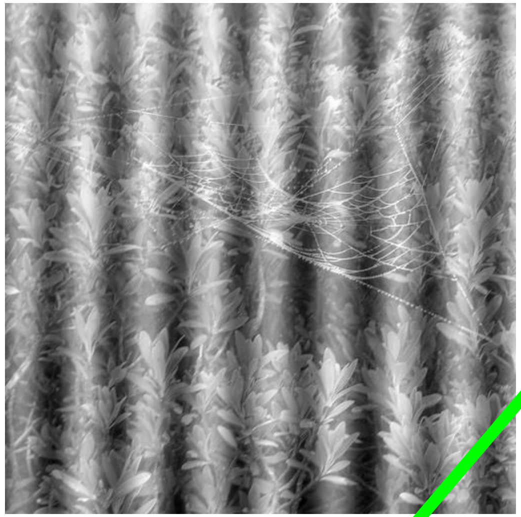
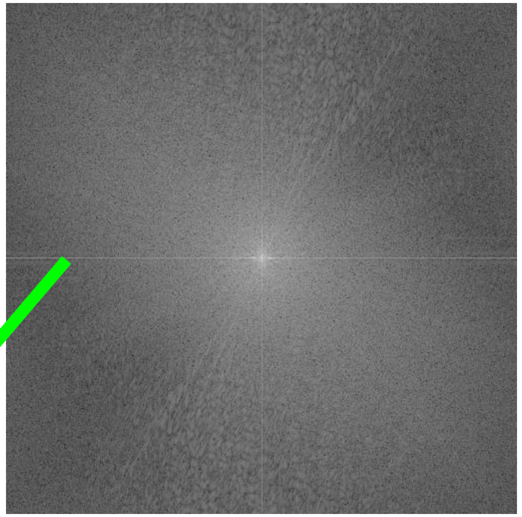
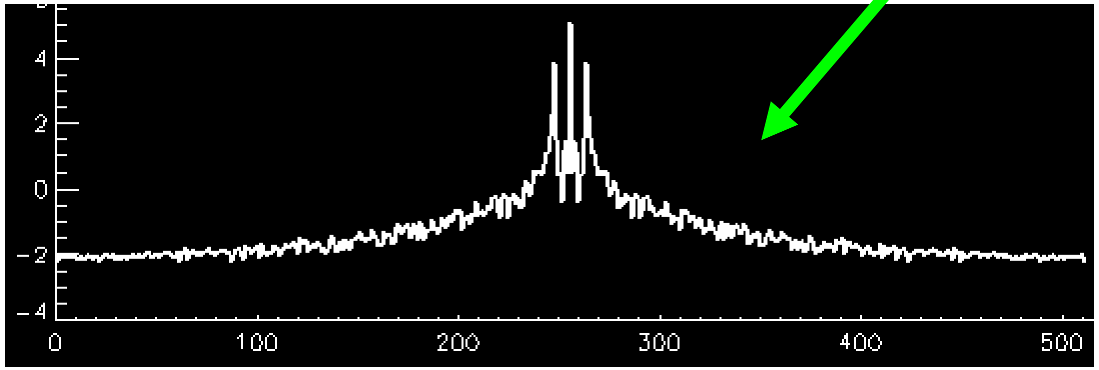
Famous Last Question

unbekannte
überlagernde Störung







Was tun??

Bildkompression

- Bilddaten sind ein Mittel, um Information zu vermitteln.
- Information ist in der Regel redundant kodiert.
- Reduktion von redundanter Daten
  - Codierungsredundanz (Anzahl der Bits zu Grauwertkodierung).
  - Interpixelredundanz (Anzahl der Datenpunkte pro Pixel).
  - Psychovisuelle Redundanz (zur Erkennung benötigte Bildinformation)

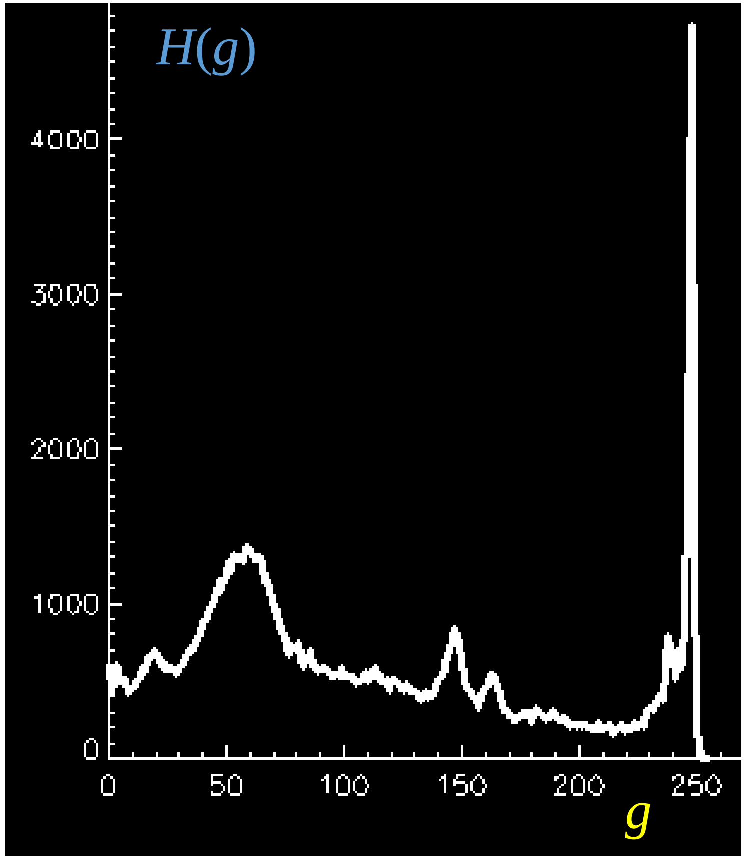
- Kompressionsrate: $C = n_1/n_2$
  - n_1 und n_2 sind die Anzahl der Informationseinheiten, um dieselbe Information zu kodieren.

