
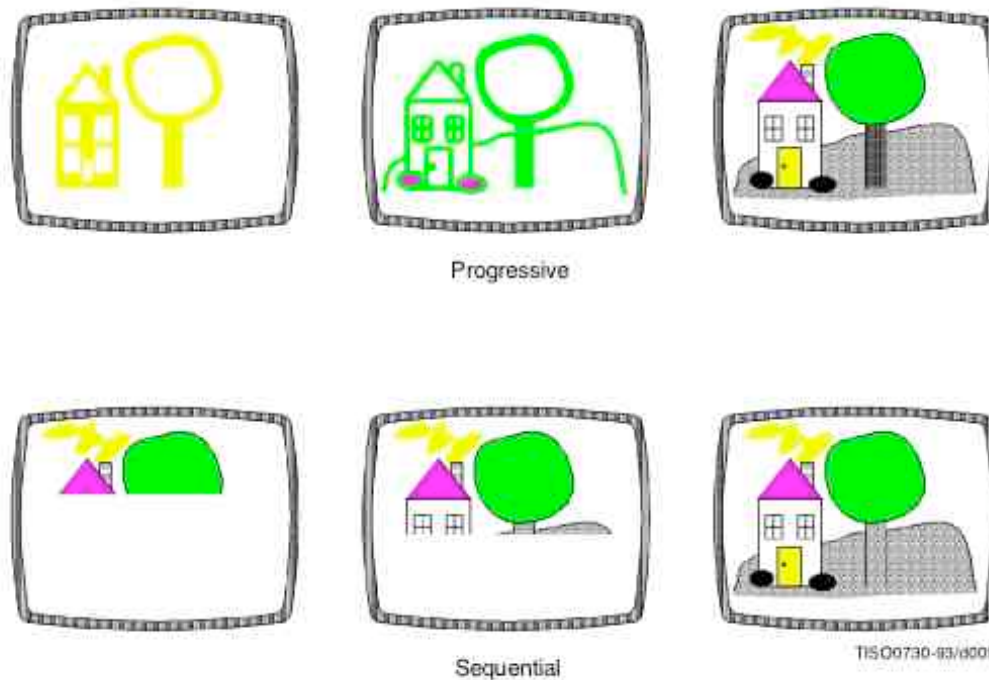


# 6. Licht, Farbe und Bilder

- 6.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 6.2 Farbmodelle
- 6.3 Raster-Bilddatenformate
- 6.4 Verlustbehaftete Kompression bei Bildern
- 6.5 Weiterentwicklungen bei der Bildkompression
  - Progressives und hierarchisches JPEG 
  - Wavelet-basierte Verfahren (verlustbehaftet)
  - Prädiktionsbasierte Verfahren (verlustfrei)

# Progressive Kodierung

- Ein Durchlauf (*scan*) durch Bilddaten kann Verschiedenes bewirken:
  - Ausgabe einer Komponente des Bildes
  - Ausgabe einer unscharfen Vorversion des Bildes
- *Progressive Coding* verbessert die Bildqualität in aufeinander folgenden *scans*.



# Progressive Kodierung durch Spektralselektion

- 8x8-Block von DCT-Koeffizienten
  - Zick-Zack-Reihenfolge geht von niedrigen Frequenzen (wenig Detail) zu hohen Frequenzen (viel Detail).
- *Band*: Teilintervall der Bildfrequenzen
  - als Intervall der DCT-Koeffizienten
- Je Band ein separater *scan*
  - Bandgrenzen im *scan header* angegeben

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Beispiel: 5 Bänder (d.h. 5 *scans*)

- Band 1: DCT-Koeffizient 0 (DC)
- Band 2: DCT-Koeffizienten 1 – 2
- Band 3: DCT-Koeffizienten 3 – 14
- Band 4: DCT-Koeffizienten 15 – 42
- Band 5: DCT-Koeffizienten 43 – 63

