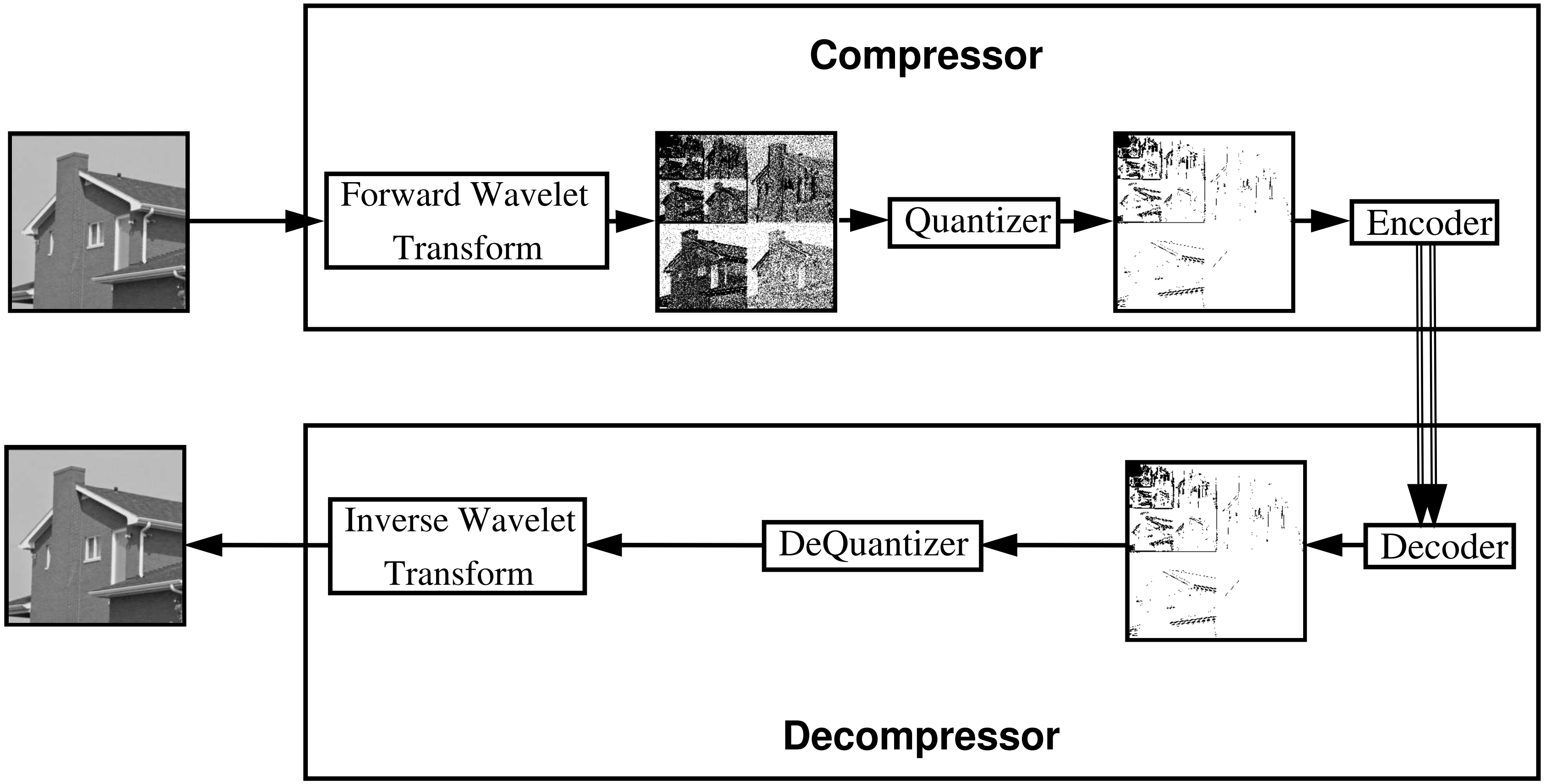
Waveletbasierte Bildkompression in der Anwendung



Übersicht

- Die Wavelet-Zerlegung
  - Hintergrund und Eigenschaften
  - Praktische Realisierung durch die FWT
  - Standardzerlegung / Pyramidalzerlegung (nach Mallat)
  - Einsatz zur Bildkompression
  - Wavelet Packet Verfahren
- Quantisierungsverfahren
  - Skalare Quantisierung
  - Vektorquantisierung
- Kodierung
  - Run-Length-Encoding (RLE)
  - Huffmannkodierung
  - Arithmetische Kodierung
- Leistungsvergleich mit anderen Bildkompressionsverfahren
- Weitere Wavelet-Anwendungen

Die stetige Wavelettransformation

- Ein **Wavelet** ist eine Funktion $\psi(t)$ im $L^2(\mathcal{R})$ mit der Eigenschaft

$$C_\psi = \int_{\mathcal{R}} \frac{|\hat{\psi}(\xi)|^2}{|\xi|} d\xi < \infty$$

- Wavelets $\psi_{\tau,s}(t)$ werden durch Translation und Skalierung aus einem **Mutter-Wavelet** $\psi(t)$ abgeleitet:

$$\psi_{\tau,s}(t) := \frac{1}{\sqrt{|s|}} \psi\left(\frac{t-\tau}{s}\right)$$

- Die Funktionen $\psi_{\tau,s}(t)$ bilden eine Funktionenfamilie $\{\psi_{\tau,s}\}$

- Interessant sind vor allem orthogonale Wavelets $\rightarrow$ die $\{\psi_{\tau,s}\}$ bilden dann eine **Orthonormalbasis** des $L^2(\mathcal{R})$

- Die Wavelettransformation einer Funktion f mit dem Wavelet ψ ist definiert als die Integraltransformation

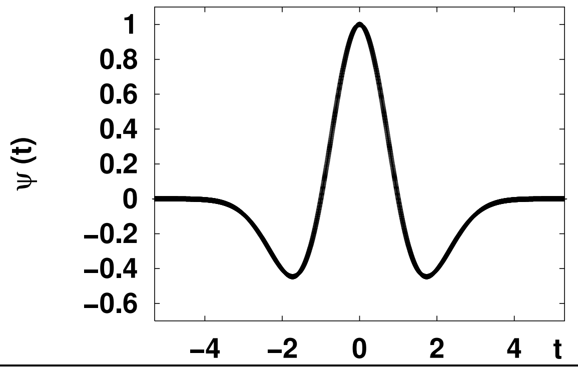
$$\mathcal{W}_{\tau,s}(f) = \langle f, \psi_{\tau,s} \rangle = \int_{\mathcal{R}} f(t)\, \psi_{\tau,s}(t)\, dt$$

- Die Originalfunktion kann wieder aus der Wavelettransformierten berechnet werden:

$$f(t) = \frac{1}{C_\psi} \int_{\mathcal{R}^2} \mathcal{W}_{\tau,s}(f)\, \psi_{\tau,s}(t)\, \frac{d\tau\, ds}{s^2}$$