Codierungsredundanz

- Anzahl der verfügbaren Codes ist größer als die der benötigten Codes.
- Beispiel (jedes Pixel = ein Byte)
  - nur Grauwerte 1 bis100 sind im Bild vorhanden.
  - 90% der Pixel haben den Grauwert 100.
- Lösung: 1-Bit-Code für 100, 9-Bit-Code für alle anderen
  - (100): 0, (1): 100000001, (2): 100000010, (3): 100000011, (4): 100000100, …
  - (100) (100) (100) (17) (15) (100)…: 0001000010011000011110…
  - Kompression: 8/(0.9*1+0.1*9)=8/1.8≈4.4
- Ziele
  - Berechnung der Redundanz (Berechnung des Informationsgehalts)
  - Reduzierung der Codelänge.

Histogramm

Häufigkeit $H(g)$ der Grauwerte $g=\{0,1,...,N\text{-}1\}$ in einem Bild.

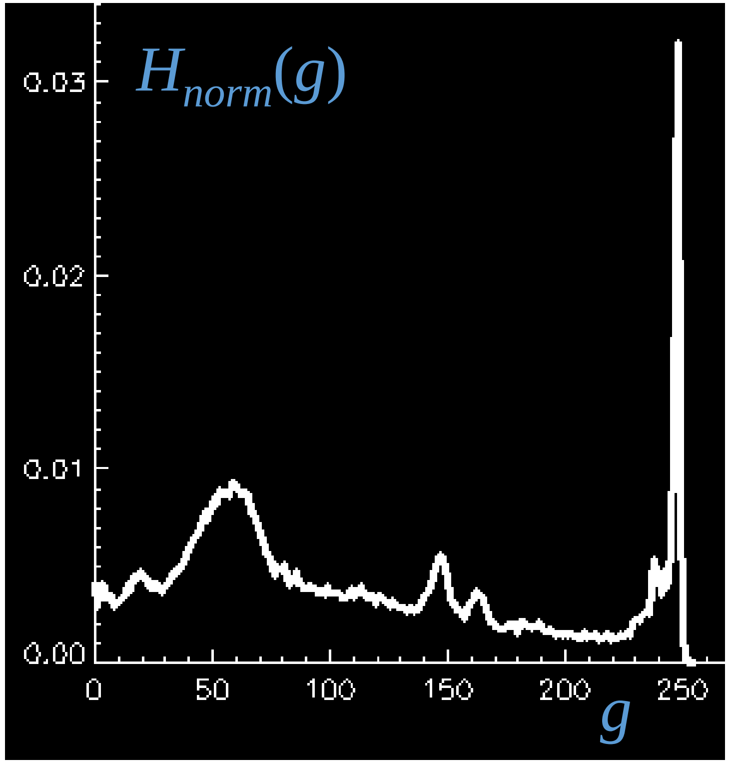


Normiertes Histogramm

- Normierung nach Anzahl der Pixel eines Bildes (Größe $M \times N$):

$$H_{norm}(g) \; = \; H(g) \, / \, (M \cdot N)$$

- Ein normiertes Histogramm gibt für jeden Grauwert g die Wahrscheinlichkeit an, dass ein beliebiges Pixel diesen Grauwert hat.



Informationsgehalt

Messbare Einheit von Information mit intuitiver Bedeutung.

Ein erster Ansatz:

Informationsgehalt *IG*(*E*) eines Grauwerts *E* ist umso höher, je größer die Gesamtanzahl *M* der verwendeten Grauwerte ist:

- $IG_M(E) = M.$
- Informationsgehalt ist unabhängig davon, welcher Grauwert aus der Liste $E = \{E_0, E_1, \ldots, E_{M-1}\}$ übermittelt wurde.
- Informationseinheiten $I_M(E)$ = Anzahl der benötigten *n*-wertigen Symbole, für die Speicherung des Informationsgehalts, also $\boldsymbol{I_M(E) = log_n IG_M(E)}$.
- Beispiel:
    - Anzahl der Grauwerte: 256
    - Informationsgehalt jedes Grauwerts: 256
    - Symbol: Bit (2-wertig)
    - Benötigte Informationseinheiten: $\log_2 256 = 8$

