

Ziel: Minimalität der Feature-Werte

Ausnutzung Kompaktheit im Frequenzbereich

Kompaktheit:

- ♦ Funktion häufig durch wenige, niedrige Frequenzkoeffizienten approximierbar, da hohe Frequenzen oft gegen Null streben
- ♦ hohe Kompaktheit bei glattem Funktionsverlauf

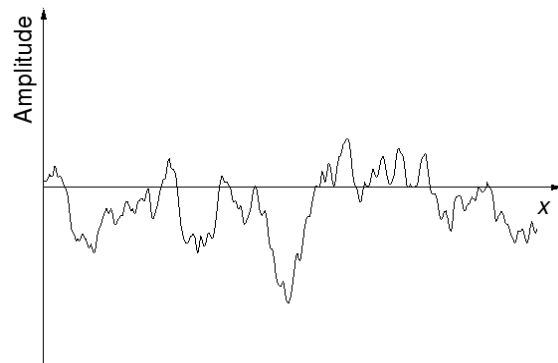
Fehler erzeugt durch Frequenzfilter im Ortsbereich nicht lokalisierbar

Orthogonalität: Fourier-Koeffizienten sind orthogonal

→ ermöglicht isolierte Manipulation einzelner Frequenzen

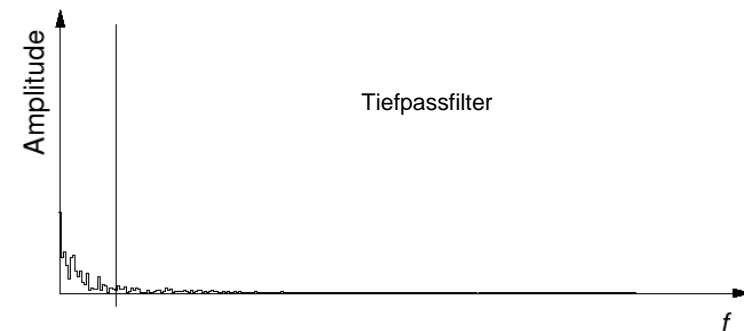
Beispiel Minimierung durch Ausnutzung Kompaktheit

Ausgangsfunktion:



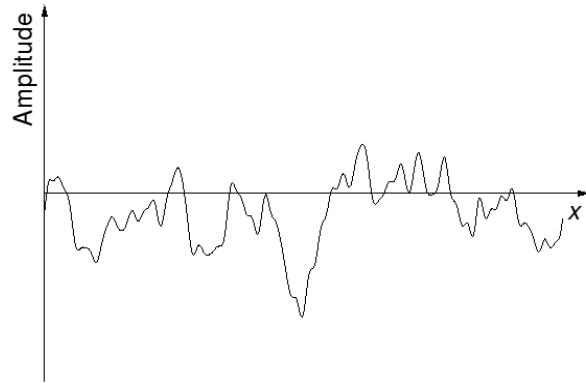
Beispiel Minimierung durch Ausnutzung Kompaktheit (2)

Frequenzspektrum und Abschneiden hoher Frequenzen:



Beispiel Minimierung durch Ausnutzung Kompaktheit (3)

approximierte, d.h. minimierte Funktion:



Erinnerung lineare Algebra und komplexe Zahlen

diskrete Funktion $f(x)$ als Vektor des komplexen Vektorraums:
(Vektorraum $D_n^{\mathbb{C}}$ hat n kanonische Basisvektoren)

Konjugation einer komplexen Zahl: $\overline{(x, y)} = (x, -y)$

$$f(x) \in D_n^{\mathbb{C}}$$

Berechnung der DFT (2)

inneres Produkt für

$$f, g \in D_n^{\mathbb{C}}: \langle f, g \rangle = \sum_{x=0}^{n-1} f(x) * \overline{g(x)}$$