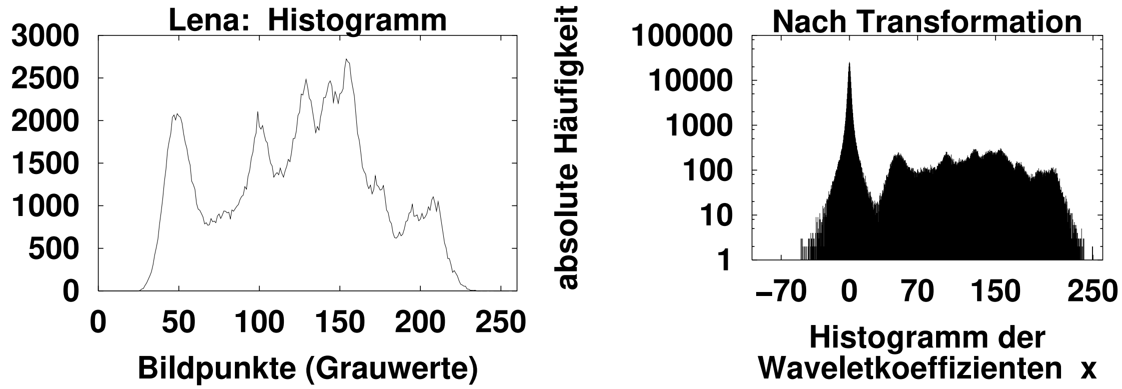
- Beispiel: *Mexikanischer Hut*

$$\psi(t) = (1-t^2)\, \mathrm{e}^{-t^2/2}$$



Eigenschaften der Wavelettransformation

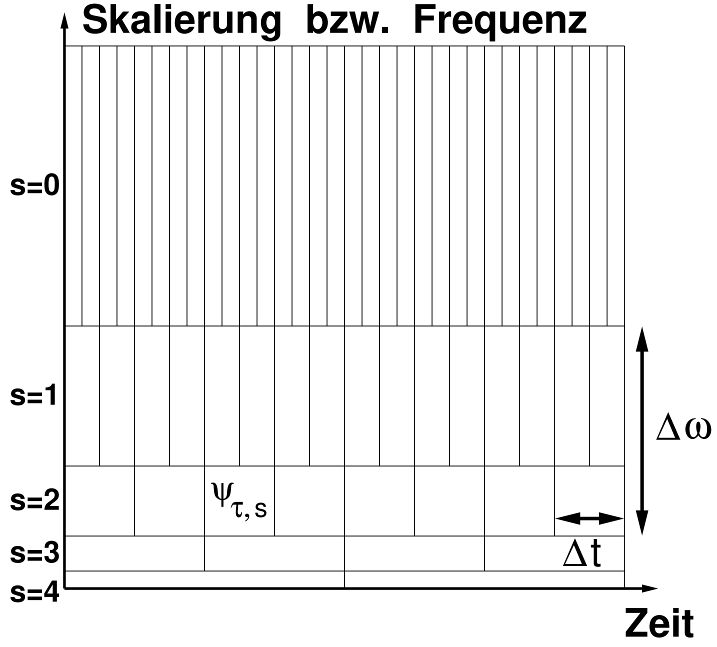
- Lineare Transformation, bei orthonormalen Wavelets/Filtern $\rightarrow$ orthonormale Transformation
- **Energieerhaltend**: $\|f\|^2 = \|\mathcal{W}(f)\|^2$
- **Energiekonzentrierend**:



$\rightarrow$ Signal kann mit wenigen Waveletkoeffizienten dargestellt werden

- Durch die Wavelettransformation wird das Eingangssignal dekorreliert

- Besseres Zeit-Frequenzverhalten als bei der FFT, es gilt: $\Delta\omega \cdot \Delta t \geq 1/2$



- Wavelets mit kompaktem Träger sind besser, da kleinere Zeitauflösung erreicht wird

Weitere Vorteile:

- Keine Block-Effekte wie bei `JPEG`

Die Fast Wavelet Transformation (FWT)

- Praktische Realisierung der WT durch Filtern der Eingangsdaten mit je einem **Hochpaßfilter** g und **Tiefpaßfilter** h, anschließend **Downsampling** um 2
- Operatoren G und H führen die Zerlegung durch, Operatoren G^* und H^* die Rekonstruktion (dann mit Upsampling)
- Dabei Entstehung von tieffrequenten Approximationssignalen a_i und hochfrequenten Detailsignalen d_i
- Benutzung von endlichen, orthonormalen FIR-Filtern, Länge $L \sim 4 \ldots 20$
  Beispiel: *Daub2*-Filter

$$g:(\quad 0.48296,\quad 0.83652,\quad 0.22414,-0.12941)$$
$$h:(-0.12941,-0.22414,\quad 0.83652,-0.48296)$$

$$\rightarrow g_k = (-1)^k h_{L-1-k}, \quad k = 0,\ldots L-1$$

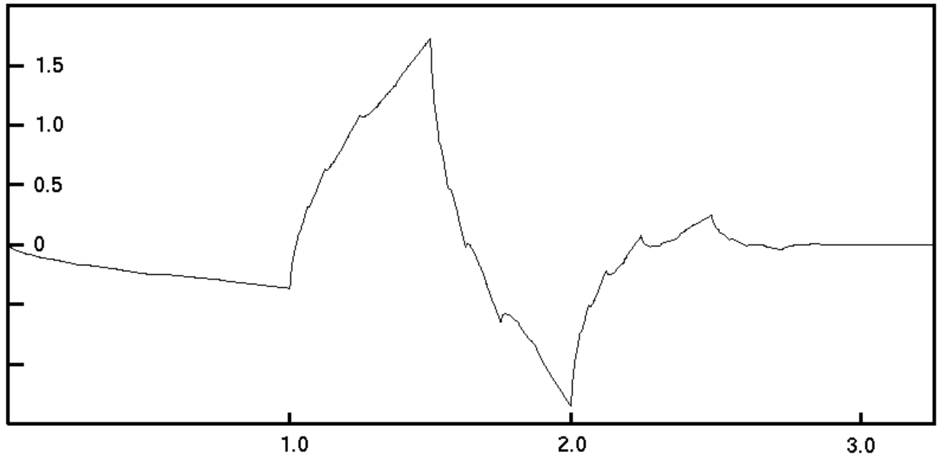
- Notwendige **Filterbedingungen**:

$$\sum_k h = \sqrt{2}, \quad \sum_k g = 0, \quad \sum_k h_k h_{k+2m} = \delta_{0,m}$$

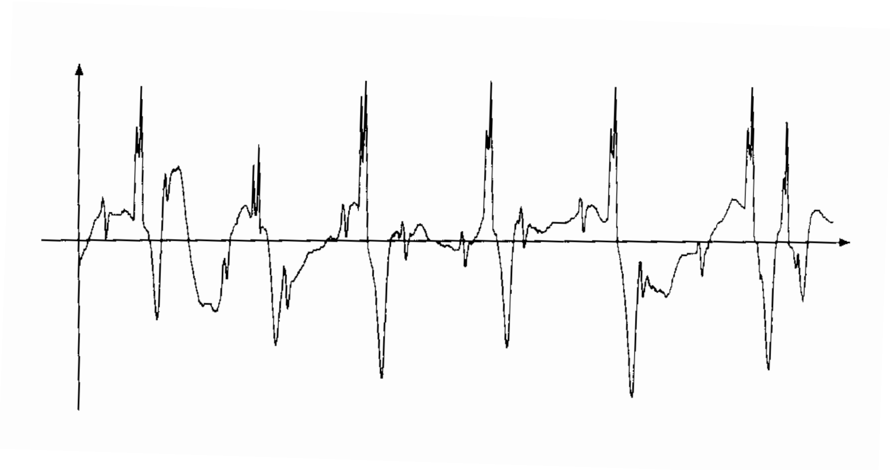
  $\rightarrow$ sichert die Umkehrbarkeit und Orthogonalität der Transformation

