leq x < 1 \\ 0 & \text{sonst.} \end{cases}$$

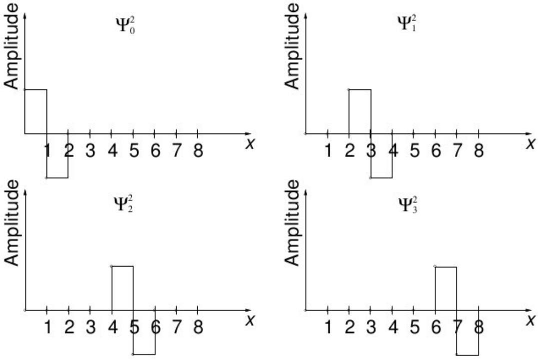
es gelten:

$$\langle \Psi_i^j(x), \Psi_k^j(x) \rangle = \delta_{i,k} \qquad \text{für } i, k = 0, \ldots, n/j - 1 \text{ mit}$$

$$\delta_{i,k} = \begin{cases} 1 & : \quad i = k \\ 0 & : \quad i \neq k. \end{cases}$$

# DWT - Detailbasisvektoren

## Detailvektoren der Stufe j=2



# DWT - Skalierungs- und Detailbasisvektoren



- Detail- und Skalierungsbasisvektoren derselben Auflösung sind orthogonal

$$\langle \Psi_i^j(x), \Phi_k^j(x) \rangle = 0 \qquad \text{für } i, k = 0, \ldots, n/j - 1.$$

- bilden gemeinsam orthonormale Basis für Vektorraum $D_{2n/j}^{\mathbb{R}}$

# DWT - Skalierungs- und Detailbasisvektoren



Grundidee: Anwendung inneres Produkt der Vektoren der Orthonormalbasis der Stufe $j$ auf $f_n(x) \in D_n^{\mathcal{R}}$

$$\begin{aligned}
\Phi_i^j &= \langle f_n(x), \Phi_i^j(x) \rangle \\
&= 1/\sqrt{j} \sum_{x=0}^{n-1} f_n(x) \cdot \Phi(x/j - i)
\end{aligned}$$

$$\begin{aligned}
\Psi_i^j &= \langle f_n(x), \Psi_i^j(x) \rangle \\
&= 1/\sqrt{j} \sum_{x=0}^{n-1} f_n(x) \cdot \Psi(x/j - i)
\end{aligned}$$

# DWT - Skalierungs- und Detailbasisvektoren

Berechnung auf $f_n(x) \in D_n^{\mathcal{R}}$

- erzeugt $n/2$ Skalierungskoeffizienten
  drücken Frequenzen innerhalb entspr. Supportintervalle aus
- erzeugt $n/2$ Detailkoeffizienten
  drückt die Funktion ohne Frequenzen innerhalb entspr. Supportintervalle aus
- erneute Berechnung auf Funktion der Detailkoeffizienten
  $\longrightarrow$ nächste Auflösungsstufe
- Stopp, wenn Auflösungsstufe und Anzahl Werte gleich sind

# DWT - Zerlegung in Wavelet-Koeffizienten



Wavelet-Koeffizienten einer Funktion $f_n(x) \in D_n^{\mathcal{R}}$ sind

$$\Phi_0^n \Psi_0^n \Psi_0^{n/2} \Psi_1^{n/2} \Psi_0^{n/4} \Psi_1^{n/4} \Psi_2^{n/4} \Psi_3^{n/4} \ldots \Psi_0^2 \ldots \Psi_{n/2-1}^2$$

mit

$$\begin{aligned} \Psi_i^j &= 1/\sqrt{j} \sum_{x=0}^{n-1} f_n(x) \cdot \Psi(x/j - i) \\ \Phi_i^j &= 1/\sqrt{j} \sum_{x=0}^{n-1} f_n(x) \cdot \Phi(x/j - i) \end{aligned}$$

# DWT - Zerlegung in Wavelet-Koeffizienten

$$
\begin{tikzcd}[column sep=small]
\text{Ausgangsfunktion:} & & & \Phi^1 \in D_8^{\mathbb{R}} \arrow[dl] \arrow[dr] & & \text{Auflösung } 1 \\
& & \Phi^2 \in D_4^{\mathbb{R}} \arrow[dl] \arrow[dr] & & \Psi^2 \in D_4^{\mathbb{R}} & 2 \\
& \Phi^4 \in D_2^{\mathbb{R}} \arrow[dl] \arrow[dr] & & \Psi^4 \in D_2^{\mathbb{R}} & & 4 \\
\Phi^8 \in D_1^{\mathbb{R}} & & \Psi^8 \in D_1^{\mathbb{R}} & & & 8
\end{tikzcd}
$$

# DWT - Darstellung als Matrizenmultiplikation



- Funktionen $f_n(z)$ und $F_n(x)$ als Vektoren aus $\mathbb{R}^n$
- $A \in \mathbb{R}^{n \times n}$ ist eine $n \times n$-Matrix, deren $n$ Zeilen den Wavelet-Basisvektoren entsprechen
- auf Grund $AA^* = I$ (Orthonomalmatrix) gilt $||f|| = ||Af||$
- quadratischer Berechnungsaufwand

# DWT - Transformation mit linearem Aufwand



```
procedure Zerlegung(c: array [1..n] of reals)
  while n>1 do
    Zerlegungsschritt(c[1..n])
    n := n/2
  end while
end procedure

procedure Zerlegungsschritt(c: array [1..n] of reals)
  for i=1 to n/2 do
    cc[i] := (c[2i-1]+c[2i])/√2
    cc[n/2+i] := (c[2i-1]-c[2i])/√2
  end for
  c := cc
end procedure
```

# DWT - Rcktransformation mit linearem Aufwand



**procedure** Rekonstruktion(**c: array [1..n] of reals**)
  g := 2
  **while** g $\leq$ n **do**
    Rekonstruktionsschritt(c[1..g])
    g := 2g
  **end while**
**end procedure**

**procedure** Rekonstruktionsschritt(**c: array [1..n] of reals**)
  **for** i=1 **to** n/2 **do**
    cc[2i-1] := (c[i]+c[n/2+i])/$\sqrt{2}$
    cc[2i] := (c[i]-c[n/2+i])/$\sqrt{2}$
  **end for**
  c := cc
**end procedure**

# DWT - Zweidimensionaler Fall

- wichtig etwa für Rasterbilder
- 2 Varianten
  - *Standardzerlegung:* Transformation in Dimension 1 komplett, bevor Transformation in Dimension 2 startet
  - *Non-Standardzerlegung:* Transformation alternierend pro Auflösungsstufe
- analoges Verfahren für beliebig viele Dimensionen anwendbar

# DWT - Algorithmus zur Standardzerlegung



**procedure** StandardZerl(**c: array [1..m,1..n] of reals**)
  **for** row := 1 **to** m **do**
    Zerlegung(c[row,1..n])
  **end for**
  **for** col := 1 **to** n **do**
    Zerlegung(c[1..m,col])
  **end for**
**end procedure**

# DWT - Algorithmus zur Non-Standardzerlegung



```
procedure NonStandardZerl(c: array [1..n,1..n] of reals)
  while n>1 do
    for row := 1 to n do
      Zerlegungsschritt(c[row,1..n])
    end for
    for col := 1 to n do
      Zerlegungsschritt(c[1..n,col])
    end for
    n := n/2
  end while
end procedure
```