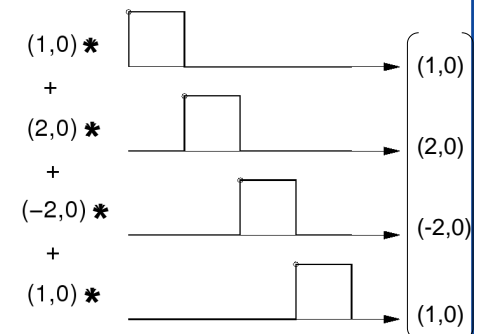
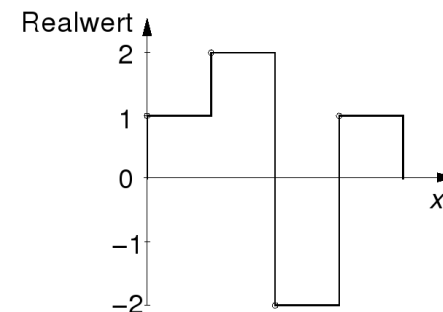
Entwicklungsformel für Orthonormalbasis

(v_0, \dots, v_{n-1}) :

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

Beispiel Funktion im Vektorraum

$$D_4^{\mathbb{C}}$$

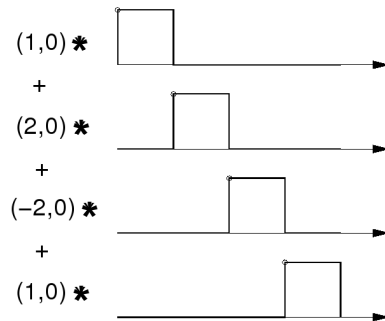


Vergleiche Vektorräume C^4 und



$$D_4^C$$

$$\begin{pmatrix} (1,0) \\ (2,0) \\ (-2,0) \\ (1,0) \end{pmatrix}$$



$\begin{pmatrix} (1,0) \\ (0,0) \\ (0,0) \\ (0,0) \end{pmatrix}, \begin{pmatrix} (0,0) \\ (0,0) \\ (-2,0) \\ (0,0) \end{pmatrix}$

Inneres Produkt

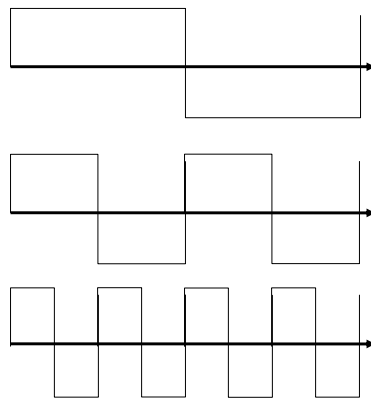
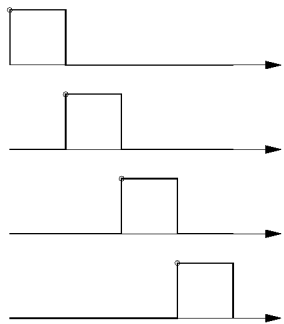
$$= (0,0) * \overline{(0,0)} + (0,0) * \overline{(0,0)} + (-2,0) * \overline{(0,0)} + (0,0) * \overline{(1,0)} = (0,0)$$

Erinnerung
 $\overline{(a,b)} = (a,-b)$

Vektorräume über Diskreten Funktionen <isweb>

Mit Ortsvariablen

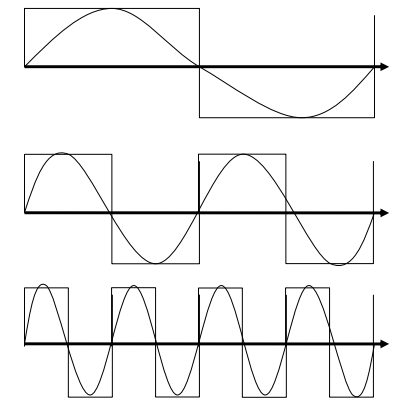
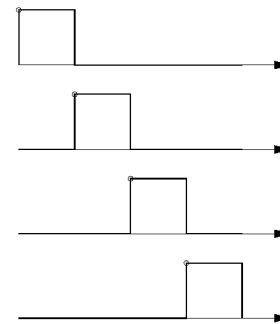
Mit Frequenzvariablen