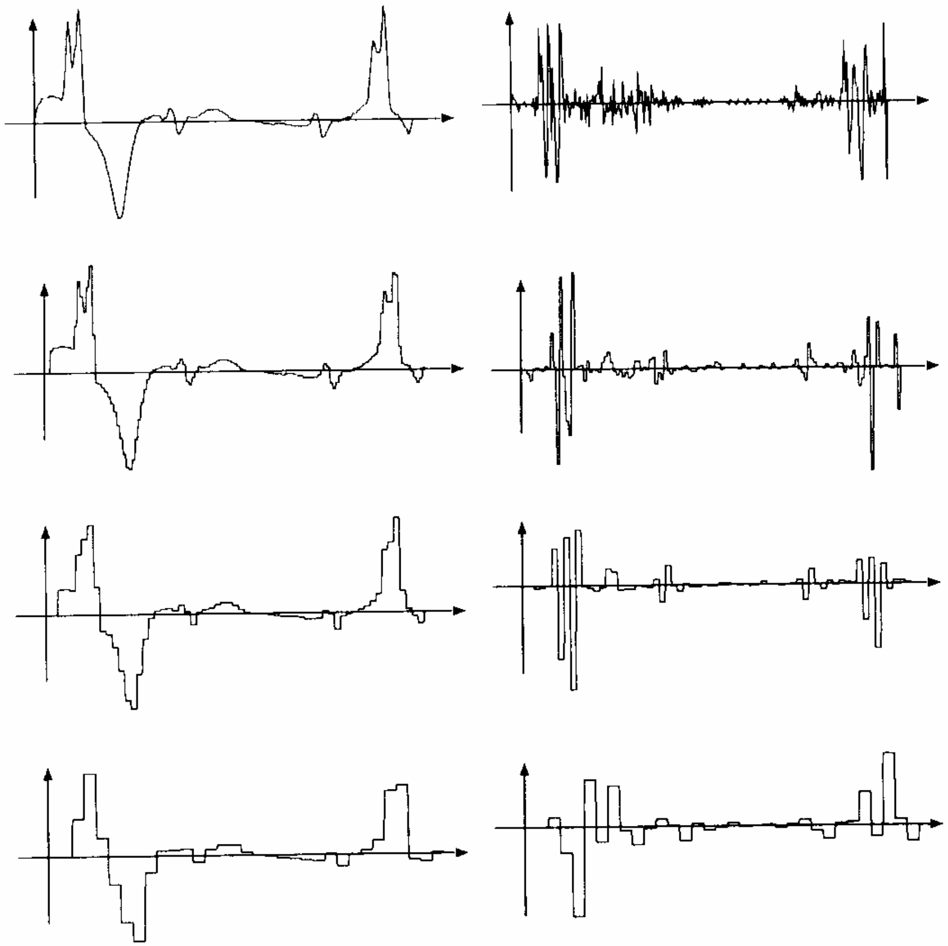
- Die iterative Anwendung von g auf $(\ldots,0,0,1,0,0,\ldots)$ (ohne Downsampling) erzeugt die zugehörige (fraktale) Waveletfunktion:

Daub2 Waveletfunktion



Zerlegung eines EKG-Signals





EKG-Signal eines Menschen

Wavelet-Zerlegung des EKG-Signals (links a_i, rechts d_i)

Durchführung der FWT

Zerlegung:

$$\begin{tikzcd}
a_0 \arrow[r, "H"] \arrow[dr, "G"'] & a_1 \arrow[r, "H"] \arrow[dr, "G"'] & a_2 \arrow[r, "H"] \arrow[dr, "G"'] & \cdots & a_{N-1} \arrow[r, "H"] \arrow[dr, "G"'] & a_N \\
& d_1 & d_2 & \cdots\cdots\cdots & & d_N
\end{tikzcd}$$

G: Hochpaß $\quad a_i = Ha_{i-1} \quad$ N: Zerlegungs-
H: Tiefpaß $\quad d_i = Ga_{i-1} \quad$ tiefe

$$\rightarrow \mathbf{W} = \text{FWT}(a_0) = (d_1, d_2, \ldots, d_N, a_N)$$

Rekonstruktion:

$$\begin{tikzcd}
a_N \arrow[r, "H^*"] & a_{N-1} \arrow[r, "H^*"] & a_{N-2} \arrow[r, "H^*"] & \cdots & a_1 \arrow[r, "H^*"] & a_0 \\
d_N \arrow[ur, "G^*"] & d_{N-1} \arrow[ur, "G^*"] & d_{N-2} \arrow[ur, "G^*"] & \cdots\cdots & d_1 \arrow[ur, "G^*"] &
\end{tikzcd}$$

G^*: adj. Hochpaß $\quad a_{i-1} = H^* a_i + G^* d_i$
H^*: adj. Tiefpaß $\quad d_i \in \mathbf{W}$

- Zerlegungsbeispiel mit **Haarfilter** $h = (1/\sqrt{2}, 1/\sqrt{2})$, $g = (1/\sqrt{2}, -1/\sqrt{2})$:

$$\begin{tikzcd}
& & a_0 \; \boxed{1, \quad -2, \quad 3, \quad -4} \arrow[dl, "H"'] \arrow[dr, "G"] & \\
& a_1 \; \boxed{-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}} \arrow[dl, "H"'] \arrow[d, "G"] & & \boxed{3\sqrt{2}, 7\sqrt{2}} \; d_1 \\
a_2 \; \boxed{-1} & \boxed{0} \; d_2 & & N = 2
\end{tikzcd}$$

- **Randploblem**: *periodische* oder *symmetrische* Erweiterung

- **Komplexität** der FWT: $O(N)$
  $\rightarrow$ schneller als die FFT mit $O(N \log N)$

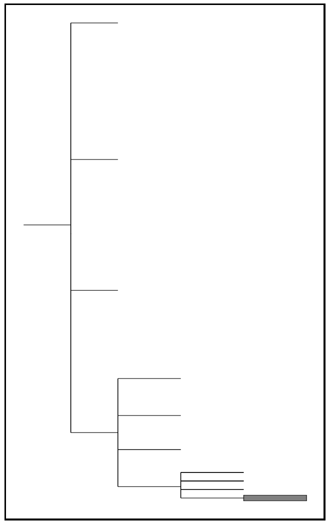
Bildkompression mit Wavelets

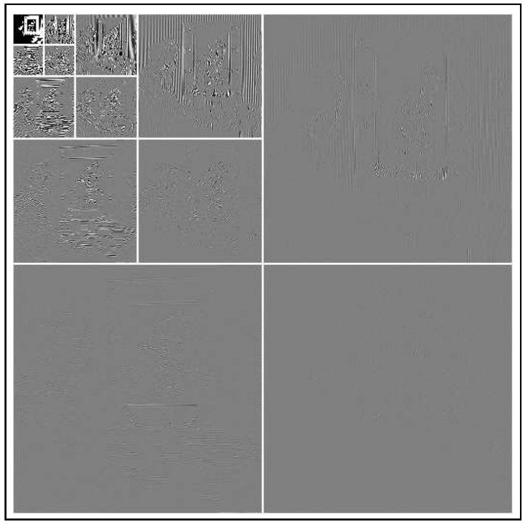
- Für *Bildkompression* Verallgemeinerung der WT auf den zweidimensionalen Fall
  $\rightarrow$ Hintereinanderausführen von 2 eindimensionalen WT
- Zuerst Filterung der Bildzeilen, anschließend Filterung der Bildspalten der Zwischenbilder
- **rekursive Weiterzerlegung** aller tieffrequenten Teilbilder mit der `FWT`
- Aus entstehendem Baum läßt sich das Originalbild verlustfrei rekonstruieren
- **Abschneiden** betragsmäßig kleiner Waveletkoeffizienten x:

  $$\texttt{if } (\|x\| \leq \text{cut}) \texttt{ then } x = 0$$

  $\rightarrow$ Verlust $\rightarrow$ ermöglicht Kompression

Standard Wavelet Zerlegung

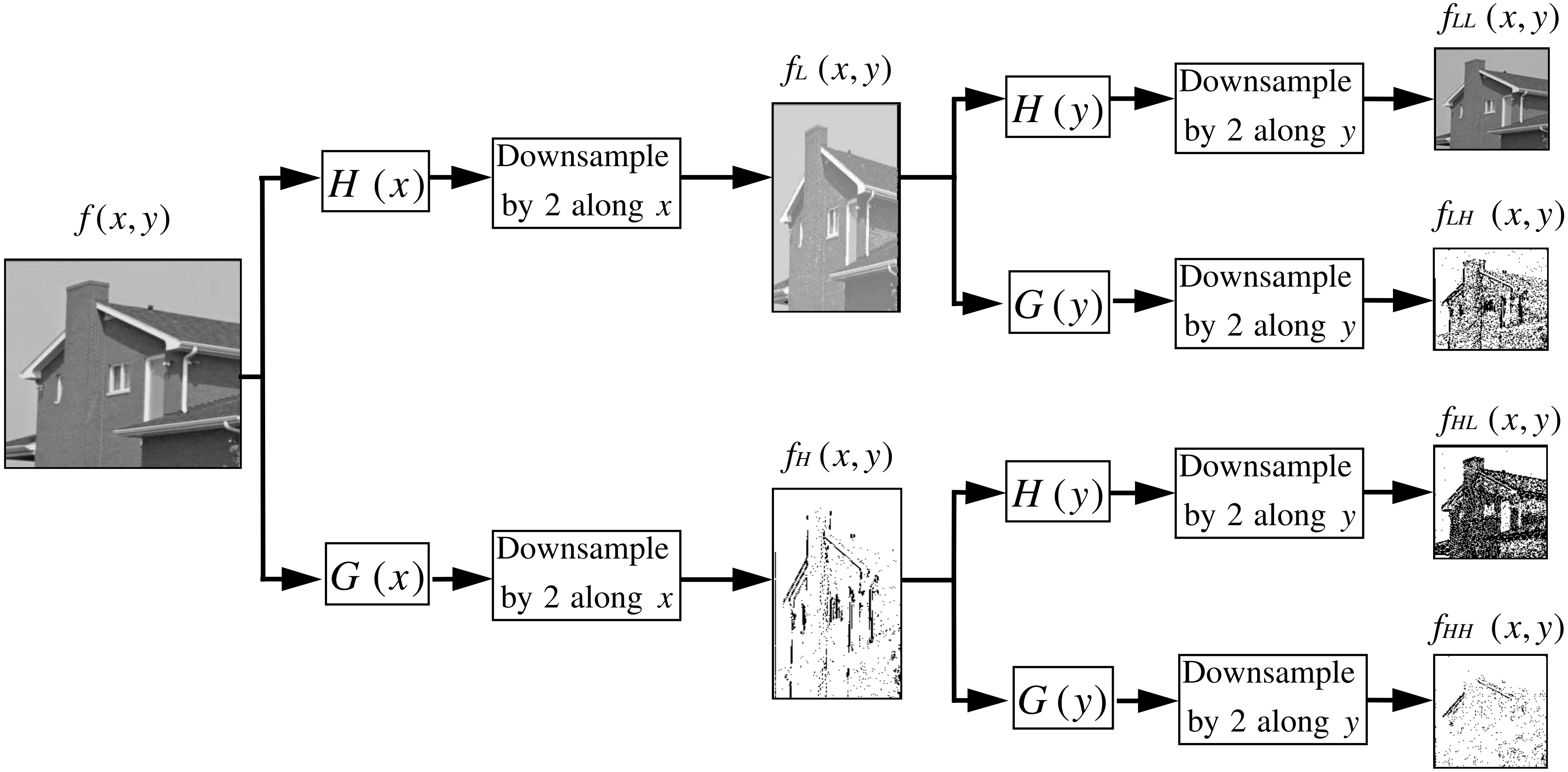




Links Original-lena, rechts nach cut=70

`PSNR`=24.11

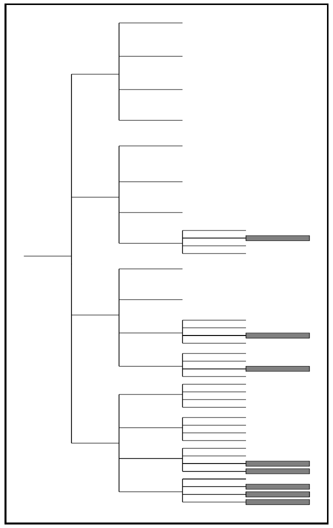
Durchführung der zweidimensionalen FWT

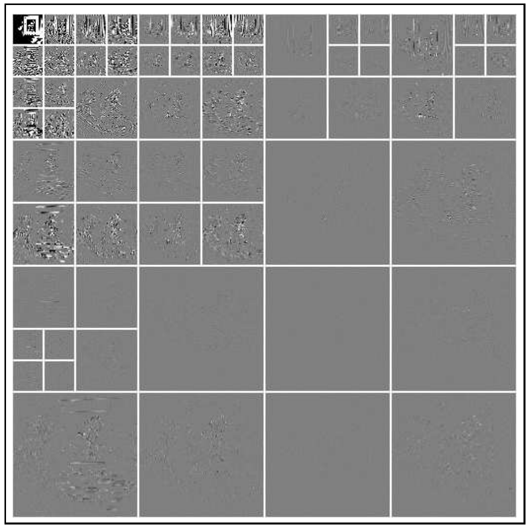


- Anschließend **Quantisierung** der (reellen) Waveletkoeffizienten$\rightarrow$ weiterer Verlust, aber Output ist sehr gut komprimierbar
- **Kodieren** / **Packen** der quantisierten Daten, mit RLE und z. B. Huffman- oder arithmetischer Kodierung
- Ergebnis abspeichern

Wavelet Packet Verfahren

Wavelet Packet Zerlegung: B_{best}





- Verallgemeinerung: Zunächst erfolgt die Erzeugung *aller* Teilbilder durch die FWT
  $\rightarrow$ voller **Quadtree** Q der Tiefe L

- Jeder Vaterknoten kann durch die *Inverse* FWT aus seinen 4 Söhnen berechnet werden $\rightarrow$ wahlweises Abspeichern des Vaters oder der Söhne

- Durch Entfernen verschiedener Teilbäume aus Q entsteht eine Vielzahl von verschiedenen “Basen” B_i für das Ausgangsbild

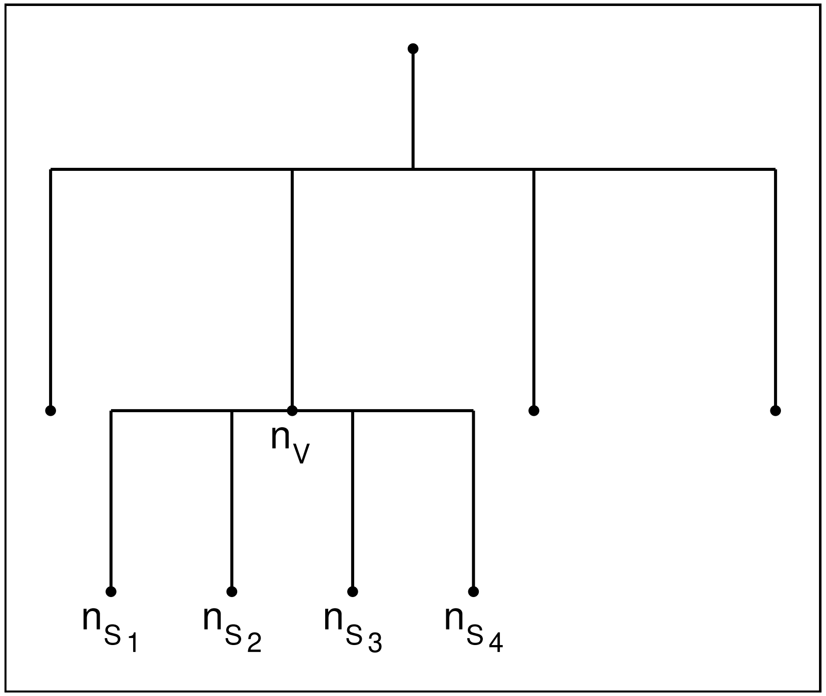
- Jede Basis B_i bestimmt eine spezifische Kompressionsrate K $\rightarrow$ Suche nach ***Bester Basis*** mit maximalem K
  (die PSNR bleibt gleich!)