Informationsgehalt

- **Nachteil**: Informationsgehalt eines häufig vorkommenden Grauwerts ist genauso groß wie die eines selten vorkommenden Werts.
- Informationsgehalt $IG(E)$ eines Pixelwerts *E* unter *Berücksichtigung der Häufigkeit* von *E*:
  - Umgekehrt proportional zur Wahrscheinlichkeit *P*(*E*) des Eintreffens $IG(E) = 1/P(E)$.
  - Anzahl der benötigten Informationseinheiten ist dann $I(E) = log_n 1/P(E) = -log_n P(E)$
- Zur Repräsentation der Information $IG(E)$ werden $I(E)$ Informationseinheiten benötigt.
- Beispiel für Schwarzweißbilder mit gleicher Anzahl schwarzer und weißer Pixel:
  - Wahrscheinlichkeit für Eintreffen von *E* ist 0.5
  - Informationsgehalt ist dann $I(E) = -\log_2 0.5 = \log_2 2 = 1.$

Information in einer Pixelfolge

- Grauwertbereich $\{g_0, g_1, ..., g_{N\text{-}1}\}$ mit Wahrscheinlichkeiten des Auftretens $\{P(g_0), ..., P(g_{N\text{-}1})\}$
- Information einer Pixelfolge der Länge k – Wahrscheinlichkeit des Auftretens gewichtet mit der Anzahl der benötigten Informationseinheiten:

$$-k \cdot P(g_0) \cdot \log_2 P(g_0) - k \cdot P(g_1) \cdot \log_2 P(g_1) - k \cdot P(g_2) \cdot \log_2 P(g_2) \ldots = -k \sum_{i=0}^{N-1} P(g_i) \cdot \log_2 P(g_i)$$

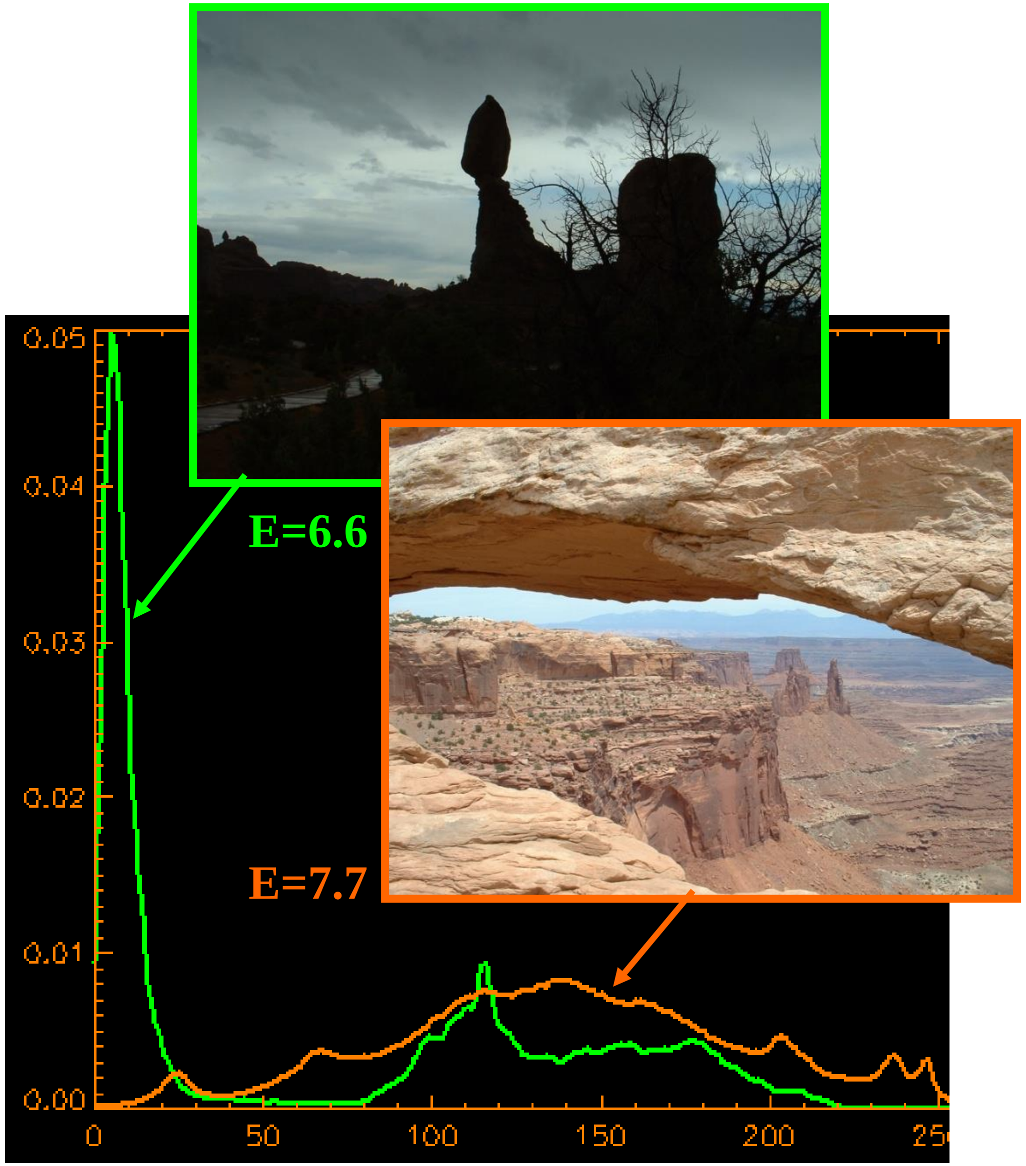
- Durchschnittlicher Informationsgehalt in Informationseinheiten = *Entropie*:

$$Entropie(P) = -\sum_{i=0}^{N-1} P(g_i) \cdot \log_2 P(g_i)$$

- Das normierte Histogramm kann als Schätzung für P verwendet werden.