Vektorräume über Funktionen <isweb>

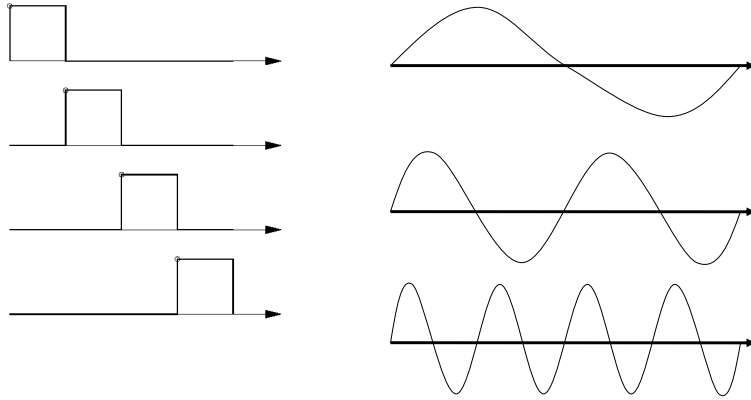
Mit Ortsvariablen

Mit Frequenzvariablen



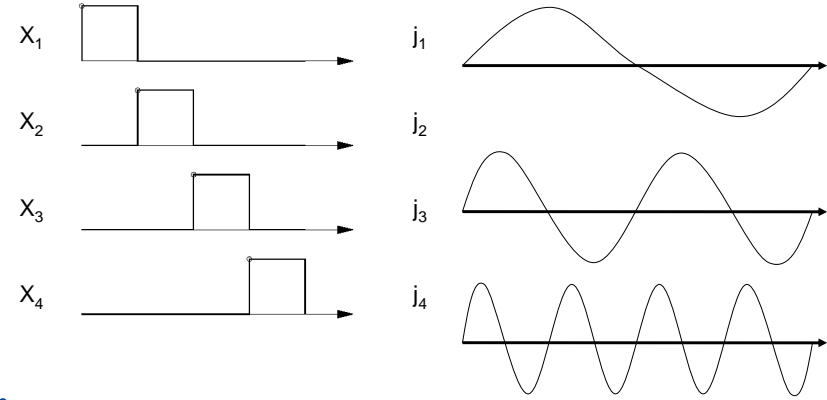
Mit Ortsvariablen

Mit Frequenzvariablen



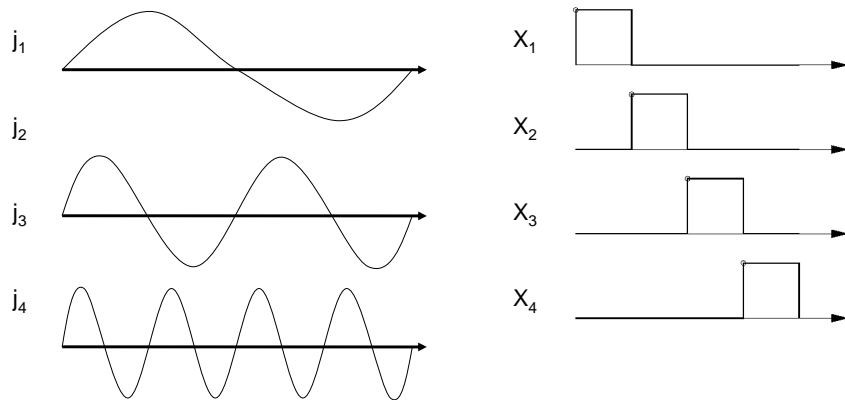
Gesucht:

Gegeben:



Gesucht:

Gegeben:



Basisvektoren:

$$e_j(x) = \frac{1}{\sqrt{n}} e^{i \frac{2\pi j x}{n}}$$

$$= \frac{1}{\sqrt{n}} \cos \frac{2\pi j x}{n} + \frac{1}{\sqrt{n}} i \sin \frac{2\pi j x}{n}$$

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

Orthonormalität:

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

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

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

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

\uparrow \uparrow \uparrow \uparrow
 Cosinus / Sinus-Anteil Ort j-ter Einheitsvektor
 für die j-te der Fourierbasis
 Schwingungsfrequenz