- Die Anzahl A_L aller B_i im Level L ist:
  $A_L \geq 2^{4^{L-1}}$ $\quad (A_4 \geq 1.84 \times 10^{19})$

- $\rightarrow$ Einschränkung notwendig:
  Nur noch Untersuchung der *parent-child* Verhältnisse

bottom-up Algorithmus zur Bestimmung der Besten Basis $\mathbf{B_{best}}$

Kosten von Vater $\mathtt{n_V}$ und Söhnen $\mathtt{n_{S_i}}$



$$S = \sum_{i=1}^{4} C(\mathtt{n_{S_i}})$$

$$C(\mathtt{n_V}) \leq S \quad ?$$

- *Schritt 1:* Bewertung aller Knoten n in Q mit den **Kosten** $C(n)$, z. B.

$$C(n) = -\sum_j |n_j|^2 \log |n_j|^2$$

  (an Entropie angelehnt)

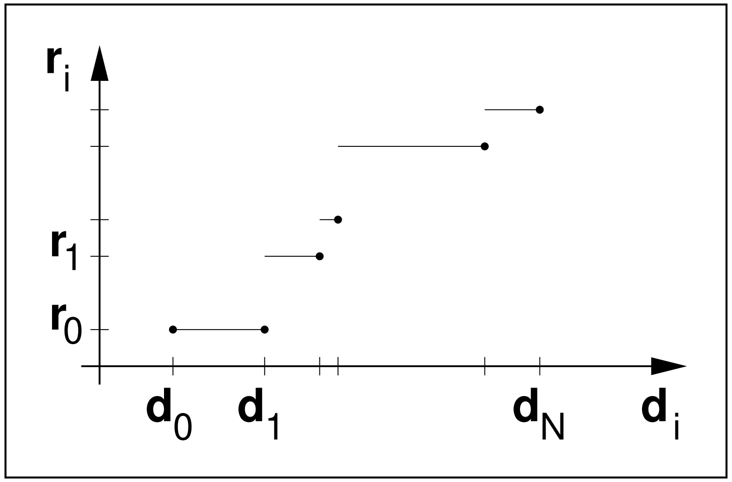
- *Schritt 2:* Für alle Knoten im untersten Level L: Vergleich der Summe S der Kosten von je 4 Söhnen mit den Kosten $C(\mathtt{n_V})$ des Vaterknotens.
  - Falls $C(\mathtt{n_V}) \leq S$: Entfernung der Söhne aus Q.
  - Falls $C(\mathtt{n_V}) > S$: setze $C(\mathtt{n_V}) = S$

- *Schritt 3:* Schritt 2 für die nächst höheren Level wiederholen, bis man bei der Wurzel angelangt ist

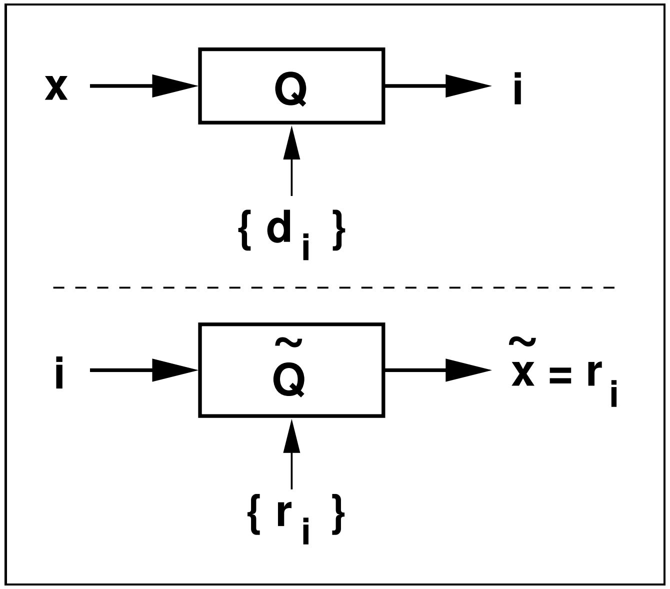
- Der verbliebene Baum ist dann B_{best}

Skalare Quantisierung

- Quantisierung: ermöglicht **verlustbehaftete Kompression** von Werten
- Jeder numerische Wert $x \in [\min, \max]$ wird einzeln quantisiert
- Quantisierer ist Treppenfunktion Q:
  $Q(x) := r_i$ falls $x \in (d_i, d_{i+1}]$, $\quad i = 0 \ldots N$



- $\mathbf{d_i}$: *decision levels* **|** $\mathbf{r_i}$: *reconstruction levels*
- Quantisierung: $x \to i$
  Dequantisierung: $i \to r_i$

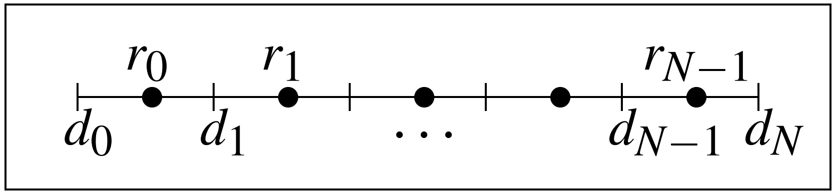


Probleme:

- Vergrößerung von N
  $\rightarrow$ bessere Bildqualität, aber schlechtere Kompressionsrate
- Berechnung von optimalen d_i und r_i bei vorgegebenem N
  $\rightarrow$ *gleichzeitige* Optimierung von Kompressionsrate und Bildqualität

Uniforme Quantisierung

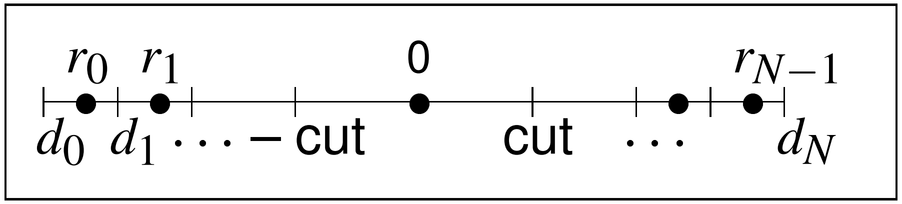
- Einfachster skalarer Quantisierer: **Äquidistante Einteilung** des Intervalls $[min, max] = [d_0, d_N]$ in die N **Quantisierungsintervalle** $(d_i, d_{i+1}]$ mit:
$$d_i = min + i \cdot \frac{max - min}{N}, \quad r_i = \frac{d_{i+1} - d_i}{2}$$



- Vorteil: Nur $\{\text{min}, \text{max}, N\}$ muß gespeichert werden
- Nachteil: Keine Anpassung an Verteilung der x-Werte
$\rightarrow$ schlechtere Bildqualität