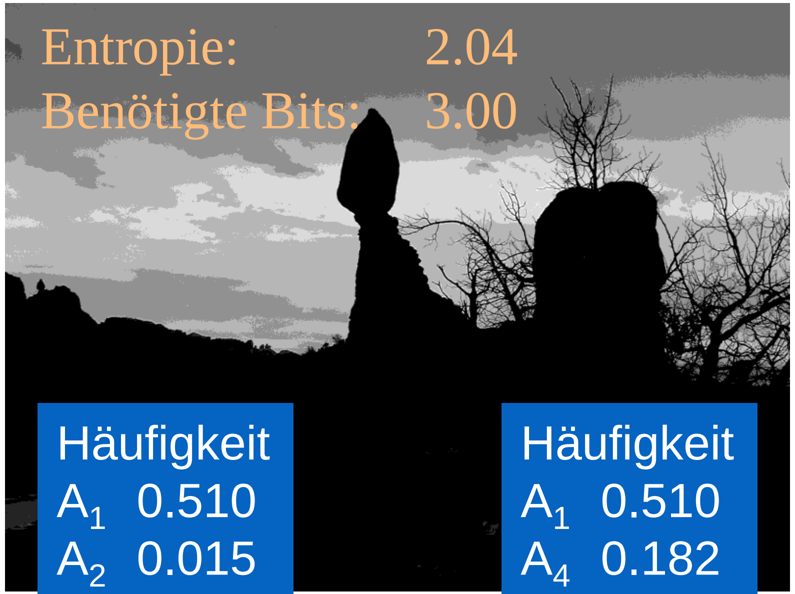
Codierungs-Redundanz

- Konstante Code-Längen sind nur bei gleichverteilten Grauwerthäufigkeiten optimal.
- Code-Länge sollte an Häufigkeit angepasst werden.
- Maximale erreichbarer Kompressionsfaktor:
  #Bits/Entropie



Huffman Kodierung

- Berechne Häufigkeitsverteilung (normiertes Histogramm)
- Eingangssymbole nach Häufigkeit ordnen
- Die zwei seltensten Symbole zu einem kombinierten Symbol zusammensetzen.
- Den Prozess solange fortsetzen, bis nur zwei Symbole existieren.



Häufigkeit	
A_1	0.510
A_2	0.015
A_3	0.010
A_4	0.182
A_5	0.104
A_6	0.131
A_7	0.047
A_8	0.001

Sortieren →

Häufigkeit	
A_1	0.510
A_4	0.182
A_6	0.131
A_5	0.104
A_7	0.047
A_2	0.015
A_3	0.010
A_8	0.001

Reduktionsschritt

	Häufigkeit					
A_1	0.510	0.510	0.510	0.510	0.510	0.510	0.510
A_4	0.182	0.182	0.182	0.182	0.182	0.308	0.490
A_6	0.131	0.131	0.131	0.131	0.177	0.182	
A_5	0.104	0.104	0.104	0.104	0.131		
A_7	0.047	0.047	0.047	0.073			
A_2	0.015	0.015	0.026				
A_3	0.010	0.011					
A_8	0.001						

Nächster Schritt:
Code-Verteilung nach Häufigkeit

Code-Zuteilung

Häufigkeit						
A_1 0.510 **1**	0.510 **1**	0.510 **1**	0.510 **1**	0.510 **1**	0.510 **1**	0.510 **1**
A_4 0.182 **00**	0.182 **00**	0.182 **00**	0.182 **00**	0.182 **00**	0.308 **01**	0.490 **0**
A_6 0.131 **010**	0.131 **010**	0.131 **010**	0.131 **010**	0.177 **011**	0.182 **00**	
A_5 0.104 **0111**	0.104 **0111**	0.104 **0111**	0.104 **0111**	0.131 **010**		
A_7 0.047 **01101**	0.047 **01101**	0.047 **01101**	0.073 **0110**			
A_2 0.015 **011001**	0.015 **011001**	0.026 **01100**				
A_3 0.010 **0110001**	0.011 **011000**					
A_8 0.001 **0110000**						

