
Echtzeit–Videoverarbeitung

Transformation

Marcus Barkowsky

Überblick

Allgemeines

- Aufgabe
- Länge der Transformation

Arten von Transformationen

- Trigonometrische Transformationen
- Sub-Band Zerlegung
- Wavelets
- Fraktale
- Vektor-Quantisierer

Schnelle Algorithmen

- Diskrete Fourier-Transformation
- Diskrete Cosinus-Transformation
- 2D-DCT

Eigenschaften von Transformationen

Aufgabengebiete:

- Bildcodierung
 - Energiekompaktierung bzw. Dekorrelation
 - Repräsentation des Signals mit möglichst wenig Redundanz
 - Möglichst wenig Koeffizienten ungleich 0
 - Staffelung der Signifikanz der Koeffizienten
- Analysevorstufe
 - Eingangssignal in gewünschte Repräsentation bringen, z.B.
 - Fourier–Transformation für Frequenzraumanalyse/bearbeitung
 - Splines für zeitkontinuierliche Analysen

Abhängigkeit von der Signalstatistik:

- Explizit: Transformations–Vorschrift ist signalabhängig
z.B. Karhunen–Loève–Transformation (KLT)
- Implizit: Transformation ist nur für bestimmte Signalarten geeignet
z.B. Diskrete–Cosinus Transformation (DCT)

Anwendungsgebiete

- Bildcodierung
- Bildanalyse
 - Fourier–Transformation
 - Frequenzraum–Analysen
 - Objekterkennung mit Kreuzkorrelation
 - Robuste Lagemessung
 - Skalierungsmessung

Länge der Transformationsbasisfunktionen

Große Ausdehnung (Ganzbildtransformation)

- Detaillierte Auflösung im transformierten Raum, z.B. viele Frequenzlinien, dadurch gute Selektionseigenschaften, z.B. Bandpaß
- Sehr viele Koeffizienten nahe 0
- Hohe Rechenzeitanforderungen

Kleine Ausdehnung (Blocktransformation)

- Gute örtliche Lokalisierungseigenschaften
 - Signifikanz einzelner Koeffizienten sehr gut feststellbar
 - Region-of-Interest Bearbeitung möglich, z.B. unterschiedlich starke Quantisierung der Blöcke
- Nur Blöcke mit hoher Aktivität enthalten viele Koeffizienten ungleich 0
- Blockkantenproblematik

Beispiel DCT



Original

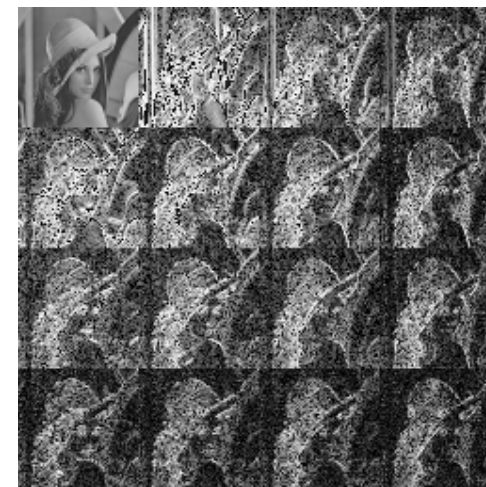


4x4 Block DCT

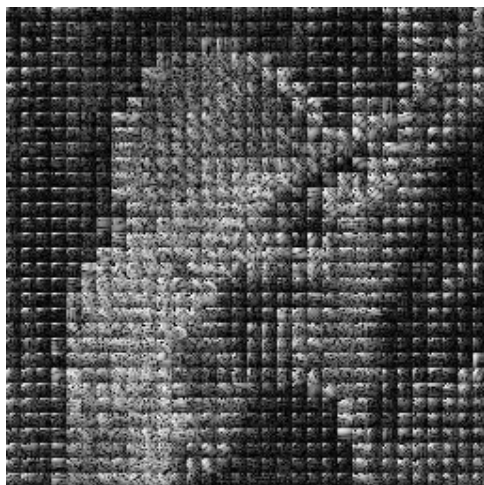
DC	AC 12	AC 13	AC 14	DC	...
AC 21	AC 22	AC 23	AC 24	AC 21	...
AC 31	AC 32	AC 33	AC 34	AC 31	...
AC 41	AC 42	AC 43	AC 44	AC 41	...
DC	AC 12	AC 13	AC 14	DC	...
⋮	⋮	⋮	⋮	⋮	⋮