# Progressive Kodierung durch Bit Plane Approximation

- Koeffizienten werden zunächst mit geringerer Präzision übertragen
  - Division mit Zweierpotenz bzw. Rechts-Shift (*point transform*)
  - Definition der verwendeten Transformation im *scan header*
- Fehlende Bits werden in weiteren *scans* nachgeliefert

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Beispiel: 6 *scans*

Scan 1: DCT-Koeffizient 0 (DC)

Scan 2: Bits 4 – 7 der DCT-Koeffizienten 1 – 63 (d.h. der AC-Koeffizienten)

Scan 3: Bit 3 der AC-Koeffizienten

Scan 4: Bit 2 der AC-Koeffizienten


Scan 5: Bit 1 der AC-Koeffizienten

Scan 6: Bit 0 der AC-Koeffizienten

# JPEG Progressive Coding

- Einfachste Variante:
  - Ausschließlich Spektralselektion
  - Relativ schlechte Bildqualität in den Zwischenstufen
- Flexiblere Variante:
  - Spektralselektion und sukzessive Approximation
  - Je Band der Spektralselektion:
    - mehrere Scans mit bitweiser Verfeinerung der Auflösung
    - Endqualität und Kompression gleichwertig zu sequentieller Codierung
- Praktische Bedeutung:
  - Progressive JPEG findet nur geringe Akzeptanz
  - Konzepte sind dennoch interessant
- Hierarchisches JPEG:
  - Differentielle Codierung von Folgebildern besserer Auflösung
  - Fast ungebräuchlich