Uniforme Quantisierung mit Deadzone

- Einführung einer **Deadzone** $[-\text{cut}, \text{cut}]$ mit $Q(x) = 0$ falls $x \in [-\text{cut}, \text{cut}]$



- Da Wavelettransformation Energiekonzentrierend $\rightarrow$ gut geeignet zur Quantisierung von Waveletkoeffizienten
- Vorteil: Nur $\{\text{min}, \text{max}, \text{cut}, N\}$ muß gespeichert werden,
Größeres cut $\rightarrow$ Verbesserung des Verhältnisses von Bildqualität und Kompressionsrate

---

Getrennte Quantisierung des letzten tieffrequenten Teilbildes notwendig:
$\rightarrow$ Uniforme Quantisierung mit z. B. $N = 256$

Lloyd-Max Quantisierer

- Bekannte Häufigkeitsverteilung $p(x)$ der numerischen Werte x

- Ziel: **optimale** Bestimmung der d_i und r_i bei festem N zur Minimierung des Rekonstruktionsfehlers

$$E(r_i) = \sum_{i=0}^{N-1} \sum_{d_i < x \leq d_{i+1}} p(x)\,(x - r_i)^2$$

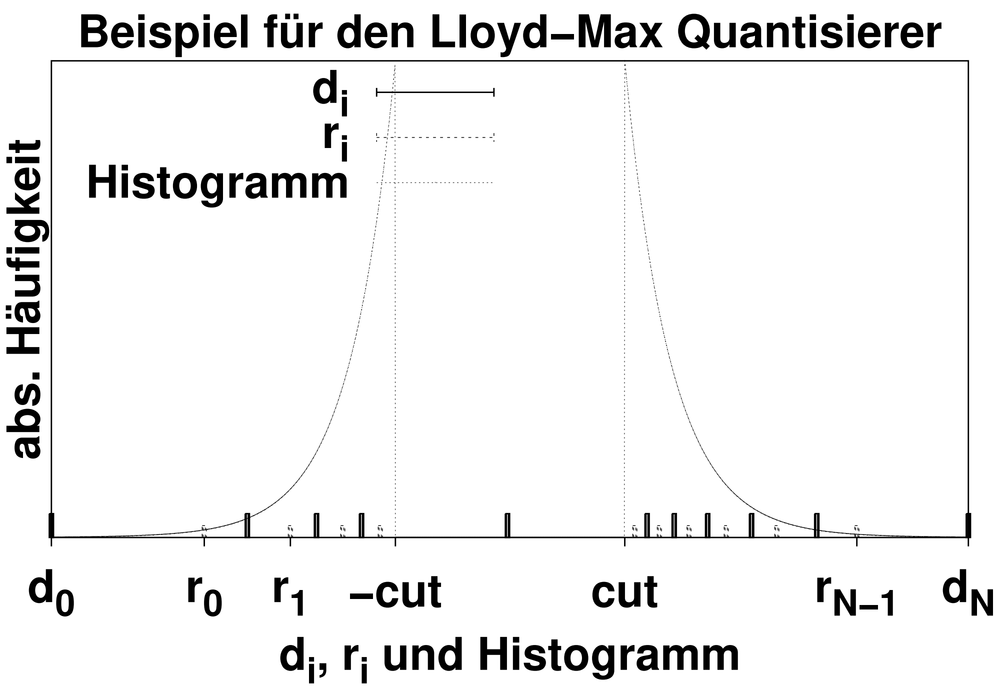
- $\rightarrow$ 2 Bedingungen, die gleichzeitig erfüllt sein müssen:

$$r_i = \frac{\sum\limits_{d_i < x \leq d_{i+1}} p(x)\,x}{\sum\limits_{d_i < x \leq d_{i+1}} p(x)} \quad \text{und} \quad d_i = \frac{r_{i-1} + r_i}{2}$$

- Achtung: Keine Berücksichtigung der Entropie!

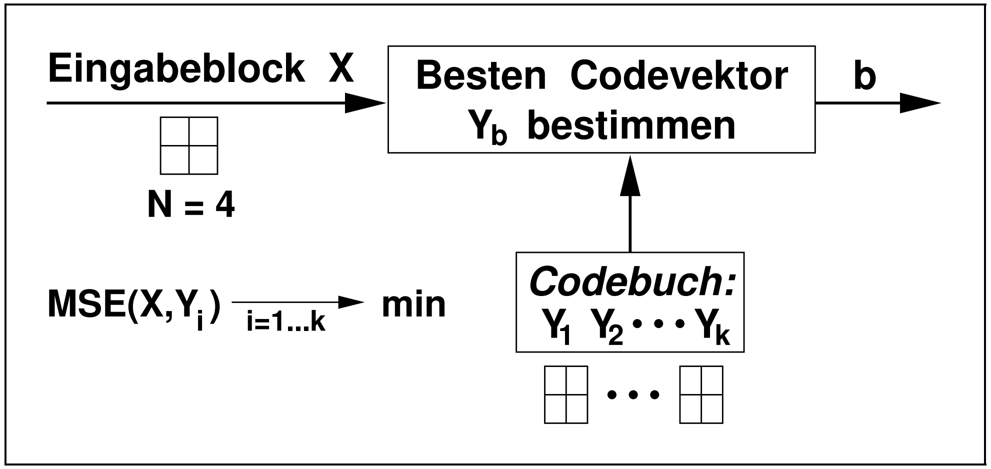
- Dazu iteratives Verfahren:

1. Start mit beliebigen d_i, z. B. äquidistant verteilt
2. Berechnung der r_i
3. Berechnung der d_i
4. Weiter bei 2., bis Konvergenzkriterium erfüllt ist (z. B. $E(r_i)$ ist minimal)

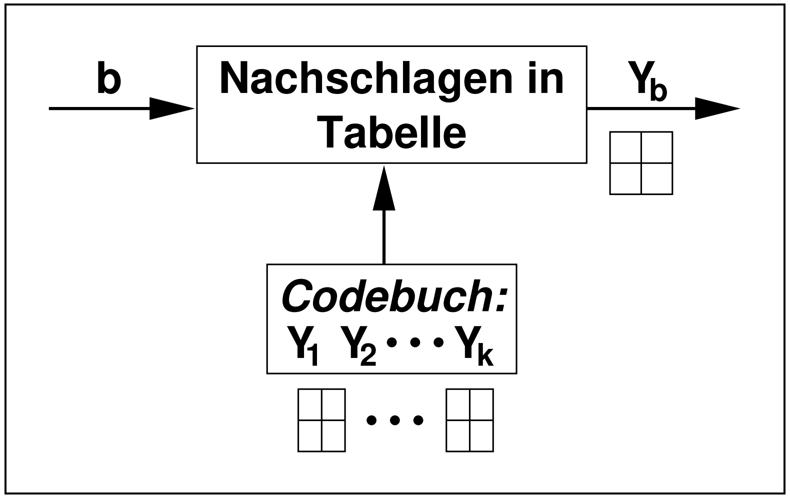


Vektor-Quantisierung

Quantisierer:



Dequantisierer:

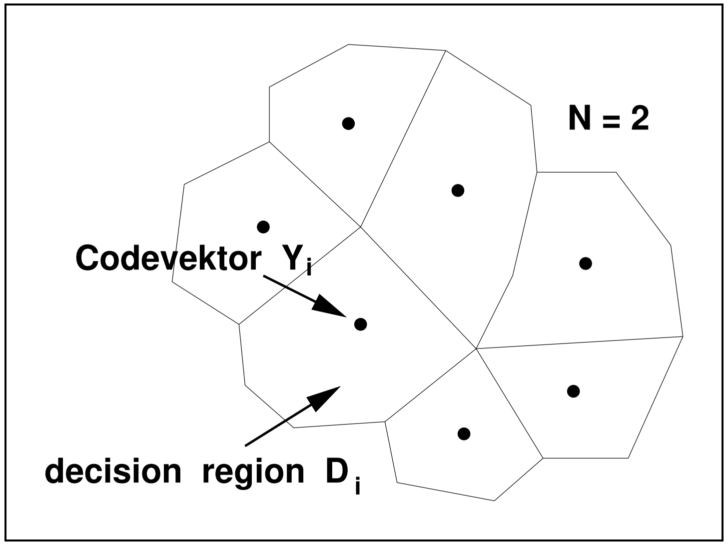


- Gemeinsame Quantisierung von mehreren Werten (Bildpunkte oder Transformationskoeffizienten)
- Disjunkte Aufteilung der Werte in Blöcke(=***Vektoren***) der Größe N
- ***Codebuch*** mit häufig auftretenden ***Codevektoren*** Y_i
- Für jeden Bildblock X Bestimmung des am besten passenden Codevektors Y_b durch Minimierung des MSE(X, Y_i)
- Quantisierung: $X \rightarrow b$
- Sehr schnelle Dequantisierung: $b \rightarrow \tilde{X}$
- Verlustbehaftete Quantisierung: $X \approx \tilde{X}$ (außer Codebuch ist *vollständig*)

Vektor-Quantisierung: Codebuch Design

- Vergrößerung des Codebuchs
  $\rightarrow$ Bessere Bildqualität
  $\rightarrow$ Schlechtere Kompressionsrate
- Große Codebücher: viel Speicher notwendig + langsame Suche
- Benötigte Bitrate bei k Codevektoren und N Werten/Pixeln je Block: $\frac{\log_2 k}{N}$ bpp

Bestimmung von Y_i und D_i:



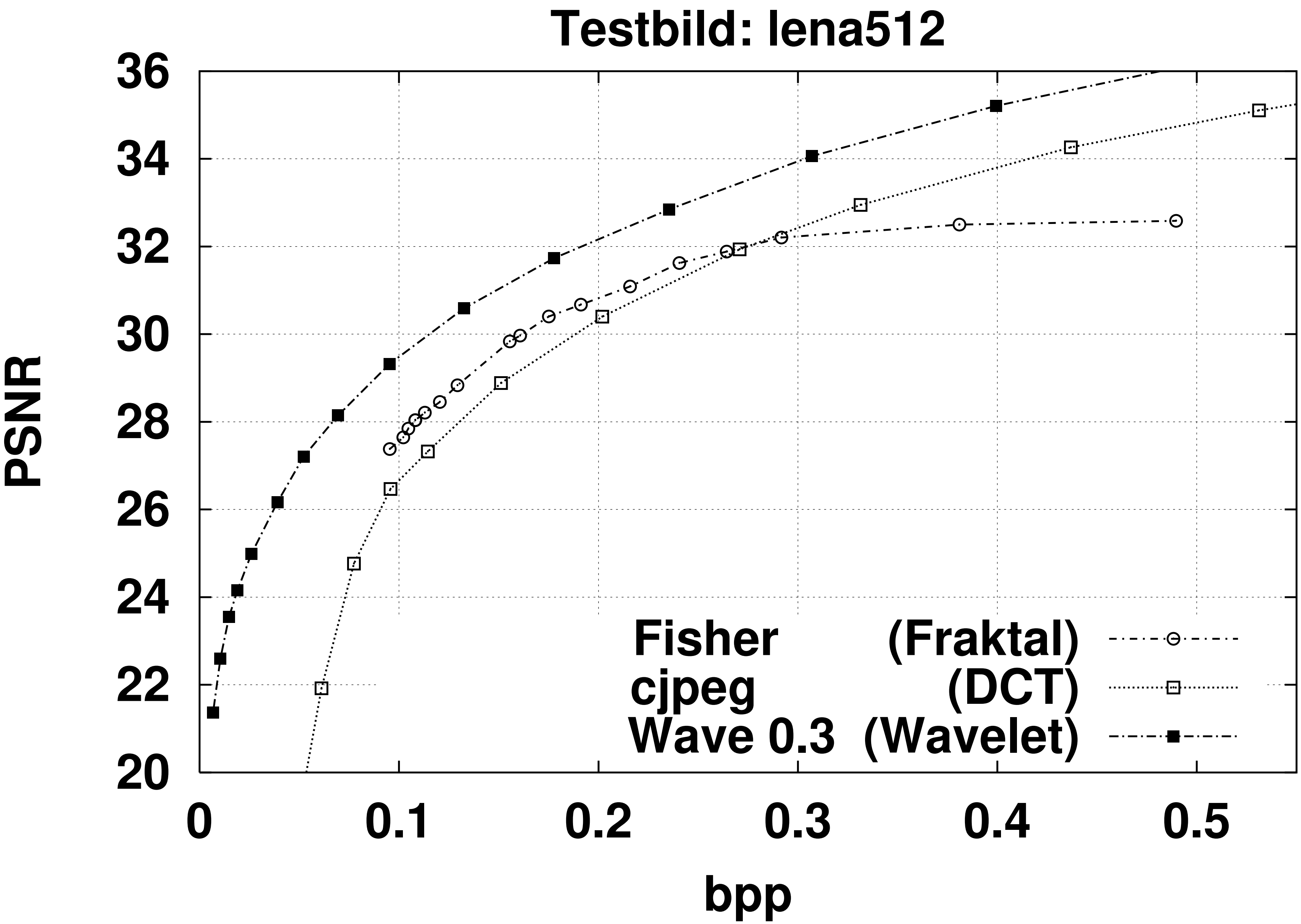
- Hauptproblem: ***Codebuch-Design***
- Klassischer LBG-Algorithmus zur Codebuch-Generierung: *Linde-Buzo-Gray* $\rightarrow$ Vektorversion des Lloyd-Max-Quantisierers:

I)	Bekannte $D_i \rightarrow Y_i =$ Schwerpunkt (D_i)
II)	Bekannte $Y_i \rightarrow D_i$ mit $Y_i \in$ nächstes D_i

- I) + II) müssen gleichzeitig erfüllt sein $\rightarrow$ Algorithmus:

1. Start-Codebuch mit k Vektoren $Y_1, \ldots, Y_k$
2. Mit Y_i Bestimmung der *decision regions* D_i
3. Neue Y_i als Schwerpunkte der D_i
4. Wiederholen von 2. und 3., bis sich der Rekonstruktionsfehler nicht weiter verringert (o. ä. Konvergenzkriterium)

Vergleich mit anderen Bildkompressionsverfahren



Optischer Vergleich der Verfahren

Verschiedene Bildqualitäten von **lena512** bei
einer Kompressionsrate von 1 : 84 bzw. 0.095 bpp

cjpeg

PSNR=26.47

Fisher

PSNR=27.38

Wave 0.3

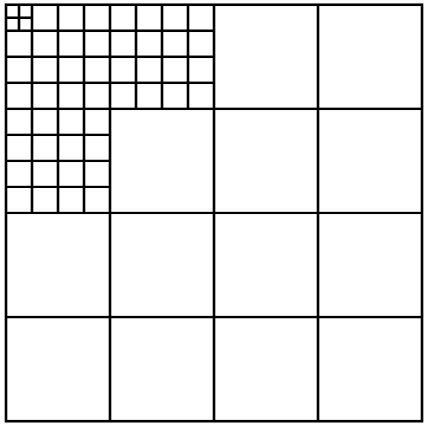
PSNR=29.32

Wavelet-Anwendungen

WSQ-Bildkompression

- Das FBI digitalisiert sein Archiv von $\sim \mathbf{200}$ **Millionen** Fingerabdrücken, die auf Karteikarten vorliegen
- WSQ-Verfahren (Wavelet/Scalar Quantization) benutzt eine spezielle Wavelet Packet Basis zum Komprimieren der Fingerabdrücke, Kompressionsrate $\sim 15:1$