DC	DC	...	AC 14
DC	DC	...	AC 24
⋮	⋮	⋮	⋮
AC 41	AC 42	...	AC 44

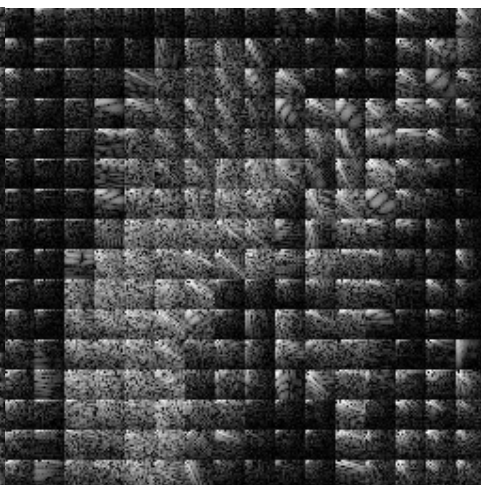
Gruppierung



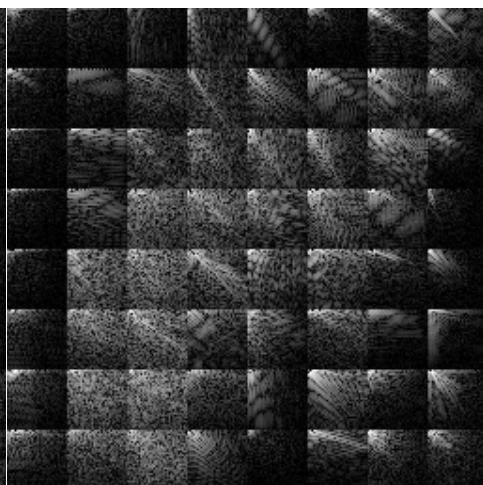
4x4 Block DCT



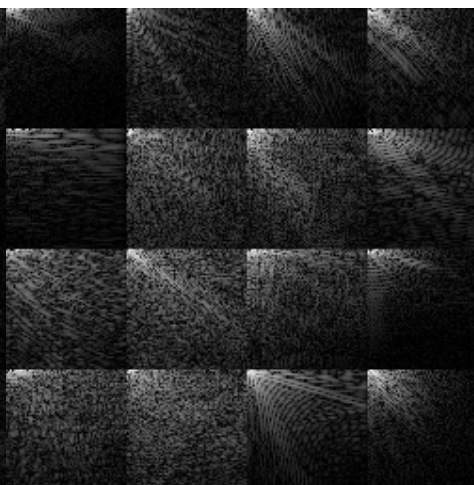
8x8 Block DCT



16x16 Block DCT



32x32 Block DCT



64x64 Block DCT

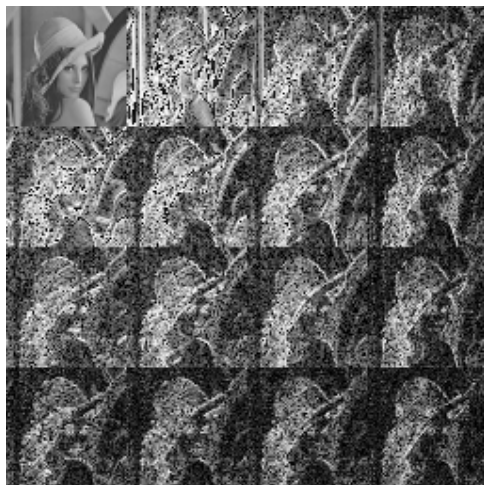
Beispiel DCT



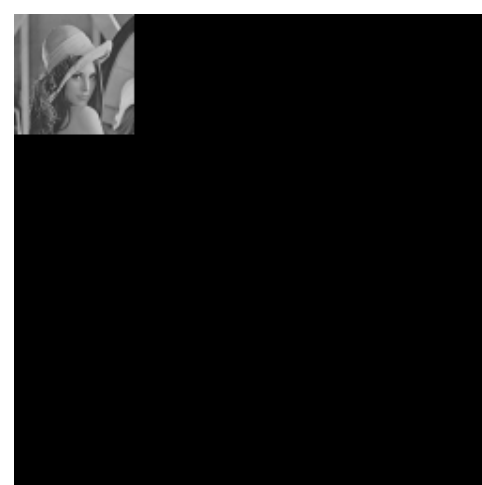
Ganzbild-DCT



Auswahl der Tiefpaß-Koeffizienten



4x4 Block DCT



Auswahl der Tiefpaß-Koeffizienten

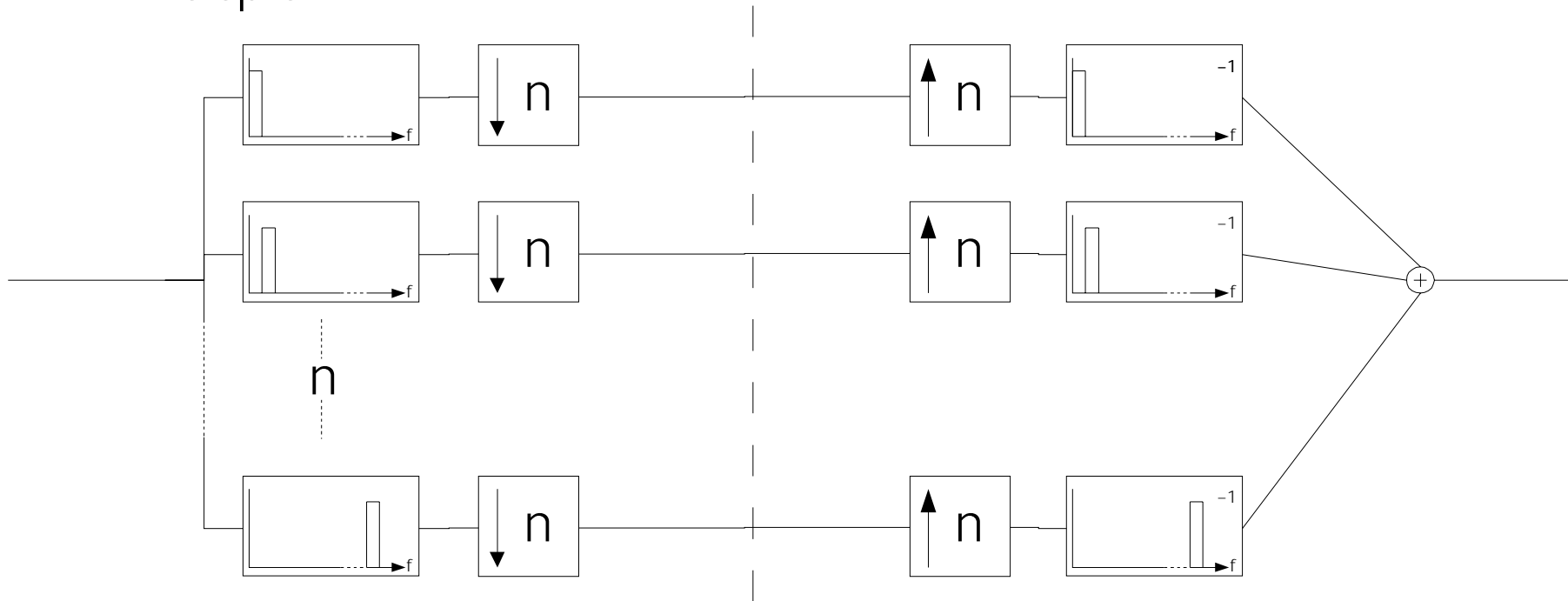


Transformationsklassen (Auswahl)

- Subband–Zerlegung
- Wavelet
- Fraktale
 - 1 dimensional
 - 2 dimensional
- Vektorquantisierer
- Trigonometrische Transformationen
 - Diskrete Fourier Transformation (DFT)
 - Diskrete Hartley Transformation (DHT)
 - Diskrete Sinus Transformation (DST)
 - Diskrete Cosinus Transformation (DCT)