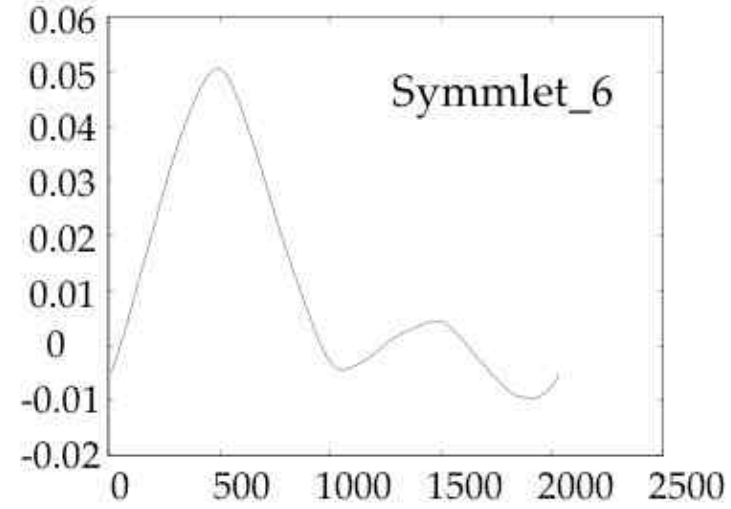
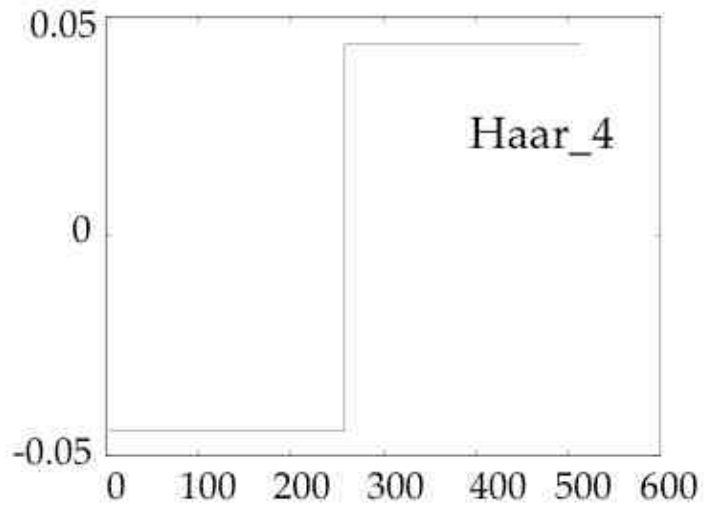
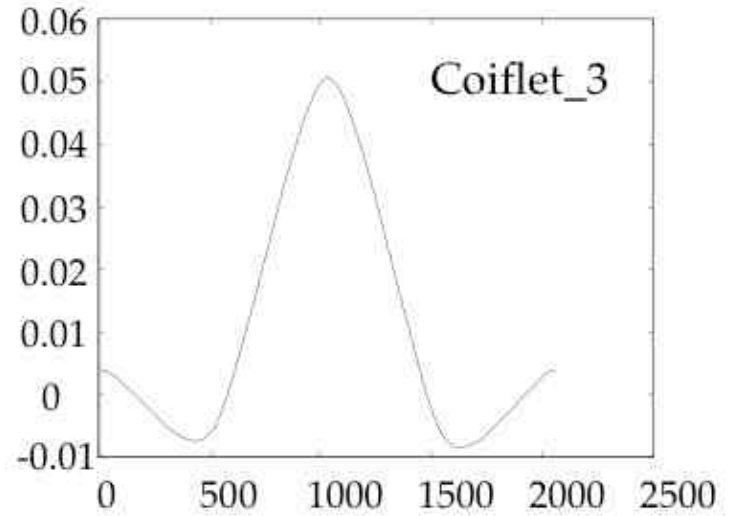
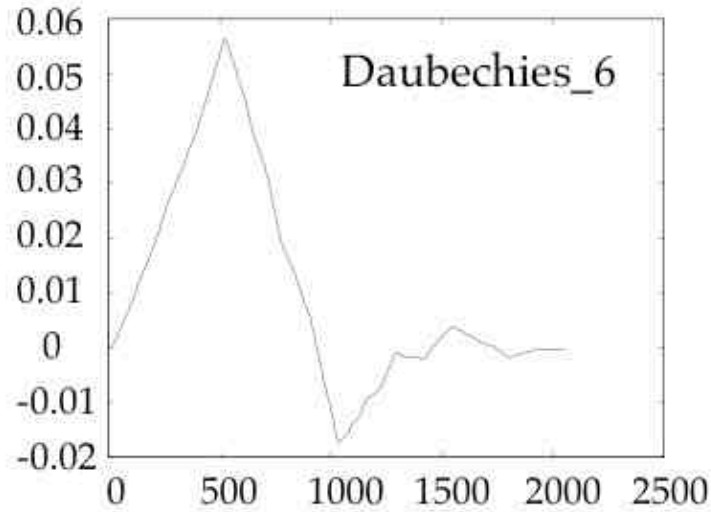
# 6. Licht, Farbe und Bilder

- 6.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 6.2 Farbmodelle
- 6.3 Raster-Bilddatenformate
- 6.4 Verlustbehaftete Kompression bei Bildern
- 6.5 Weiterentwicklungen bei der Bildkompression
  - Progressives und hierarchisches JPEG
  - Wavelet-basierte Verfahren (verlustbehaftet) 
  - Prädiktionsbasierte Verfahren (verlustfrei)

# Wavelets

- Wavelets sind spezielle mathematische Funktionen, die sich als „Basis“ zur Erzeugung beliebiger Wellenformen besonders gut eignen.
  - „kompakte Unterstützung“, d.h. null außerhalb eines endlichen Intervalls
  - unendlich oft differenzierbar
  - orthonormale Basis
- Ermöglichen Zeit- bzw. Ortsanalyse *und* Frequenzanalyse
- Historische Perspektive:
  - Erste Ideen ca. um 1900 (Haar)
  - Grosse Entwicklungssprünge ab 1960, insbesondere in den 80ern (Mallat, Daubechies)
  - Anwendungen in verschiedenen Disziplinen:  
Beispiele: Fingerabdruckerkennung, Analyse von Turbulenzen, Erdbebenvorhersage ... und Bildkompression