→ Transformation als einfache Multiplikation mit DFT-Matrix

Transformation entspricht Rotation im komplexen,
hochdimensionalen Raum

Ergebnis: komplexe Fourier-Koeffizienten

- Realteil für Kosinusamplituden
- Imaginärteil für Sinusamplituden

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

Allgemein:

$$F = A \times f,$$

Mit

$$a_{j+1, k+1} = \overline{e_j(k)}$$

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

Allgemein:

$$\begin{aligned}
 A^* F &= \\
 A^* A x &= f \\
 I x &= f,
 \end{aligned}$$

Mit:
A* ist adjungierte Matrix

komplexe Zahl (x, y) als Polarkoordinate 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

Parseval-Theorem:

→ euklidische Distanzen sind im Orts- und Frequenzbereich gleich

$$\langle f_n^1(x), f_n^2(x) \rangle = \langle F_n^1(x), F_n^2(x) \rangle$$

Translation im Ortsbereich ändert ausschließlich Phasenwinkel

Symmetrie der Fourier-Koeffizienten:

Werte sind spiegelsymmetrisch

→ n reelle Zahlen reichen zur Darstellung von $F_n(x)$

Nyquist-Theorem (Abtasttheorem):

zur Abbildung bestimmter Frequenzen sind min. doppelt
so viele Abtastwerte erforderlich

→ Symmetrie

Kompaktheit

abhängig von Glattheit der Funktion im Ortsbereich

Klassifikation von Funktionen anhand Verlauf der quadrierten Fourier-Beträge in O-Notation:

 $O(F^{-b})$

Kompaktheit $O(F^{-b})$	Rauschfarbe
$b = 0$	weiß
$b = 1$	pink
$b = 2$	braun
$b > 2$	schwarz

Kompaktheit

weißes Rauschen: geringste Kompaktheit

schwarzes Rauschen: sehr glatt, etwa Flusspegelstände

braunes Rauschen: etwa Verlauf von Aktienkursen

abrupte Funktionswertsprünge verursachen geringe Kompaktheit

extreme Sprünge häufig aufgrund fehlender Periodizität an den Rändern

Vermeidung Randeffekt: Spiegelung der Funktion

→ Kosinustransformation DCT (keine Imaginärteile)

Hintransformation:

$$F_n(j) = \frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} f(x) \cdot \cos \frac{\pi j(x+0,5)}{n} \quad \text{für } j = 0, \dots, n-1.$$

Rücktransformation:

$$f_n(x) = \frac{F_n(0)}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{j=1}^{n-1} F(j) \cdot \cos \frac{\pi j(x+0,5)}{n} \quad \text{für } j = 0, \dots, n-1.$$

FFT: schneller Transformationsalgorithmus mit $O(n \log n)$ statt $O(n^2)$

zweidimensionale DFT durch einfache Multiplikation der Basisvektoren erreichbar

$$F_{m,n}(j, k) = \frac{1}{\sqrt{mn}} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f_{m,n}(x, y) \cdot e^{-\frac{i2\pi jx}{m}} \cdot e^{-\frac{i2\pi ky}{n}}$$

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)

lokal versus globale Änderung:

- ♦ lokale Änderung im Ortsbereich
→ 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

gemeinsame Darstellung von Frequenz und Ort

Ansatz für Fourier-Transformation:

Window-Fourier-Transformation

- ♦ Zerlegung des Ausgangssignals in disjunkte Intervalle (Fenster) konstanter Breite
- ♦ Fourier-Transformation isoliert auf einzelnen Intervallen
- ♦ Problem: **statische** Intervallbreite

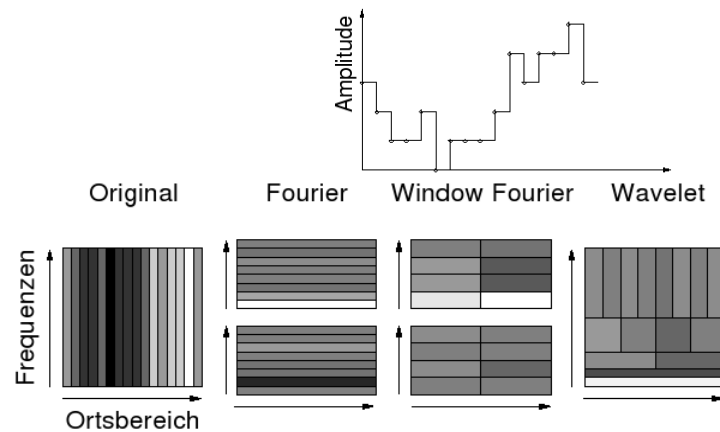
Wavelet-Transformation: Frequenzen bei unterschiedlicher Ortsauflösung