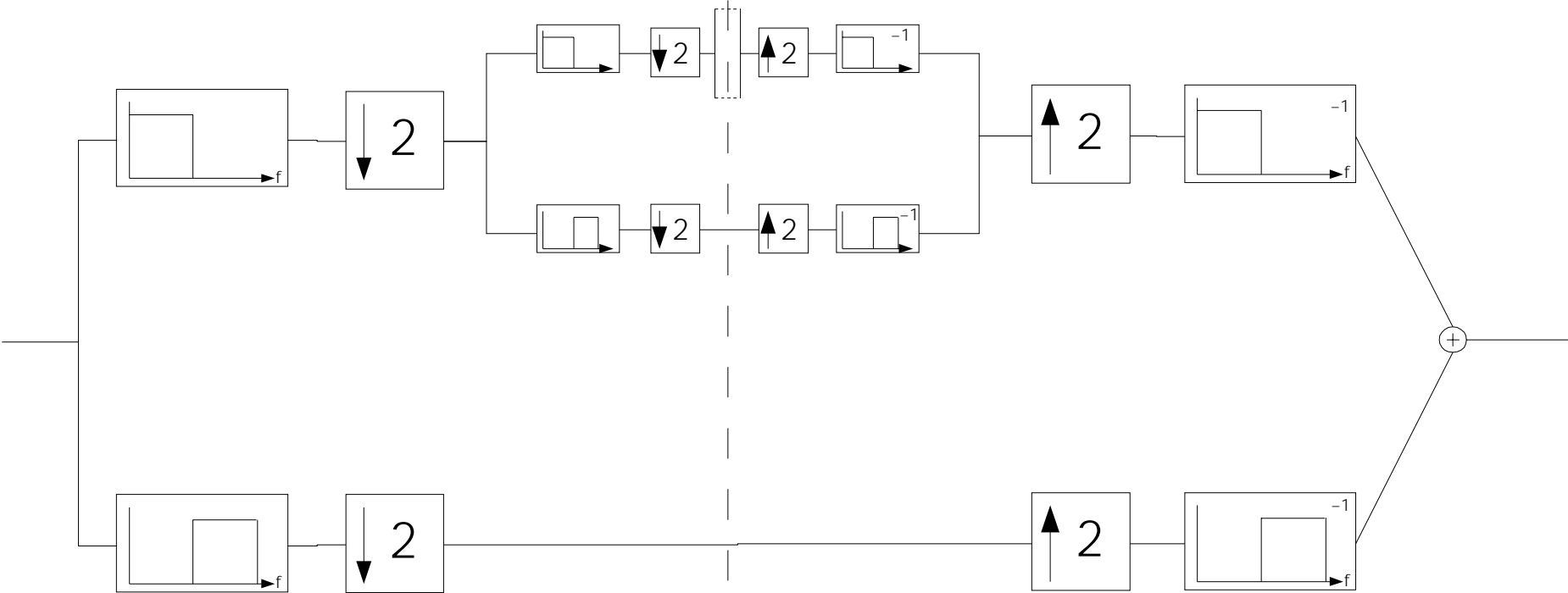
Subband-Zerlegung

Beispiel:

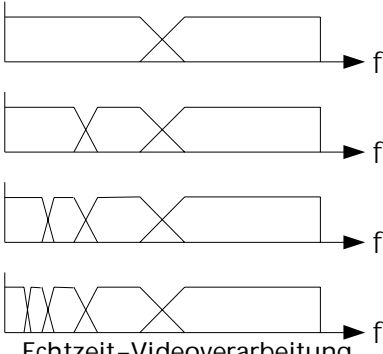


Problem: Auswahl der Filterfunktionen zur exakten Rekonstruktion ohne Aliasing, z.B. mittels Quadratur-Spiegel-Filtern (QMF)

Kaskadierte Subband-Zerlegung



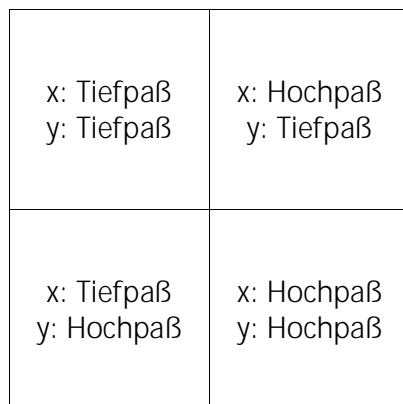
In dieser Form:
Oktavbandzerlegung



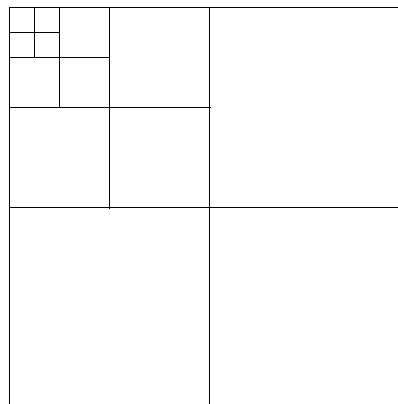
Echtzeit-Videoverarbeitung

Wavelet

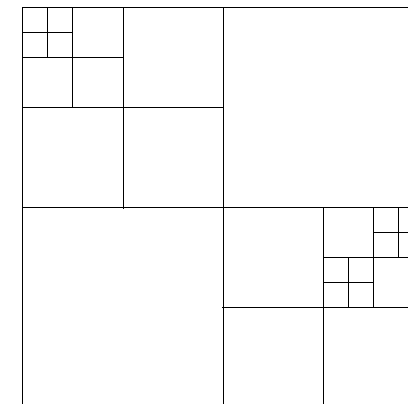
- Meist Oktavbandzerlegung mit speziellen Filterfunktionen
- Optional bei erhöhter Aktivität in Hochpaßanteilen weitere Zerlegung
- Verwendung spezieller Verfahren zur Gruppierung der Koeffizienten (z.B. Zero-Tree)
- Erweiterung auf zeitliche Komponente möglich (z.B. 3DSLCCA 3D-significance-linked connected component analysis)



1. Zerlegungsschritt



Vollständige Zerlegung



Optionale HP-Zerlegung

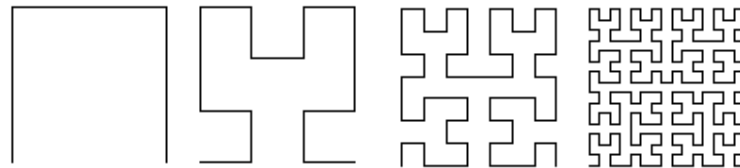
Fraktale Verfahren

Idee: Auffinden von Selbstähnlichkeit im Bild
meist aus einem unterabgetasteten Bild

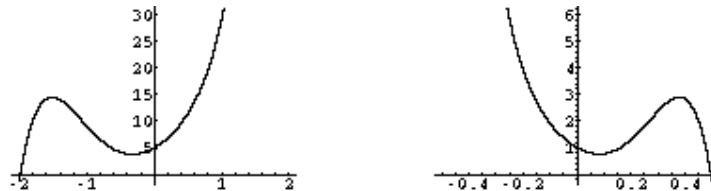
Ziel: möglichst kleine Auswahl unterschiedlicher Signalformen

Eindimensional:

- Reduktion des 2D-Bildes auf 1D, z.B. mittels Hilbert-Kurve



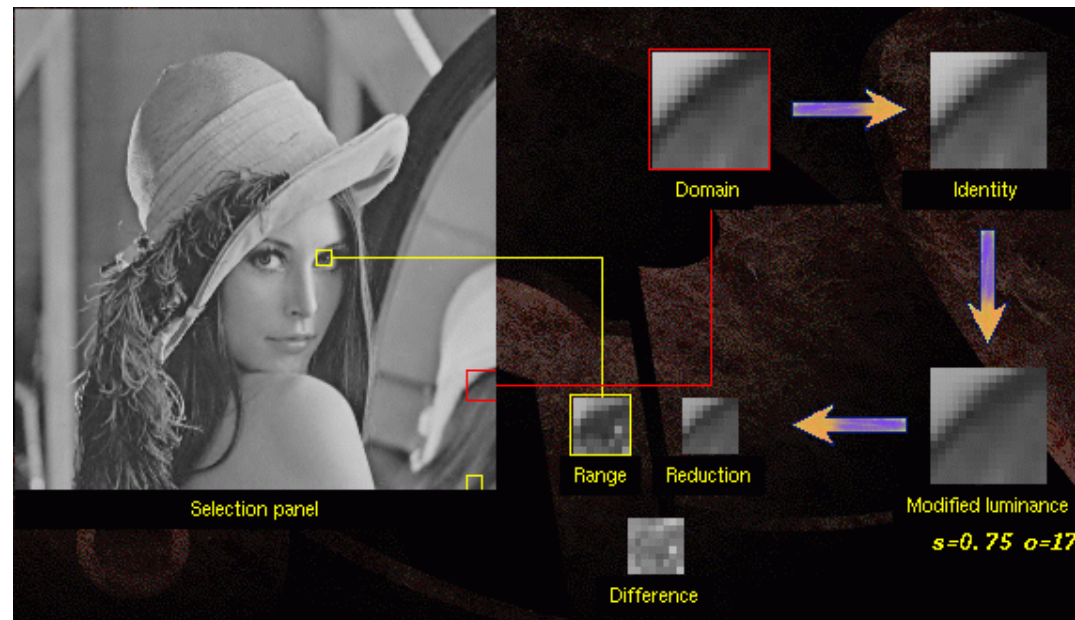
- Aufteilung des Signals in einzelne Abschnitte und Auffinden von Ähnlichkeiten, Skalierung und Streckung u.ä. möglich