# Beispiele von Wavelets



# Frequenz- und Zeit/Ortanalyse

- Klassische Transformation in den Frequenzraum (Fourier, DCT):
  - Sinus- und Cosinus-Funktionen wiederholen sich periodisch
  - Fourier-Transformation arbeitet sogar mit periodischer Fortsetzung nicht-periodischer Funktionen
  - Analyse bezieht sich immer auf die gesamte Zeitachse (z.B. bei Ton) bzw. gesamte Ortsachse (bei Bild)
- Gleichzeitige präzise Auflösung in der Zeit/Ortachse und in der Frequenz nicht erreichbar
  - Abhilfe z.B. bei JPEG und MP3: Einteilung in kleine Blöcke/Zeitfenster
  - Probleme bei Blockgrenzen und bei Diskontinuitäten
- Wavelets:
  - erlauben eine Mischung aus langen Wavelet-Funktionen für Frequenzanalyse und kurzen, hochfrequenten Wavelet-Funktionen für Zeit/Ortanalyse

# Grundprinzip der Wavelet-Analyse

- Bild wird zerlegt in
  - Tiefe Frequenzanteile (Tiefpass)
  - Hohe Frequenzanteile (Hochpass) = Details
- Zeilen- und spaltenweise Analyse mit Filtern
  - Vier Bilder:  
(TP-hor + TP-vert, HP-hor + TP-vert, TP-hor + HP-vert, HP-hor + HP vert)
- Subsampling: Jeder zweite Koeffizient verworfen in Zeilen und Spalten
- Rekursive Fortsetzung mit dem Teilbild “TP-hor + TP-vert”  
(= Tiefpass-gefiltertes Bild)
- Verlustfreie Transformation!



# Kompression bei Wavelet-Transformation

- Die hohen Frequenz-Koeffizienten können quantisiert (gerundet) werden
  - Basis der Darstellung ist das niederfrequent gefilterte Bild
- Flexibler Kompressionsgrad
  - Mehr hohe Frequenzen quantisiert: Bild beruht auf stärkerer Tiefpass-Filterung, also schlechtere Qualität
  - Verschiedene Kompressionsraten aus einer Basisinformation
- Kompression führt kaum zu Block-Artefakten



# Beispiel für Tiefpass und Hochpass

- (Nach Heyna/Briede/Schmidt)
- Haar-Transformation:
  - $TP(n) = 0.5 (x(n) + x(n+1))$
  - $HP(n) = 0.5 (x(n) - x(n+1))$

	x(0)	x(1)	x(2)	x(3)	x(4)
Original-Pixelwerte	26	8	17	3	5
TP-Koeffizienten	17	12.5	10	8	
HP-Koeffizienten	9	-4.5	7	-1	
Subsampling TP-Koeff.	17		10		
Subsampling HP-Koeff.	9		7		

Rekonstruktion:

$$x(0) = TP(0) + HP(0) = 17 + 9 = 26$$

$$x(1) = TP(0) - HP(0) = 17 - 9 = 8 \text{ usw.}$$

# JPEG2000

- März 1997
  - Start der Entwicklung eines verbesserten Standards für Bildkompression „JPEG 2000“ („j2k“)
  - Bessere verlustbehaftete Kompression als JPEG (mit Wavelets)
  - Leistungsfähige verlustfreie Kompression als Option
  - In Auflösung und Präzision lokal skalierbare Bilder
  - Wahlfreier Zugriff auf Bildteile in höherer Auflösung
  - Einbeziehung von Schwarz/weiss-Bildern
- Final Draft International Standard August 2000
  - Draft ISO 15444-1 und ITU Rec. T.800
  - Entwicklung seit 2000 nur noch sehr langsam
  - Praktischer Einsatz z.B. im medizinischen Bereich, im neuen Reisepass
- Grundarchitektur wie bei JPEG:
  - Forwärtstransformation (Discrete Wavelet Transform DWT)
  - Quantisierung (oder verlustfrei)
  - Entropiecodierung (hier mit arithmetischer Codierung)