→ Multi-Resolution-Analyse

Einschränkung der Frequenzen durch Nyquist-Abtasttheorem

→ je größer Ortsauflösung, desto geringer Frequenzauflösung und umgekehrt

Ausgangssignal:



Support (Funktionswert ungleich Null) lokal begrenzen → Wellchen

Generierung von Basisfunktionen aus „Mutter-Wavelet“ durch Verschiebung und Skalierung

Existenz diverser Mutter-Wavelets (hier nur Haar-Mutter-Wavelet)

Funktionsprinzip (stark vereinfacht):

Ausgangspunkt: diskrete Funktion mit 2^n Funktionswerten

schrittweises und iteriertes Berechnen der Summen (Skalierungswerte) und Differenzen (Detailkoeffizienten)

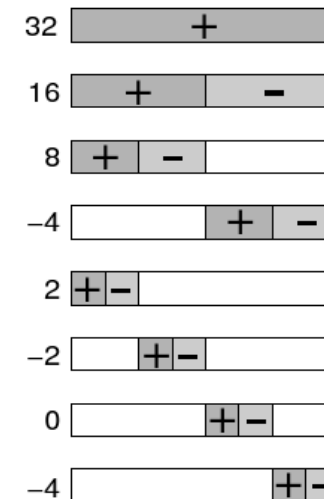
Abbildung der Ausgangsfunktion auf $2^n - 1$ Detailkoeffizienten und einen Skalierungswert (Gesamtsumme)

Ausgangsfunktion kann verlustfrei rekonstruiert werden

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]



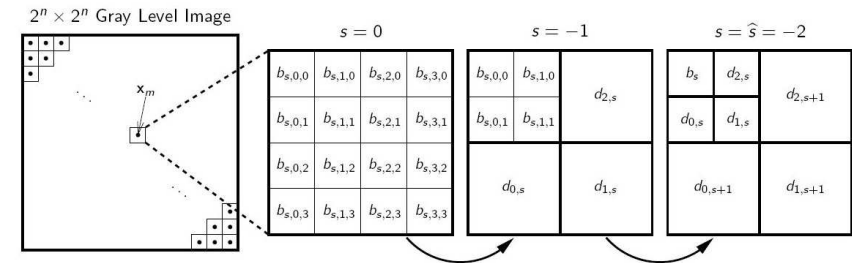
Störfrequenzen lassen sich lokal begrenzt entfernen

Mutter-Wavelet kann an Störsignal angepasst werden
→ aufwändige Analyse erforderlich

Beispiel: Entfernen von Knackgeräuschen aus Audio-Signale

Anwendung DWT für Feature-Erkennung/Aufbereitung

Anwendung für lokale Frequenzanalyse, etwa für Textur-Feature



$$\mathbf{c}_m = \mathbf{c}(x_m) = \begin{pmatrix} c_{m,1} \\ c_{m,2} \end{pmatrix} = \begin{pmatrix} \ln(2^s |b_s|) \\ \ln[2^s (|d_{0,s}| + |d_{1,s}| + |d_{2,s}|)] \end{pmatrix}$$

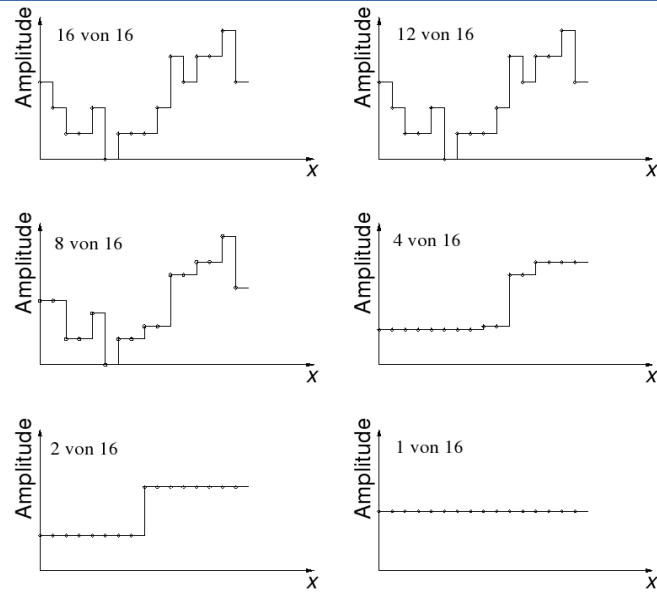
Anwendung DWT für Feature-Erkennung/Aufbereitung (2)