Fraktale Verfahren

Zweidimensional:

- Auffinden gleicher Bildbereiche
- Freiheitsgrade:
 - Helligkeit, Kontrast, ...
 - Drehung, Skalierung, affine Transformation
- Optional: Zwischen En- und Decoder vereinbar, fester Mustervorrat
- Beispiel vom Institut Eurecom (Frankreich)

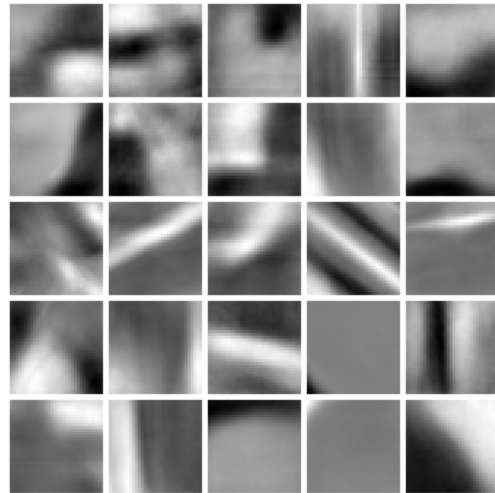


Vektorquantisierer

Idee: Die Menge aller Muster, aus denen ein Bild zusammengesetzt wird, ist begrenzt

Ziel: Aus einem festgelegten Mustervorrat diejenigen auswählen, die das Bild am besten approximieren

Beispiel: Ansatz von Strathclyde (H.26L)

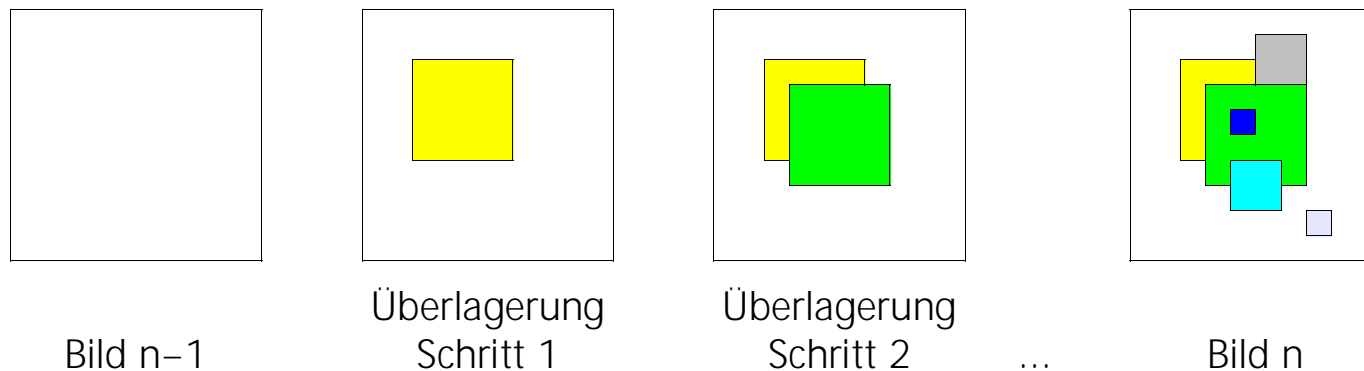


Auszug aus dem Mustervorrat für 32x32
(generiert durch neuronales Netz)

Strathclyde Compression Transform (SCT)

Verfahren:

- Letztes Bild als Startwert verwenden
- Successive Überlagerung (Collage) der im Mustervorrat vorhandenen Muster oder von Bildteilen aus vergangenen Bildern



- Einschränkungen: Mustervorrat enthält quadratische Blöcke der Größe 32x32, 16x16 und 8x8
- Falls notwendig, wird an kritischen Stellen die Differenz Wavelet-codiert (nur bei Bitraten >128kBit/s)
- Besonderheit: Es wird nicht zwingend der Bereich mit der größten Abweichung verbessert, sondern das hinzugefügte Muster soll die Gesamtbildqualität (PSNR) am stärksten verbessern.

Trigonometrische Transformationen

Aufbau für diskrete Werte mit Transformationsfunktion $t(x,n)$

- 1D:

$$F(x) = \frac{2}{N} \cdot c(x) \cdot \sum_{n=0}^{N-1} t(n,x) \cdot f(n)$$

$$f(x) = \sum_{n=0}^{N-1} c(n) \cdot t^{-1}(n,x) \cdot F(n)$$

$$\text{mit: } c(n) = \begin{bmatrix} \frac{1}{\sqrt{2}} & \text{für } n=0 \\ 1 & \text{sonst} \end{bmatrix}$$

- Anmerkung: Bei der inversen Transformation werden meist Indizes vertauscht und $c(n)$ muß in der Summe berücksichtigt werden!

Trigonometrische Transformationen

2D:

$$F(x_1, x_2) = \frac{4}{N_1 \cdot N_2} \cdot c(x_1) \cdot c(x_2) \cdot \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} t(n_1, x_1) \cdot t(n_2, x_2) \cdot f(n_1, n_2)$$

$$f(x_1, x_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} c(n_1) \cdot c(n_2) \cdot t^{-1}(n_1, x_1) \cdot t^{-1}(n_2, x_2) \cdot F(n_1, n_2)$$

$$\text{mit: } c(n) = \begin{bmatrix} \frac{1}{\sqrt{2}} & \text{für } n=0 \\ 1 & \text{sonst} \end{bmatrix}$$

Funktionen für trigonometrische Transformationen

Diskrete Fourier Transformation (DFT)

$$t(n,x) = e^{\frac{2 \cdot \pi \cdot i \cdot x \cdot n}{N}} \quad t^{-1}(n,x) = e^{\frac{-2 \cdot \pi \cdot i \cdot x \cdot n}{N}} \quad \text{mit } i^2 = -1$$

Diskrete Hartley Transformation (DHT)

$$t(n,x) = \cos\left(\frac{2 \cdot \pi \cdot x \cdot n}{N}\right) + \sin\left(\frac{2 \cdot \pi \cdot x \cdot n}{N}\right) \quad t^{-1}(n,x) = t(x,n)$$