Code-Verteilung nach Häufigkeit

Benötigter Speicherplatz

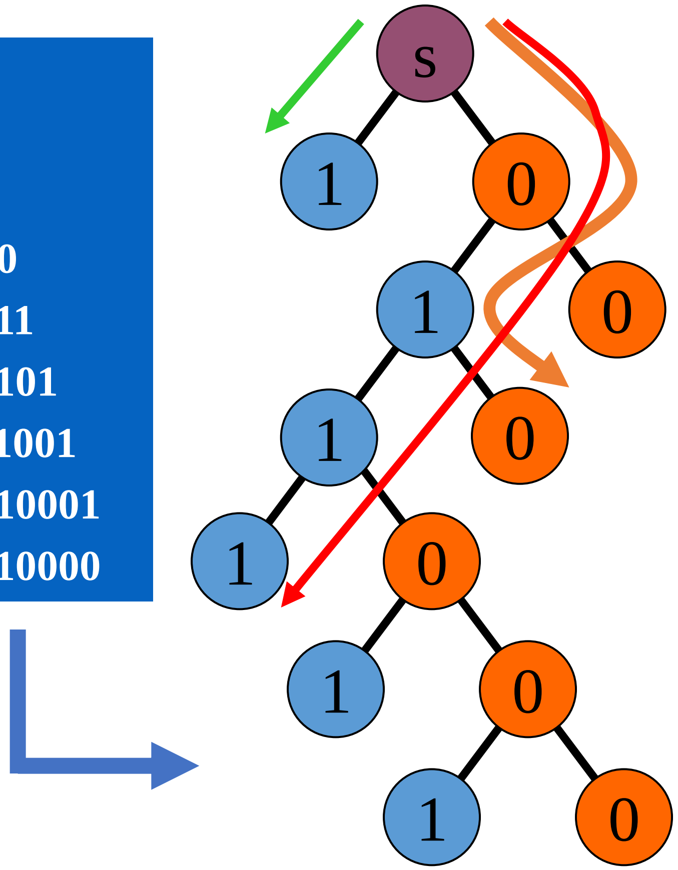
$1 \cdot 0.510 + 2 \cdot 0.182 + 3 \cdot 0.131 + 4 \cdot 0.104 + 5 \cdot 0.047 + 6 \cdot 0.015 + 7 \cdot 0.010 + 7 \cdot 0.001 = 2.079$

Bei gleicher Code-Länge: = 3.000

Entropie: = 2.041

Dekodierung

Häufigkeit		
A_1	0.510	**1**
A_4	0.182	**00**
A_6	0.131	**010**
A_5	0.104	**0111**
A_7	0.047	**01101**
A_2	0.015	**011001**
A_3	0.010	**0110001**
A_8	0.001	**0110000**



Baumrepräsentation

1 | 0 1 0 | 0 1 1 1 | 0 1 0 1 ...

Während des Dekodierens wird der Kodierungsbaum anhand der Bitfolge abgearbeitet.

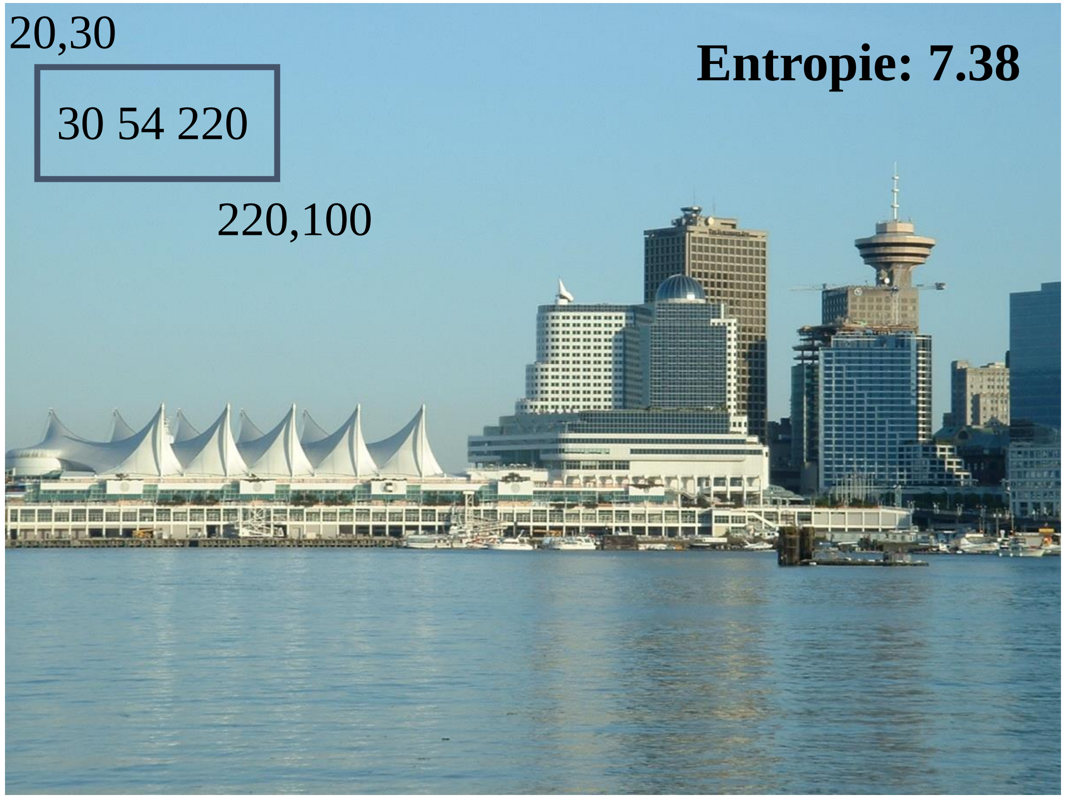
Ein Zeichen ist gefunden, wenn ein Blatt erreicht wurde.