Basis



Beispiel

45621 Byte, $K = 12.9$

Summus Netscape-Plugin *WI*

- Schneller als JPEG oder Fraktale sowie bessere Bildqualität
- Freies Netscape-Plugin oder ActiveX-Control für Windows und NT
- Kommerzielles Plugin für Photoshop
- Unterstützt 8 bit Graustufenbilder oder 24 bit Farbbilder
- Bildformat wird von CoralDRAW (ab 6.0) unterstützt
- Videokompression in Vorbereitung

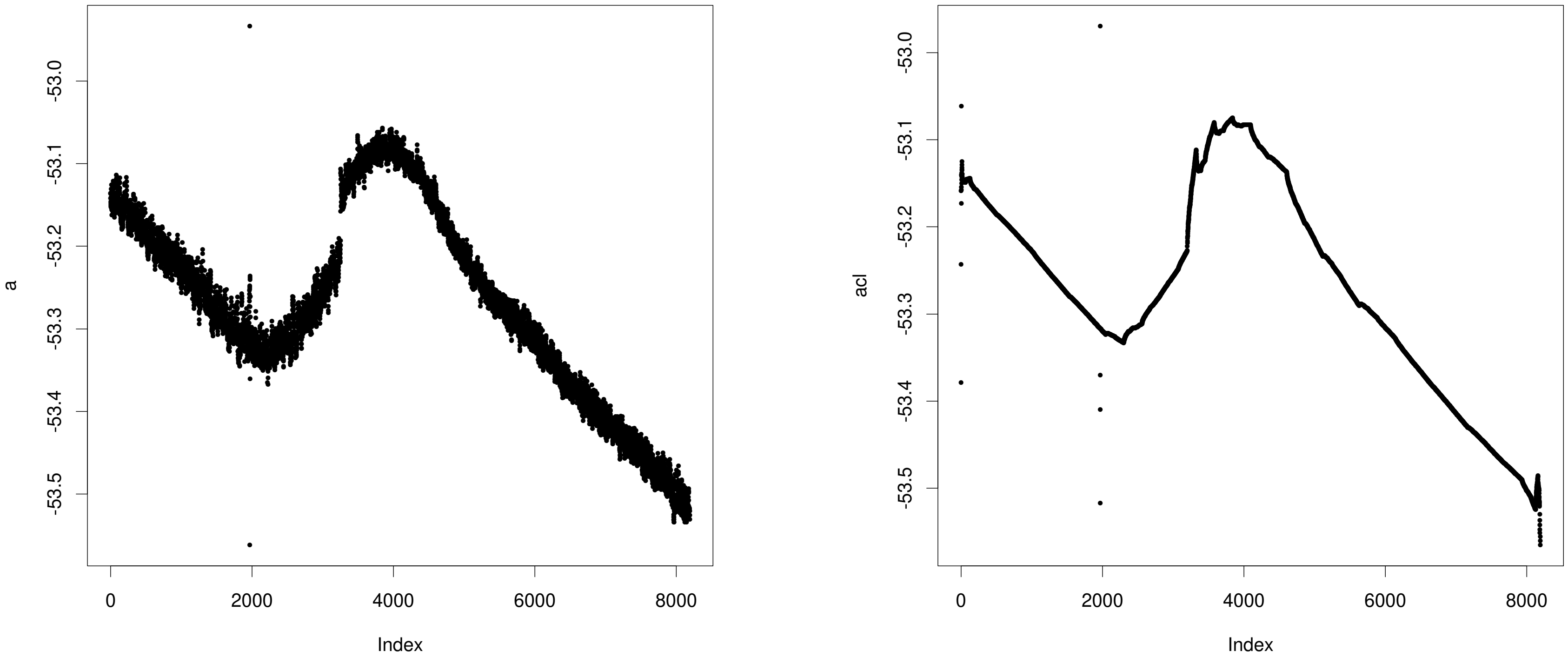
- Ähnliche Plugins werden auch von anderen Softwarefirmen angeboten: (*Lightning Strike, RICOH, SPIHT*)

Wavelet-Anwendungen: JPEG 2000

- Geplanter Nachfolger des JPEG-Standards, wird auf Wavelets basieren
- Gegenwärtig läuft der *call for contributions* und *call for proposals*
- Bei Fertigstellung $\rightarrow$ ISO-Norm
- Für den MPEG-Nachfolger ist ebenfalls der Einsatz von Wavelets geplant

Weitere Wavelet-Anwendungen

- Medizin: EEG-Klassifizierung
- Musik: Restauration sehr alter Musikstücke
- Geologie: Analyse seismischer Daten → **Beispiel:**



(a) Raw data, water level vs. time

(b) After thresholding the wavelet transformation.

Figure 10: Panel (a) shows $n = 8192$ hourly measurements of the waterlevel for a well in an earthquake zone. Notice the wide range of waterlevels at the time of an earthquake around $t = 2000$.

Literatur und Webseiten über Bildkompression und Wavelets

Literatur

[CGV95] P. C. Cosman, R. M. Gray, and M. Vetterli. Vector Quantization of Image Subbands: A Survey. Technical report, June 1995.

[CW92] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, 1992.

[Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, 1992. URL: `http://www.siam.org/catalog/mcc02/daubechi.htm`.

[HJS95] M. L. Hilton, B. D. Jawerth, and A. Sengupta. Compressing still and moving images with wavelets. *Multimedia Systems*, 3(2), 1995. URL: `ftp://ftp.math.scarolina.edu/pub/wavelets/papers/varia/tutorial`.

[Max60] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, (IT-6(1)):7–12, 1960.

[VM94] Brani Vidaković and Peter Müller. Wavelets for kids: A tutorial introduction. 1994. URL: `ftp://ftp.isds.duke.edu/pub/brani/papers/wav4kids[A|B].ps.Z`.

[Zel11] Meinrad Zeller. Flinkes Wellenspiel. *c't*, Seiten 258–264, 1994, Heft 11.

WWW-URL's

[Bri96] Chris Brislawn. The FBI Fingerprint Image Compression Standard, 1996. URL: `http://www.c3.lanl.gov/~brislawn/FBI/FBI.html`.

[Com97] JPEG Committee. Call for contributions for JPEG 2000: Image Coding System, March 1997. URL: `http://www.disc.org.uk/public/jpegnew.htm`.

[Gra95] Amara Graps. An introduction to Wavelets. *IEEE Computational Science & Engineering*, Summer, 1995. URL: `http://www.amara.com/IEEEwave/IEEEwavelet.html`.

[Pol] Robi Polikar. The Wavelet Tutorial. URL: `http://www.public.iastate.edu/~rpolikar/WAVELETS/WTtutorial.html`.

[Sch98] René Scholz. Wavelet Links, 1998. URL: `http://www2.informatik.uni-jena.de/~mrz/Wavelets.html`.

[Sum] Summus Ltd. Wavelet Technology & Information Access. URL: `http://www.summus.com/`.

[Use97] Usenet. Compression FAQ, 1997. URL: `http://www.cis.ohio-state.edu/hypertext/faq/usenet/compression-faq/top.html`.

[War] Dartmouth College Wavelet Warriors. Wavelet Warrior Research Interests. URL: `http://http://www.cs.dartmouth.edu/~gdavis/warriors.html`.