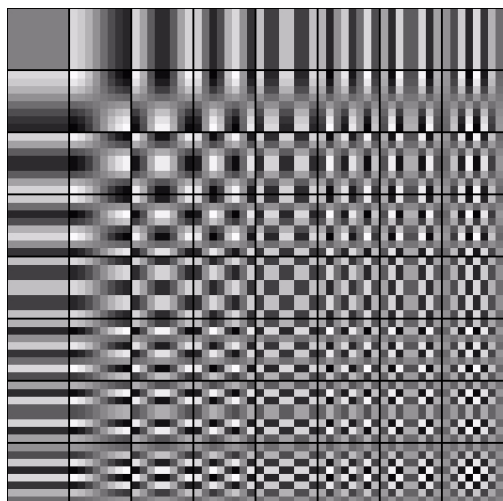
Diskrete Sinus Transformation (DST)

$$t(n,x) = \sin\left(\frac{x \cdot (2 \cdot n + 1) \cdot \pi}{2N}\right) \quad t^{-1}(n,x) = t(x,n)$$

Diskrete Cosinus Transformation (DCT)

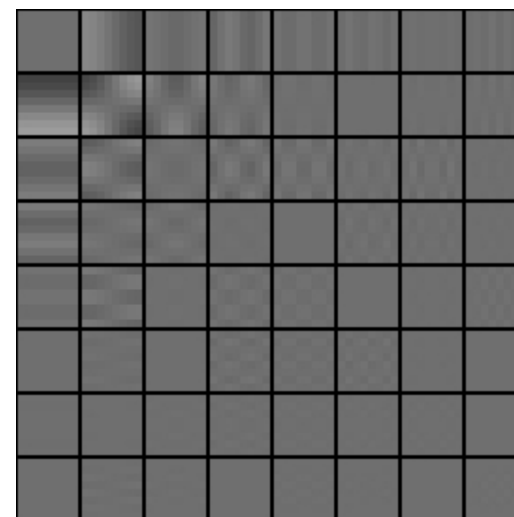
$$t(n,x) = \cos\left(\frac{x \cdot (2 \cdot n + 1) \cdot \pi}{2N}\right) \quad t^{-1}(n,x) = t(x,n)$$

Visualisierung der inversen Transformation IDCT



55	82	51	-63	41	-43	-25	-37
-97	-87	70	55	-31	4	22	-33
69	-71	-41	58	48	-37	42	-17
68	47	-46	-21	-6	36	-31	-15
41	60	16	43	-39	10	12	-34
8	32	-4	41	34	-39	8	-7
13	-11	-19	26	22	15	-21	-3
7	30	-16	4	21	13	-5	-16

=



Basisfunktionen

Koeffizienten

gewichtete Basisfunktionen

Successiver Aufbau mittels Basisfunktionen

1 Koeffizient



2x2 Koeffizienten



3x3 Koeffizienten



8x8 Koeffizienten



Schnelle Algorithmen

- DFT
 - Zerlegung nach Danielson–Lanczos und Cooley–Tukey
- DCT
 - Chen 1977
 - Verfeinerungen
 - Tran: Lifting Steps 1999
 - Tran: Multiplierless DCT
 - 2D–DCT

Diskrete Fourier Transformation

$$F(k) = \sum_{j=0}^{N-1} e^{\frac{2\pi i \cdot k \cdot j}{N}} f(j)$$

$$F(k) = \sum_{j=0}^{N/2-1} e^{\frac{2\pi i \cdot k \cdot (2j)}{N}} f(2j) + \sum_{j=0}^{N/2-1} e^{\frac{2\pi i \cdot k \cdot (2j+1)}{N}} f(2j+1)$$

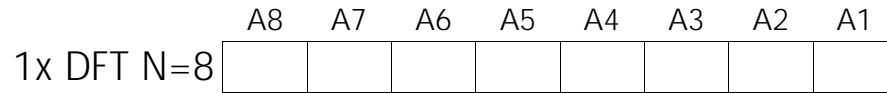
j gerade j ungerade

$$F(k) = \sum_{j=0}^{N/2-1} e^{\frac{2\pi i \cdot k \cdot (2j)}{N}} f(2j) + \sum_{j=0}^{N/2-1} e^{\frac{2\pi i \cdot k \cdot (2j)}{N}} \cdot e^{\frac{2\pi i \cdot k}{N}} f(2j+1)$$

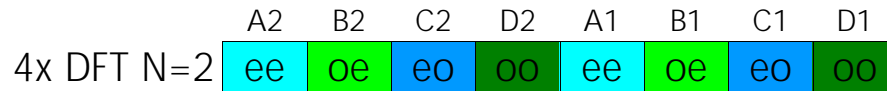
$$F(k) = \sum_{j=0}^{N/2-1} e^{\frac{2\pi i \cdot k \cdot j}{N/2}} f(2j) + e^{\frac{2\pi i \cdot k}{N}} \cdot \sum_{j=0}^{N/2-1} e^{\frac{2\pi i \cdot k \cdot j}{N/2}} \cdot f(2j+1)$$

DFT der Länge N/2 über alle j gerade Konstante DFT der Länge N/2 über j ungerade

Diskrete Fourier Transformation



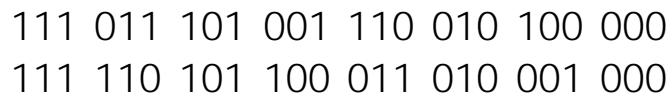
Aufspaltung in odd (o) und even(e)



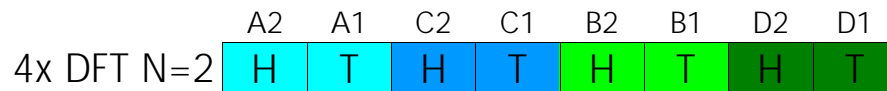
Aufspaltung der Einzel-DFTs



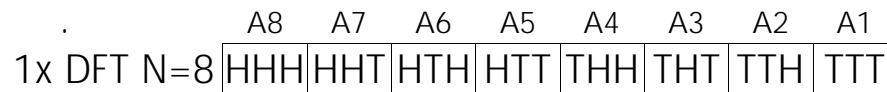
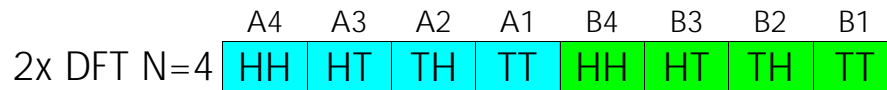
1D-DFT: Funktionswert=
transformierter Wert



Koeffizienten sind in dieser Ordnung
Wenn die Bits von hinten nach vorn
gelesen werden, ist die Reihenfolge
korrekt



Umsortierung vor der Transformation



ergibt korrekte Reihenfolge nach der
Transformation

Fast Fourier Transformation (FFT)

Vorteil:

- Reduktion von N^2 Operationen auf $N \log_2 N$
Beispiel $N=256$: Von 65536 auf 2048 → Faktor 32
 $N=65536$: Faktor 4096
- Konstanten müssen nur einmal berechnet werden für jedes N
- Ausgangspunkt für viele weitere Algorithmen

Nachteil:

- Nur sinnvoll anwendbar für $N=2^x$

DCT nach Chen

Formeldarstellung für 8x8:

$$F(x) = \frac{2}{N} \cdot c(x) \cdot \sum_{n=0}^{N-1} \cos\left[\frac{(2n+1) \cdot x \cdot \pi}{2N}\right] \cdot f(n)$$