Aufwand $O(N)$, wenn N die Anzahl der Bits ist.

Interpixel-Redundanz

- Minderung der Kodierungsredundanz berücksichtigt die *Homogenität* innerhalb des Bilds *nicht*.
- Interpixelredundanz: Repräsentation durch
  - Blockspezifikation (z.B. [20,30,220,100])
  - Blockintensität (z.B. [30 54 220])
- Bekannteste Methode:

  Run-Length-Encoding (RLE), bzw.
  Lauflängenkodierung



Run-Length-Encoding

- Bild wird zeilenweise codiert.

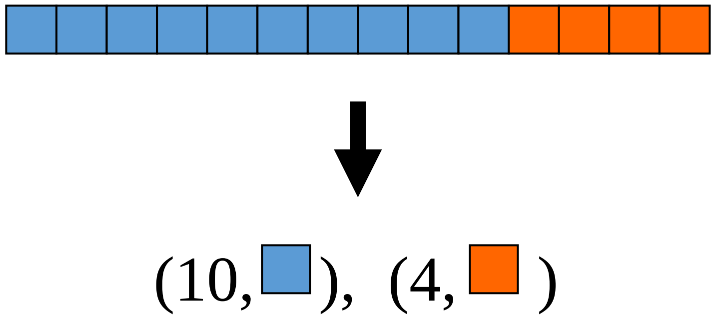
- Eine Zeile mit Grauwertfolge $g(0), g(2), \ldots, g(n)$ wird zerlegt in Zeilenstücke mit gleichem Grauwert (die *runs*):

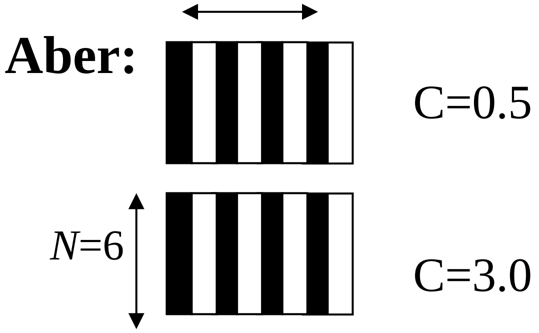
$$(g(0), \ldots, g(r_1)), (g(r_1+1) \ldots g(r_1+1+r_2)), \ldots$$

- Die Zeilenstücke werden durch Angabe von Länge und Grauwert codiert:

$$(r_1, g(r_1)), (r_2, g(r_2)), \ldots$$

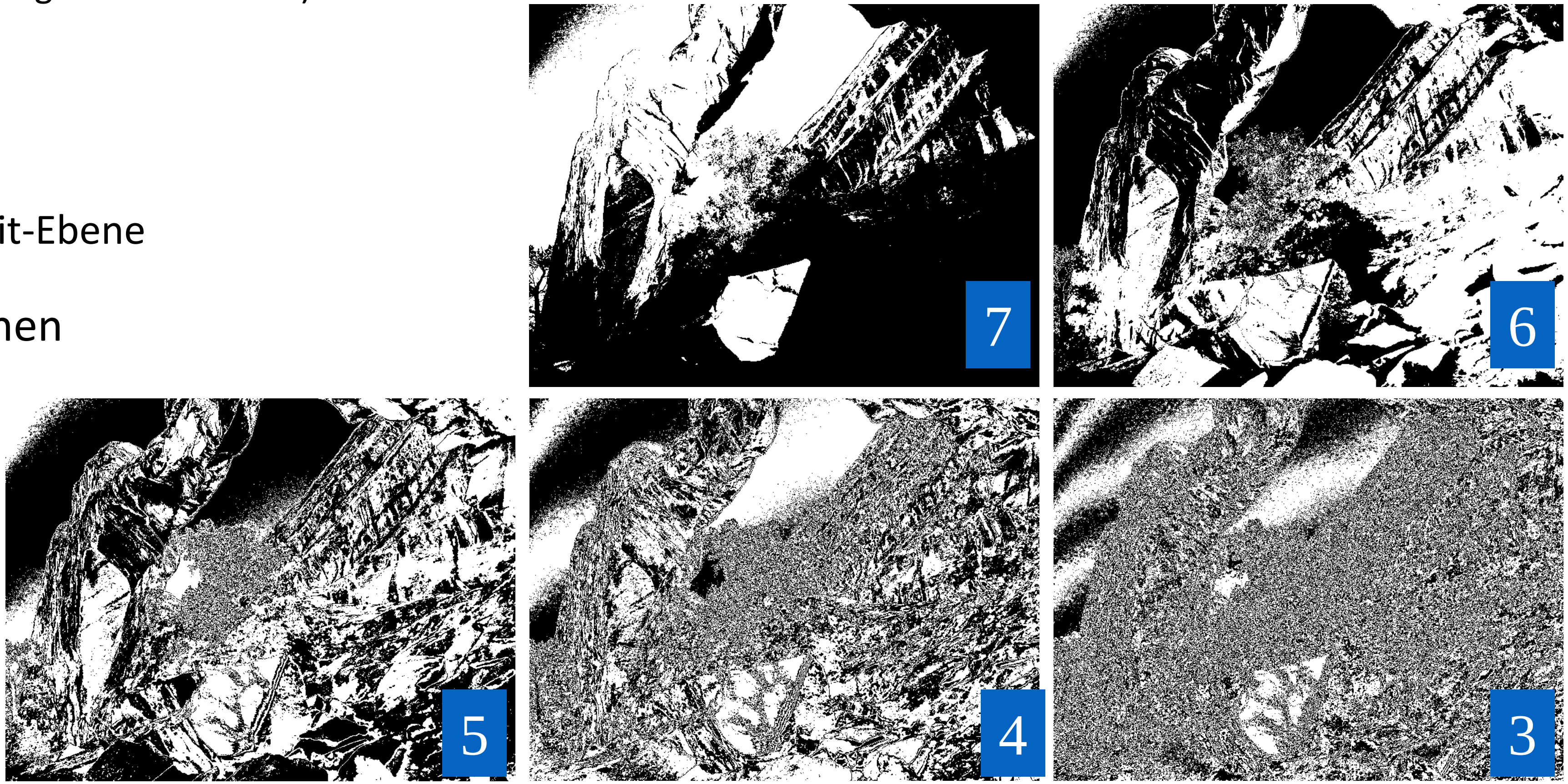
- Kompressionsrate für jede Zeile ist $n/(2 \cdot rd)$, wobei r_d die durchschnittliche Länge des *runs* ist.