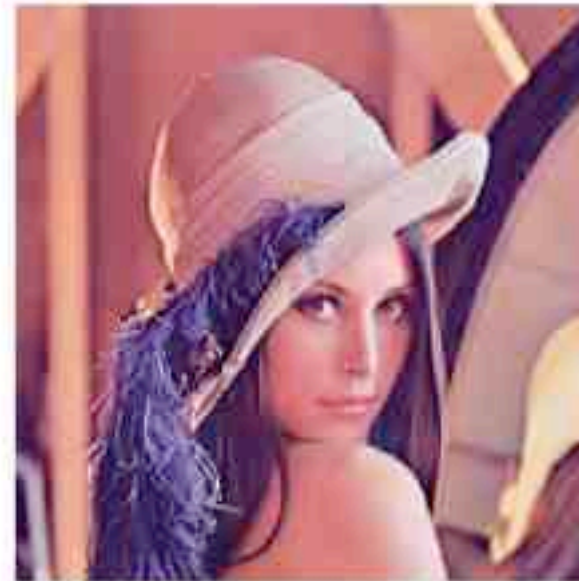
# Qualitätsunterschied JPEG – JPEG2000

AWARE'S JPEG2000 SDK DEMO  
JPEG vs JPEG 2000 Comparison

JPEG  
PSNR =  
24.9



JPEG 2000  
PSNR =  
28.7



20:1 40:1 60:1 80:1 100:1  
Compression Ratio (click to select)


Quelle: [www.aware.com](http://www.aware.com)

# Region-of-Interest (ROI) Coding in JPEG2000

- Bestimmte (beliebig geformte) Regionen des Bildes oft „interessanter“ als der Hintergrund (*region of interest ROI*)
- ROI kann mit besserer Qualität codiert werden als der Hintergrund
- Sogenannter „MAXSHIFT“-Algorithmus platziert die ROI an einer Stelle (höhere *bitplane*), wo sie zeitlich *vor* dem Hintergrund decodiert wird



# 6. Licht, Farbe und Bilder

- 6.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 6.2 Farbmodelle
- 6.3 Raster-Bilddatenformate
- 6.4 Verlustbehaftete Kompression bei Bildern
- 6.5 Weiterentwicklungen bei der Bildkompression
  - Progressives und hierarchisches JPEG
  - Wavelet-basierte Verfahren (verlustbehaftet)
  - Prädiktionsbasierte Verfahren (verlustfrei) 

# Prädiktoren für JPEG (Lossless Operation Mode)

	c	b	
	a	x	

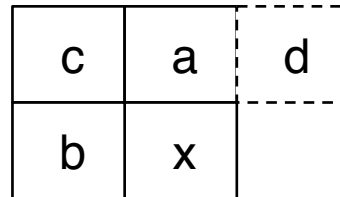
- Prädiktor = Formel zur Berechnung des x-Wertes aus dem Kontext (hier Werte für a, b, c)
  - Prädizierter Wert  $P_x$  und tatsächlicher Wert  $R_x$
  - Übertragen werden: Prädiktor-Regel und Differenzen  $P_x - R_x$
  - Je besser  $P_x$  mit  $R_x$  übereinstimmt, desto häufiger treten Null und sehr niedrige Differenzen auf: Gute Kompression mit Entropiecodierung möglich

- Eindimensionale Prädiktoren:
  - $P_x = R_a$ ,  $P_x = R_b$ ,  $P_x = R_c$
- Zweidimensionale Prädiktoren:
  - $P_x = (R_a + R_b)/2$
  - $P_x = R_a + (R_b - R_c)/2$
  - $P_x = R_b + (R_a - R_c)/2$
  - $P_x = R_a + R_b - R_c$  ("Paeth-Prädiktor")