Matrixdarstellung für 8x8:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos\left[\frac{\pi}{16}\right] & \cos\left[\frac{3\pi}{16}\right] & \cos\left[\frac{5\pi}{16}\right] & \cos\left[\frac{7\pi}{16}\right] & \cos\left[\frac{9\pi}{16}\right] & \cos\left[\frac{11\pi}{16}\right] & \cos\left[\frac{13\pi}{16}\right] & \cos\left[\frac{15\pi}{16}\right] \\ \cos\left[\frac{\pi}{8}\right] & \cos\left[\frac{3\pi}{8}\right] & \cos\left[\frac{5\pi}{8}\right] & \cos\left[\frac{7\pi}{8}\right] & \cos\left[\frac{9\pi}{8}\right] & \cos\left[\frac{11\pi}{8}\right] & \cos\left[\frac{13\pi}{8}\right] & \cos\left[\frac{15\pi}{8}\right] \\ \cos\left[\frac{3\pi}{16}\right] & \cos\left[\frac{9\pi}{16}\right] & \cos\left[\frac{15\pi}{16}\right] & \cos\left[\frac{21\pi}{16}\right] & \cos\left[\frac{27\pi}{16}\right] & \cos\left[\frac{33\pi}{16}\right] & \cos\left[\frac{39\pi}{16}\right] & \cos\left[\frac{45\pi}{16}\right] \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos\left[\frac{5\pi}{16}\right] & \cos\left[\frac{15\pi}{16}\right] & \cos\left[\frac{25\pi}{16}\right] & \cos\left[\frac{35\pi}{16}\right] & \cos\left[\frac{45\pi}{16}\right] & \cos\left[\frac{55\pi}{16}\right] & \cos\left[\frac{65\pi}{16}\right] & \cos\left[\frac{75\pi}{16}\right] \\ \cos\left[\frac{3\pi}{8}\right] & \cos\left[\frac{9\pi}{8}\right] & \cos\left[\frac{15\pi}{8}\right] & \cos\left[\frac{21\pi}{8}\right] & \cos\left[\frac{27\pi}{8}\right] & \cos\left[\frac{33\pi}{8}\right] & \cos\left[\frac{39\pi}{8}\right] & \cos\left[\frac{45\pi}{8}\right] \\ \cos\left[\frac{7\pi}{16}\right] & \cos\left[\frac{21\pi}{16}\right] & \cos\left[\frac{35\pi}{16}\right] & \cos\left[\frac{49\pi}{16}\right] & \cos\left[\frac{63\pi}{16}\right] & \cos\left[\frac{77\pi}{16}\right] & \cos\left[\frac{91\pi}{16}\right] & \cos\left[\frac{105\pi}{16}\right] \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \end{bmatrix}$$

DCT nach Chen

Aufgabe: Vereinfachung der Matrix

Verfahren: Aufspaltung der Matrix in schwach besetzte Teilmatrizen

Ansatz: $\lambda(x) = \cos\left(\frac{x \cdot \pi}{16}\right)$

$$\begin{bmatrix} \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] \\ \lambda[1] & \lambda[3] & \lambda[5] & \lambda[7] & \lambda[9] & \lambda[11] & \lambda[13] & \lambda[15] \\ \lambda[2] & \lambda[6] & \lambda[10] & \lambda[14] & \lambda[18] & \lambda[22] & \lambda[26] & \lambda[30] \\ \lambda[3] & \lambda[9] & \lambda[15] & \lambda[21] & \lambda[27] & \lambda[33] & \lambda[39] & \lambda[45] \\ \lambda[4] & \lambda[12] & \lambda[20] & \lambda[28] & \lambda[36] & \lambda[44] & \lambda[52] & \lambda[60] \\ \lambda[5] & \lambda[15] & \lambda[25] & \lambda[35] & \lambda[45] & \lambda[55] & \lambda[65] & \lambda[75] \\ \lambda[6] & \lambda[18] & \lambda[30] & \lambda[42] & \lambda[54] & \lambda[66] & \lambda[78] & \lambda[90] \\ \lambda[7] & \lambda[21] & \lambda[35] & \lambda[49] & \lambda[63] & \lambda[77] & \lambda[91] & \lambda[105] \end{bmatrix}$$

DCT nach Chen

mit $\cos(2\pi + x) = \cos(x)$ folgt: $\lambda(32 + x) = \lambda(x)$

dann $\cos(2\pi - x) = \cos(x)$ folgt: $\lambda(32 - x) = \lambda(x)$

und $\cos(\pi + x) = -\cos(\pi - x)$ folgt: $\lambda(16 + x) = -\lambda(16 - x)$

$$\begin{bmatrix} \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] \\ \lambda[1] & \lambda[3] & \lambda[5] & \lambda[7] & -\lambda[7] & -\lambda[5] & -\lambda[3] & -\lambda[1] \\ \lambda[2] & \lambda[6] & -\lambda[6] & -\lambda[2] & -\lambda[2] & -\lambda[6] & \lambda[6] & \lambda[2] \\ \lambda[3] & -\lambda[7] & -\lambda[1] & -\lambda[5] & \lambda[5] & \lambda[1] & \lambda[7] & -\lambda[3] \\ \lambda[4] & -\lambda[4] & -\lambda[4] & \lambda[4] & \lambda[4] & -\lambda[4] & -\lambda[4] & \lambda[4] \\ \lambda[5] & -\lambda[1] & \lambda[7] & \lambda[3] & -\lambda[3] & -\lambda[7] & \lambda[1] & -\lambda[5] \\ \lambda[6] & -\lambda[2] & \lambda[2] & -\lambda[6] & -\lambda[6] & \lambda[2] & -\lambda[2] & \lambda[6] \\ \lambda[7] & -\lambda[5] & \lambda[3] & -\lambda[1] & \lambda[1] & -\lambda[3] & \lambda[5] & -\lambda[7] \end{bmatrix}$$

Deutliche Symmetrie ist erkennbar, daher Faktorisierung mit einer entsprechenden Matrix