RLE auf Bit-Ebenen

- RLE ist ineffizient bei verrauschten Bildern
  (sehr viele kleine Grauwertänderungen = kurze Runs).
- Lösung:
  - Zerlegung in Bit-Ebenen
  - Runs separat auf jeder Bit-Ebene
- Problem: Niedrige Bit-Ebenen



Gray-Code

- Code, bei dem sich zwei aufeinander folgenden Zahlen um höchstens 1 Bit unterscheiden.
- Generierung für Bitfolge $b_1, b_2, \dots b_n$ (höchstwertiges Bit hat niedrigsten Index):

  - $k=n,\dots 2$: falls $g_{k-1}=1$, dann $g_k=1-b_k$ sonst $g_k=b_k$
  - $k=1$: $g_1=b_1$

Beispiel:

b: 000 001 010 011 100 101 110 111

g: 000 001 011 010 110 111 101 100

Beispiel

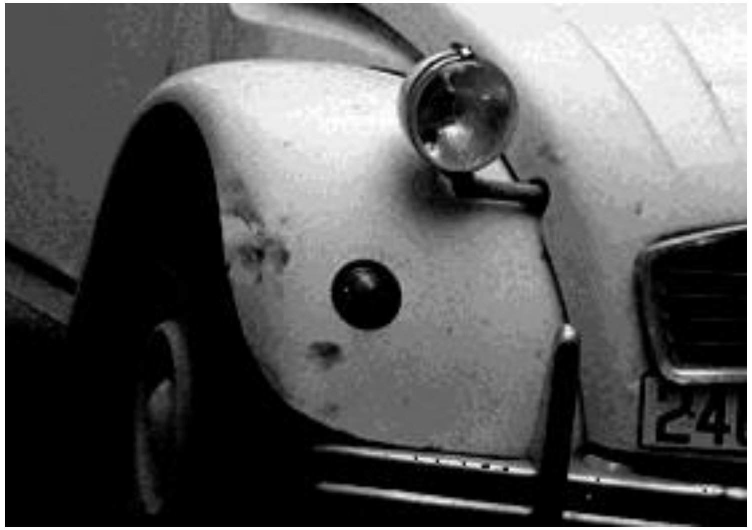
Normal und gray-kodierte untere drei Bit-Ebenen des Rotkanals.

Psychovisuelle Redundanz

- Nichtredundante Daten tragen redundante Information.
- Reduktion der psycho-visuellen Redundanz ist verlustbehaftet (lossy) bezogen auf die Daten.
- Erhalt des Informations-gehalts ist zielabhängig.

Reduktion der Kontrastauflösung

- Unterscheidbare Helligkeitsstufen (30-100) ist geringer, als die repräsentierbare Anzahl.
- Bildinformation ist auch bei noch kleinerer Anzahl wahrnehmbar.
- Machbandeffekt kann durch Addition von Rauschen vermieden werden.



Reduktion hochfrequenter Anteile

Transformation zerlegt in einzelne Frequenzanteile (z.B., Fouriertransformation, Diskrete Kosinustrans-formation, Wavelet Transformation).