Invarianzen

- ♦ können an Orts- und Frequenzinformationen geknüpft sein
- ♦ Verschiebungsinvarianz durch unsortierte Koeffizienten
- ♦ Invarianz bzgl. Skalierung (Verdopplung/-Halbierung der Ortsauflösung) durch Nichtbeachtung der Auflösungsstufen

Anwendung DWT für Feature-Erkennung/Aufbereitung (3)

Haar-Wavelet: geringe Berechnungs-komplexität:
Kompaktheit und Orthogonalität der Koeffizienten
lokale Beschränkung bei Modifikation der
Wavelet-Koeffizienten $O(n)$



Ausgangspunkt:

diskrete Funktion $f_n(x) \in D_n^{\mathbb{R}}$

Berechnung der Detailkoeffizienten Ψ^j
und Skalierungswerte Φ^j in verschiedenen
Auflösungsstufen $j = 1, 2, 4, 8, 16, 32, \dots$

orthonormale Basisvektoren: $\Psi(x)$ und $\Phi(x)$

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

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

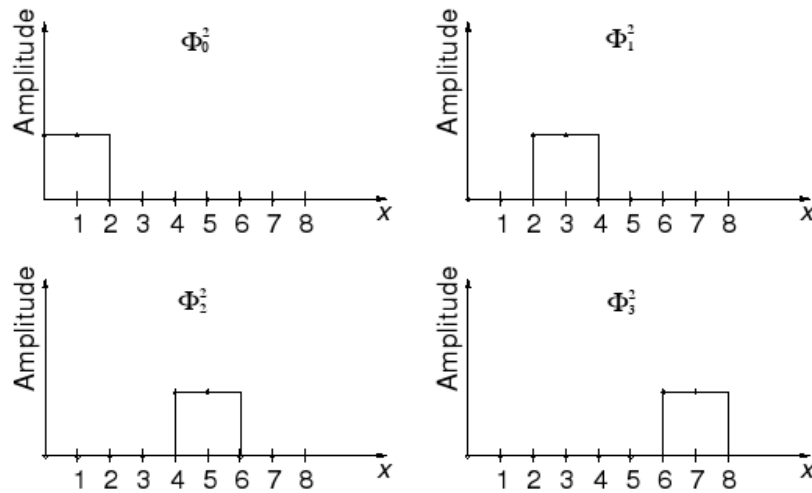
$$\Phi(x) = \begin{cases} 1 & \text{für } 0 \leq x < 1 \\ 0 & \text{sonst.} \end{cases}$$

es gelten:

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

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

Skalierungsvektoren der Stufe j=2



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

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

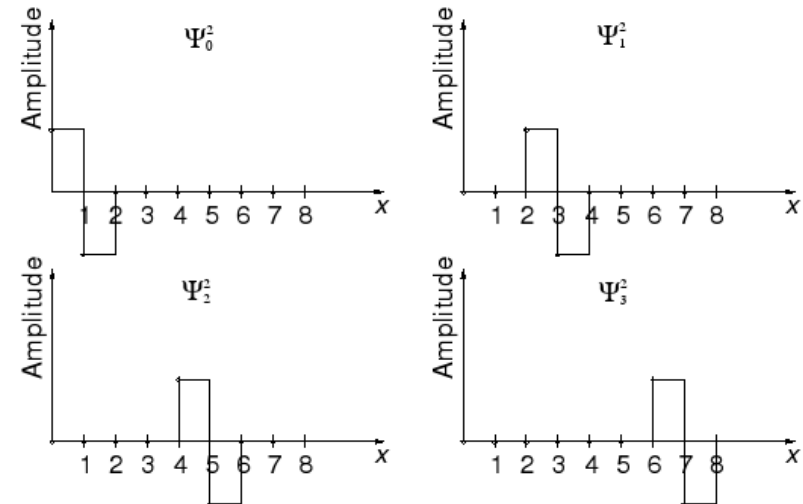
$$\Psi(x) = \begin{cases} 1 & \text{für } 0 \leq x < 1/2 \\ -1 & \text{für } 1/2 \leq x < 1 \\ 0 & \text{sonst.} \end{cases}$$