DCT nach Chen

$$\begin{bmatrix} \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] \\ \lambda[1] & \lambda[3] & \lambda[5] & \lambda[7] & -\lambda[7] & -\lambda[5] & -\lambda[3] & -\lambda[1] \\ \lambda[2] & \lambda[6] & -\lambda[6] & -\lambda[2] & -\lambda[2] & -\lambda[6] & \lambda[6] & \lambda[2] \\ \lambda[3] & -\lambda[7] & -\lambda[1] & -\lambda[5] & \lambda[5] & \lambda[1] & \lambda[7] & -\lambda[3] \\ \lambda[4] & -\lambda[4] & -\lambda[4] & \lambda[4] & \lambda[4] & -\lambda[4] & -\lambda[4] & \lambda[4] \\ \lambda[5] & -\lambda[1] & \lambda[7] & \lambda[3] & -\lambda[3] & -\lambda[7] & \lambda[1] & -\lambda[5] \\ \lambda[6] & -\lambda[2] & \lambda[2] & -\lambda[6] & -\lambda[6] & \lambda[2] & -\lambda[2] & \lambda[6] \\ \lambda[7] & -\lambda[5] & \lambda[3] & -\lambda[1] & \lambda[1] & -\lambda[3] & \lambda[5] & -\lambda[7] \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] \\ \lambda[4] & -\lambda[4] & -\lambda[4] & \lambda[4] & \lambda[4] & -\lambda[4] & -\lambda[4] & \lambda[4] \\ \lambda[2] & \lambda[6] & -\lambda[6] & -\lambda[2] & -\lambda[2] & -\lambda[6] & \lambda[6] & \lambda[2] \\ \lambda[6] & -\lambda[2] & \lambda[2] & -\lambda[6] & -\lambda[6] & \lambda[2] & -\lambda[2] & \lambda[6] \\ \lambda[1] & \lambda[3] & \lambda[5] & \lambda[7] & -\lambda[7] & -\lambda[5] & -\lambda[3] & -\lambda[1] \\ \lambda[5] & -\lambda[1] & \lambda[7] & \lambda[3] & -\lambda[3] & -\lambda[7] & \lambda[1] & -\lambda[5] \\ \lambda[3] & -\lambda[7] & -\lambda[1] & -\lambda[5] & \lambda[5] & \lambda[1] & \lambda[7] & -\lambda[3] \\ \lambda[7] & -\lambda[5] & \lambda[3] & -\lambda[1] & \lambda[1] & -\lambda[3] & \lambda[5] & -\lambda[7] \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & 0 & 0 & 0 & 0 \\ \lambda[4] & -\lambda[4] & -\lambda[4] & \lambda[4] & 0 & 0 & 0 & 0 \\ \lambda[2] & \lambda[6] & -\lambda[6] & -\lambda[2] & 0 & 0 & 0 & 0 \\ \lambda[6] & -\lambda[2] & \lambda[2] & -\lambda[6] & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda[7] & \lambda[5] & \lambda[3] & \lambda[1] \\ 0 & 0 & 0 & 0 & \lambda[3] & \lambda[7] & -\lambda[1] & \lambda[5] \\ 0 & 0 & 0 & 0 & -\lambda[5] & -\lambda[1] & -\lambda[7] & \lambda[3] \\ 0 & 0 & 0 & 0 & -\lambda[1] & \lambda[3] & -\lambda[5] & \lambda[7] \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

4x4 DCT
bitreversed
→ Rekursion

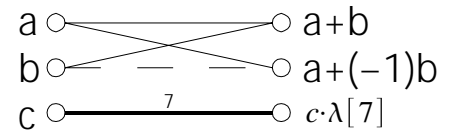
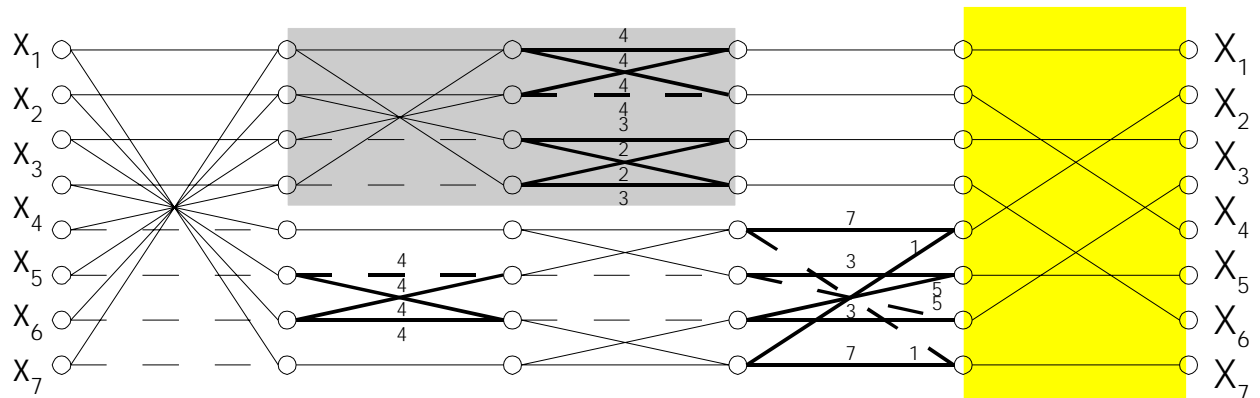
DCT nach Chen

$$\begin{bmatrix} \lambda[7] & \lambda[5] & \lambda[3] & \lambda[1] \\ \lambda[3] & \lambda[7] & -\lambda[1] & \lambda[5] \\ -\lambda[5] & -\lambda[1] & -\lambda[7] & \lambda[3] \\ -\lambda[1] & \lambda[3] & -\lambda[5] & \lambda[7] \end{bmatrix} = \begin{bmatrix} \lambda[7] & \lambda[7] & \lambda[1] & \lambda[1] \\ \lambda[3] & -\lambda[3] & -\lambda[5] & \lambda[5] \\ -\lambda[5] & \lambda[5] & -\lambda[3] & \lambda[3] \\ -\lambda[1] & -\lambda[1] & \lambda[7] & \lambda[7] \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\lambda[4] & \lambda[4] & 0 \\ 0 & \lambda[4] & \lambda[4] & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \lambda[7] & 0 & 0 & \lambda[1] \\ 0 & \lambda[3] & \lambda[5] & 0 \\ 0 & -\lambda[5] & \lambda[3] & 0 \\ -\lambda[1] & 0 & 0 & \lambda[7] \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\lambda[4] & \lambda[4] & 0 \\ 0 & \lambda[4] & \lambda[4] & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

DCT nach Chen

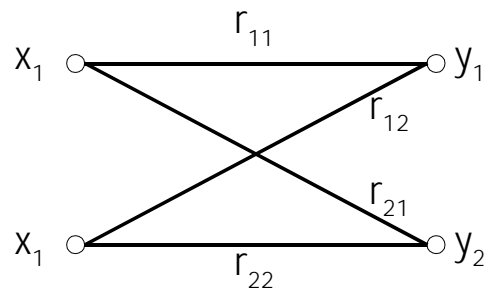
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda[4] & \lambda[4] & \lambda[4] & \lambda[4] & 0 & 0 & 0 & 0 \\ \lambda[4] & -\lambda[4] & -\lambda[4] & \lambda[4] & 0 & 0 & 0 & 0 \\ \lambda[2] & \lambda[6] & -\lambda[6] & -\lambda[2] & 0 & 0 & 0 & 0 \\ \lambda[6] & -\lambda[2] & \lambda[2] & -\lambda[6] & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda[7] & \lambda[5] & \lambda[3] & \lambda[1] \\ 0 & 0 & 0 & 0 & \lambda[3] & \lambda[7] & -\lambda[1] & \lambda[5] \\ 0 & 0 & 0 & 0 & -\lambda[5] & -\lambda[1] & -\lambda[7] & \lambda[3] \\ 0 & 0 & 0 & 0 & -\lambda[1] & \lambda[3] & -\lambda[5] & \lambda[7] \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} \lambda[7] & \lambda[5] & \lambda[3] & \lambda[1] \\ \lambda[3] & \lambda[7] & -\lambda[1] & \lambda[5] \\ -\lambda[5] & -\lambda[1] & -\lambda[7] & \lambda[3] \\ -\lambda[1] & \lambda[3] & -\lambda[5] & \lambda[7] \end{bmatrix} = \begin{bmatrix} \lambda[7] & 0 & 0 & \lambda[1] \\ 0 & \lambda[3] & \lambda[5] & 0 \\ 0 & -\lambda[5] & \lambda[3] & 0 \\ -\lambda[1] & 0 & 0 & \lambda[7] \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\lambda[4] & \lambda[4] & 0 \\ 0 & \lambda[4] & \lambda[4] & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



