Multimedia Retrieval im WS 2011/2012

4. Feature-Transformationsverfahren

4.2 Diskrete Wavelet-Transformation

Prof. Dr.-Ing. Marcin Grzegorzek
Juniorprofessur für Mustererkennung
Institut für Bildinformatik im Department ETI
Fakultät IV der Universität Siegen

14. November 2011

Inhalte und Termine

1. Einführung

1.1 Grundlegende Begriffe

1.2 Suche in einem MMDBS

1.3 MMDBMS-Anwendungen 11.10.2011

2. Prinzipien des Information Retrieval

2.1 Einführung

2.2 Information-Retrieval-Modelle

2.3 Relevance Feedback

2.4 Bewertung von Retrieval-Systemen 17.10.2011

2.5 Nutzerprofile

Inhalte und Termine

3. Prinzipien des Multimedia Retrieval

3.1 Besonderheiten der Verwaltung und des Retrievals	
3.2 Ablauf des Multimedia-Information-Retrievals	
3.3 Daten eines Multimedia-Retrieval-Systems	24.10.2011
3.4 Feature	
3.5 Eignung verschiedener Retrieval-Modelle	
3.6 Multimedia-Ähnlichkeitsmodell	25.10.2011

4. Feature-Transformationsverfahren

4.1 Diskrete Fourier-Transformation	08.11.2011
4.2 Diskrete Wavelet-Transformation	14.11.2011
4.3 Karhunen-Loeve-Transformation	
4.4 Latent Semantic Indexing und Singulärwertzerlegung	

Inhalte und Termine

5. Distanzfunktionen

5.1 Eigenschaften und Klassifikation

5.2 Distanzfunktionen auf Punkten

5.3 Distanzfunktionen auf Binärdaten

5.4 Distanzfunktionen auf Sequenzen

5.5 Distanzfunktionen auf allgemeinen Mengen

6. Ähnlichkeitsmaße

6.1 Einführung

6.2 Distanz versus Ähnlichkeit

6.3 Grenzen von Ähnlichkeitsmaßen

6.4 Konkrete Ähnlichkeitsmaße

6.5 Aggregation von Ähnlichkeitswerten

6.6 Umwandlung von Distanzen in Ähnlichkeitswerte und Normierung

6.7 Partielle Ähnlichkeit

Inhalte und Termine

7. Effiziente Algorithmen und Datenstrukturen

7.1 Hochdimensionale Indexstrukturen

7.2 Algorithmen zur Aggregation von Ähnlichkeitswerten

8. Anfragebehandlung

8.1 Einführung

8.2 Konzepte der Anfragebehandlung

8.3 Datenbankmodell

8.4 Sprachen

9. Zusammenfassung

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

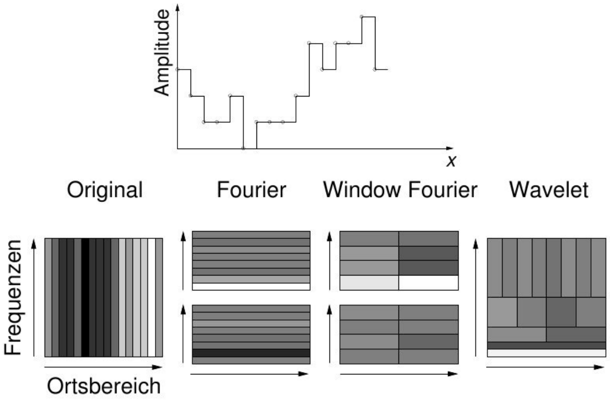
- lokale versus globale Änderung:
    - lokale Änderung im Ortsbereich
      $\longrightarrow$ globale Änderung im Frequenzbereich und umgekehrt
    - Problem: etwa temporäre Störgeräusche aus Audio-Signal entfernen
- Ort und Frequenz als Feature-Wert:
    - beides nicht gemeinsam in einer Darstellung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

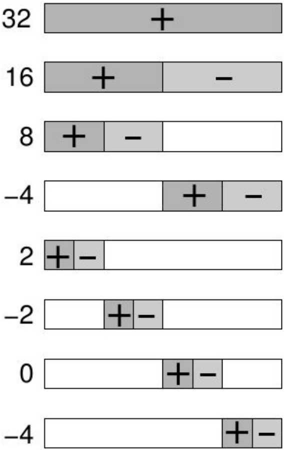
Ausgangsfunktion: [9 7 3 5 1 1 1 5]

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]

DWT - Haar-Wavelet

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

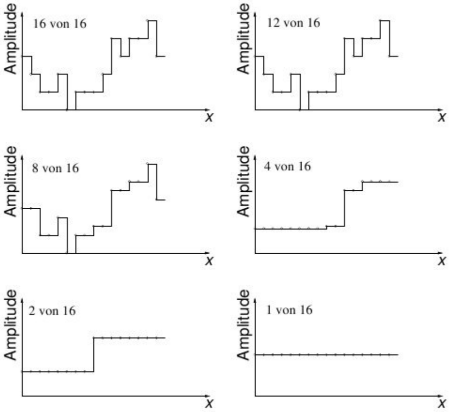
DWT - Anwendungen

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

DWT - Anwendungen

Anwendung zur verlustbehafteten Komprimierung



DWT - Berechnung

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

DWT - Skalierungsbasisvektoren

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

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

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

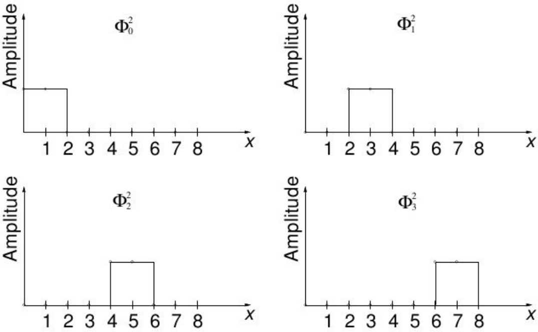
es gelten:

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

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

DWT - Skalierungsbasisvektoren

Skalierungsvektoren der Stufe j=2



DWT - Detailbasisvektoren

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

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

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

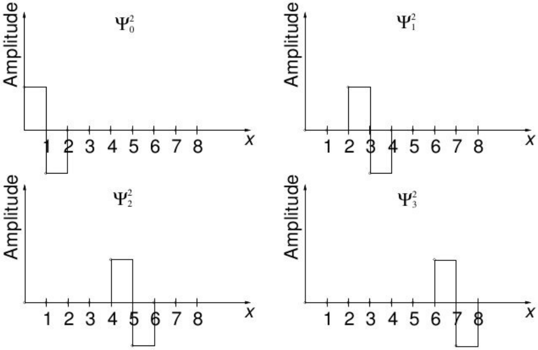
es gelten:

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

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

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

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

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 - Rücktransformation mit linearem Aufwand

```
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
```

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
```