es gelten:

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

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

Detailvektoren der Stufe j=2



Detail- und Skalierungsbasisvektoren der selben Auflösung sind orthogonal

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

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

der Skalierungs- und Detailkoeffizienten

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

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

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

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

erzeugt $n/2$ Skalierungskoeffizienten
drücken Frequenzen innerhalb entsprechender Supportintervalle aus

erzeugt $n/2$ Detailkoeffizienten
drückt die Funktion ohne Frequenzen innerhalb entsprechender Supportintervalle aus

erneute Berechnung auf Funktion der Detailkoeffizienten
→ nächste Auflösungsstufe

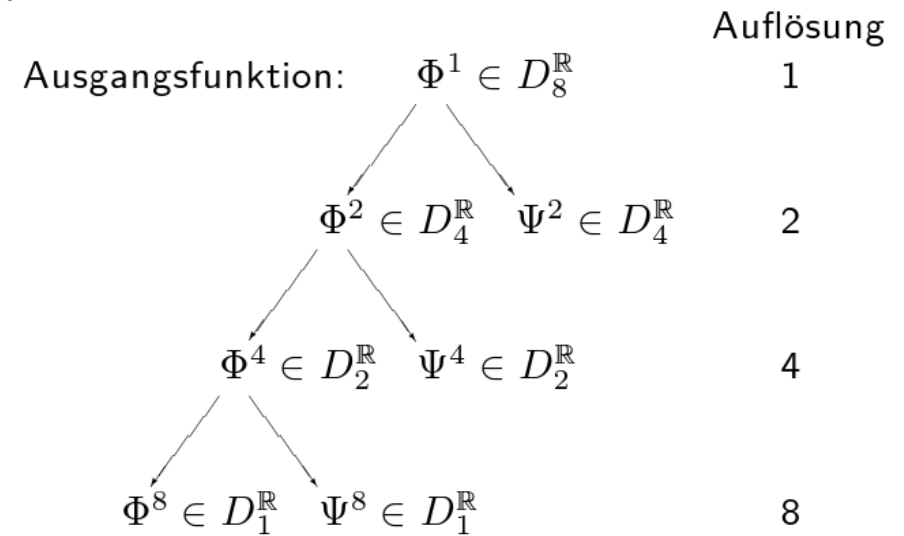
Stopp, wenn Auflösungsstufe und Werte gleich sind

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

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

$$\Psi_i^j = 1/\sqrt{j} \sum_{x=0}^{n-1} f_n(x) \cdot \Psi(x/j - i)$$

$$\Phi_i^j = 1/\sqrt{j} \sum_{x=0}^{n-1} f_n(x) \cdot \Phi(x/j - i)$$



Funktion $f_n(z)$ 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 $AA^* = I$ (Orthonormalmatrix) gilt

quadratischer Berechnungsaufwand

$$\|f\| = \|Af\|$$

```

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
    
```

```

procedure Rekonstruktion(c: array [1..n] of reals)
  g := 2
  while g ≤ 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])/√2
    cc[2i] := (c[i]-c[n/2+i])/√2
  end for
  c := cc
end procedure

```

wichtig etwa für Rasterbilder

2 Varianten

- ♦ *Standardzerlegung: Transformation in Dimension 1 komplett, bevor Transformation in Dimension 2*
- ♦ *Non-Standardzerlegung: Transformation alternierend pro Auflösungsstufe*

analoges Verfahren für beliebig viele Dimensionen
anwendbar

```

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

```

```

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

```