26 Additionen/Subtraktionen und 20 Multiplikationen

Umwandlung der Rotationsblöcke



Givens Rotation

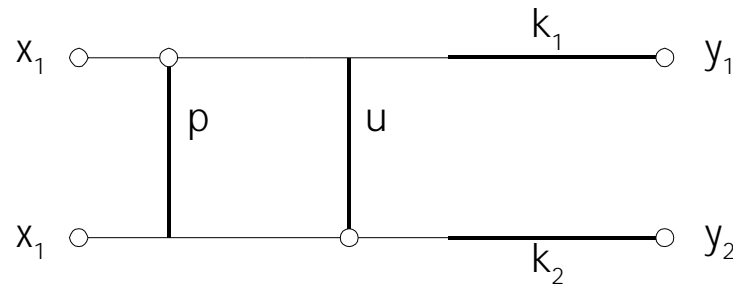
mit:

$$p = \frac{r_{12}}{r_{11}}$$

$$u = \frac{r_{11} r_{21}}{r_{11} r_{22} - r_{21} r_{12}}$$

$$k_1 = r_{11}$$

$$k_2 = \frac{r_{11} r_{22} - r_{21} r_{12}}{r_{11}}$$



Lifting Step

Vorteil: Neue Koeffizienten, die besser approximiert werden können

Beispiel: für: $r_{11} = \cos \alpha$ $r_{12} = \sin \alpha$ $r_{21} = -\sin \alpha$ $r_{22} = \cos \alpha$

folgt: $p = \tan \alpha$ $u = -\sin \alpha \cdot \cos \alpha$ $k_1 = \cos \alpha$ $k_2 = \frac{1}{\cos \alpha}$

Multiplilerless DCT

Idee: Annäherung der Multiplikations-Koeffizienten durch
einfach realisierbare Brüche

Methode: Brüche als dyadische Brüche umformen: $\frac{z}{2^n}$

Ansatz: Multiplikation mit konstanter Größe läßt sich in Shifts und
Additionen aufteilen, z.B.

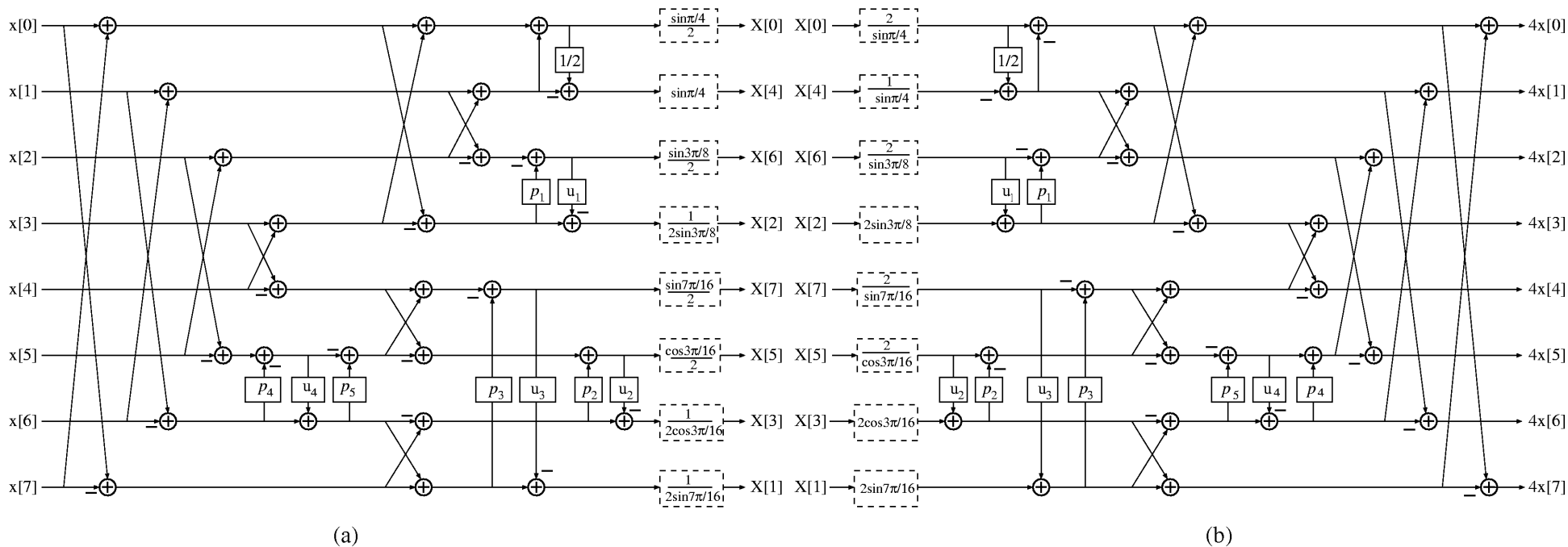
$$x \cdot 15 = x \cdot 111_b = (x \ll 3) + (x \ll 2) + (x \ll 1) + x$$

oder einfach:

$$x \cdot 15 = x \cdot (16 - 1) = (x \ll 4) - x$$

Weiterer Vorteil: exakte Rekonstruktion möglich, da Koeffizienten
stets Integer-Werte darstellen und IDCT rückwärts
durchlaufene DCT ist

Flow-Graph einer Multiplierless DCT



Config	p1	u1	p2	u2	p3	u3	p4	u4	p5	Shifts	Adds	MSE
1	1/2	1/2	1	1/2	1/4	1/4	1/2	3/4	1/2	9	28	2.30E-3
2	1/2	3/8	7/8	1/2	3/16	1/4	7/16	3/4	3/8	14	33	5.78E-4
...												
7	13/32	11/32	11/16	15/32	3/16	3/16	13/32	11/16	13/32	23	42	1.11E-5

2D-DCT

Allgemein:

$$y = A \cdot x \quad x = A^{-1} \cdot y$$

mit: $\dim(y) = N^2$, $\dim(x) = N^2$, $\dim(A) = N^2 \times N^2$

DCT ist allerdings orthonormal und separierbar, daher:

$$y' = A' \cdot x' \cdot A'^T$$

mit: $\dim(y') = N \times N$, $\dim(x') = N \times N$, $\dim(A') = N \times N$ und $A = A' \circ A'$

Fazit: Reduktion der Komplexität von $O(N^4)$ auf $O(2N^2)$
durch Zerlegung in Spalten- und Zeilentransformation

Weitere Vereinfachung möglich: statt $2N$ eindimensionaler DCTs
nur N eindimensionale DCTs

Signalgraph für 4x4 DCT aus 4 1D-DCTs