# JPEG-LS

- 1998:
  - Final *Draft* International Standard ISO 14495-1 / ITU Rec. T.87
- Verlustfreie und fast verlustfreie Kompression von Standbildern
  - Hohe Kompressionsrate, geringe Komplexität
  - Unabhängig vom JPEG-Standard
- Basiert auf „LOCO-I“ (Low Complexity Image Compression)
  - HP Labs: M. Weinberger, G. Seroussi, G. Sapiro
  - Bessere Einbeziehung des Kontextes in Prädiktion
  - Einfache Kantenentdeckung möglich
  - Entropie-Codierung: Adaptive Variante der Golomb-Rice-Kodierung
- Frühere Algorithmen: entweder wesentlich komplexer oder benutzten arithmetische Entropie-Kompression.
- Derzeit noch kaum im praktischen Einsatz

# Prädiktionsmodell von JPEG-LS

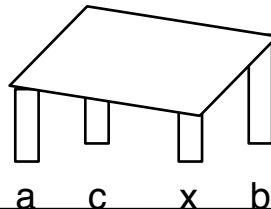


5	1	1	
4	4	x	(x = 1)
2	2	1	
5	6	x	(x = 6)
3	4	5	
1	2	x	(x = 3)

- $P_x = \min(R_a, R_b)$  falls  $R_c \geq \max(R_a, R_b)$
- $P_x = \max(R_a, R_b)$  falls  $R_c \leq \min(R_a, R_b)$
- $P_x = R_a + R_b - R_c$  sonst
- Wert von d für „Kontexterkenkung“ benutzt

- Einfache Kantenerkennung (*median edge detector*):

- Vertikale Kante links von x: führt (oft) zur Wahl von  $P_x = R_a$
- Horizontale Kante oberhalb von x: führt (oft) zur Wahl von  $P_x = R_b$
- Keine Kante erkannt:  $P_x$  entsprechend einer Ebene durch  $R_a, R_b, R_c$



# Verwendung von Kontextinformation

c	a	d
b	x	

- Kontextbestimmung
  - $g1 = R_d - R_b$ ,  $g2 = R_b - R_c$ ,  $g3 = R_c - R_a$
  - Einteilung in 365 verschiedene Kontextsituationen
- Adaptive Korrektur der Prädiktion:
  - Je Kontext:
    - » Zahl der Kontextvorkommen mitrechnen
    - » Bisherige Vorhersagefehler kumulieren
  - Prädiktionswert um bisherigen durchschnittlichen Vorhersagefehler korrigieren
- Kontextinformation auch benutzt zur Wahl des Code-Typs in spezieller Entropiecodierung

# Golomb-Rice Codierung

- Grundidee: Entropie-Codierung für (Ganz-)Zahlwerte mit geometrischer Häufigkeitsverteilung
  - Niedrige Werte häufiger und deshalb kürzer codiert
  - Trifft bei den Restwerten (Residuen) von Prädiktion meist zu
- Golomb-Codierung (Solomon Golomb, 60er Jahre):
  - Bestimme Quotient  $q$  und Rest  $r$  zu einem festen Divisor  $M$
  - Codiere  $q$  als Unärzahl,  $r$  als abgeschnittene Binärzahl
- Golomb-Rice-Codierung:
  - Divisor  $M$  ist Zweierpotenz
- Praktischer Algorithmus (Golomb-Rice-Codierung der Ordnung  $k$ ):
  - Teile  $n$  durch  $2^k$ , Quotient ist  $q$ , Rest ist  $r$
  - Bilde Codewort aus: ( $q$ -mal **0**), (1-mal **L**),  $k$  letzte Bits der Binärform von  $n$
- Beispiel ( $k = 2$ ):
  - $n = 3$ :  $q = 0$ ,  $r = 3$ , Code ist **LLL**
  - $n = 13$ :  $q = 3$ ,  $r = 1$ , Code ist **000L0L**