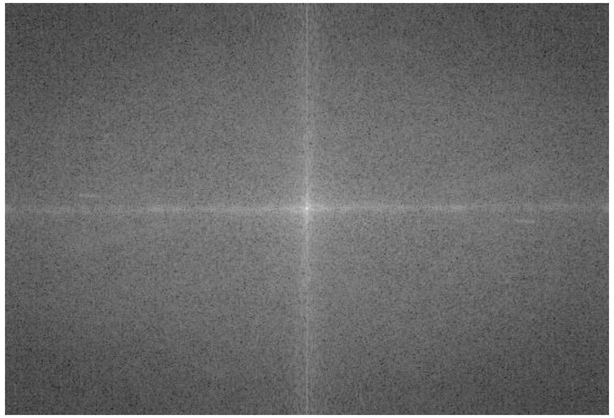
Sehr viel der Bildinformation ist in den niedrigen Frequenzen kodiert.

Kompression:

- Transformation
- Sortierung der Koeffizienten nach Größe
- Quantisierung, d.h. Abbildung auf ganze Zahlen:

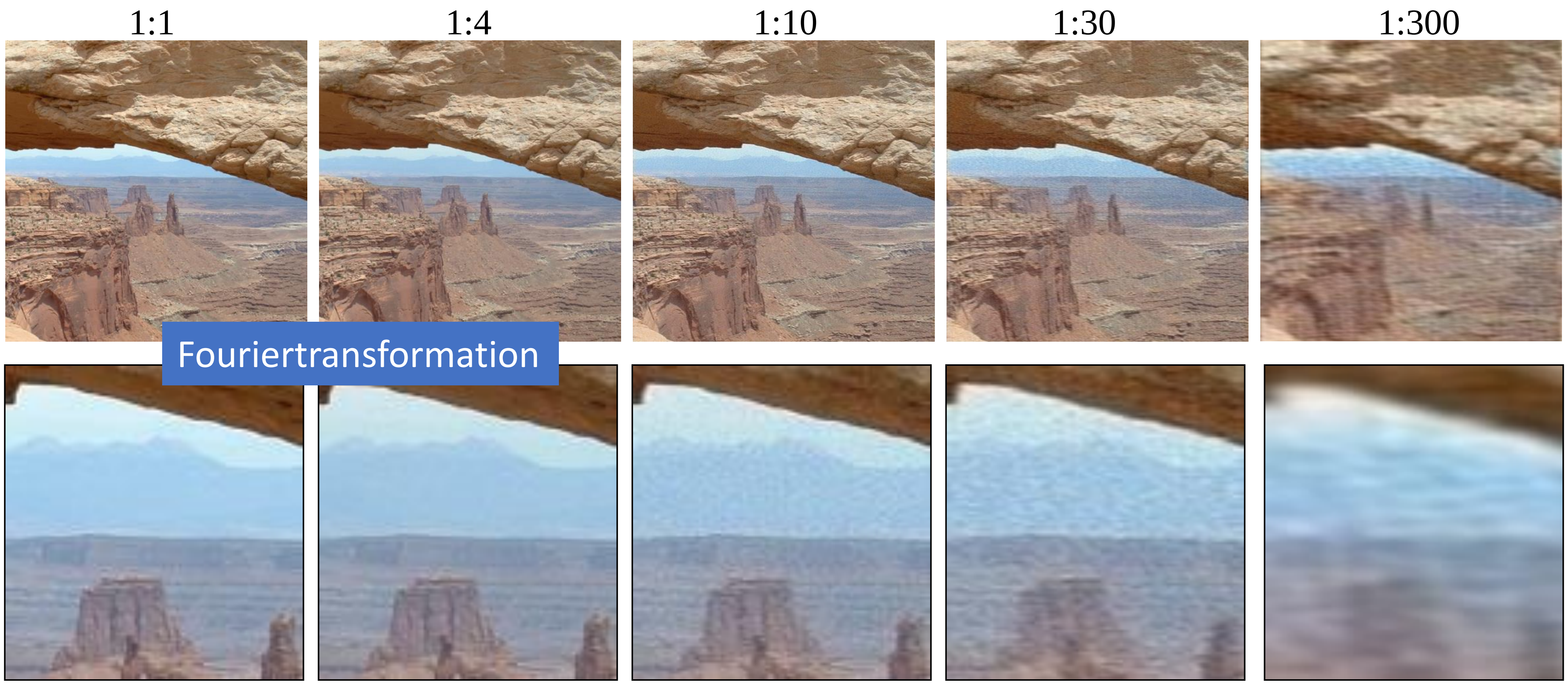
$$q(e) = Q \cdot sign(e) \cdot (|e| - e_{min}) / (e_{max} - e_{min}).$$

- Kompression der quantisierten Werte durch verlustfreies Verfahren.



Dekodierung durch Rücktransformation.

Reduktion hochfrequenter Anteile



Diskrete Kosinustransformation (DCT)

- Zerlegung in Wellen unterschiedlicher Frequenz.
- Sehr ähnlich zur Fouriertransformation.
- Basisfunktionen sind reell.
- Wird vor allem bei Kompressionsverfahren verwendet (JPEG).

- ***Wie lassen sich genügend reellwertige Basisfunktionen finden?***

Basisfunktionen

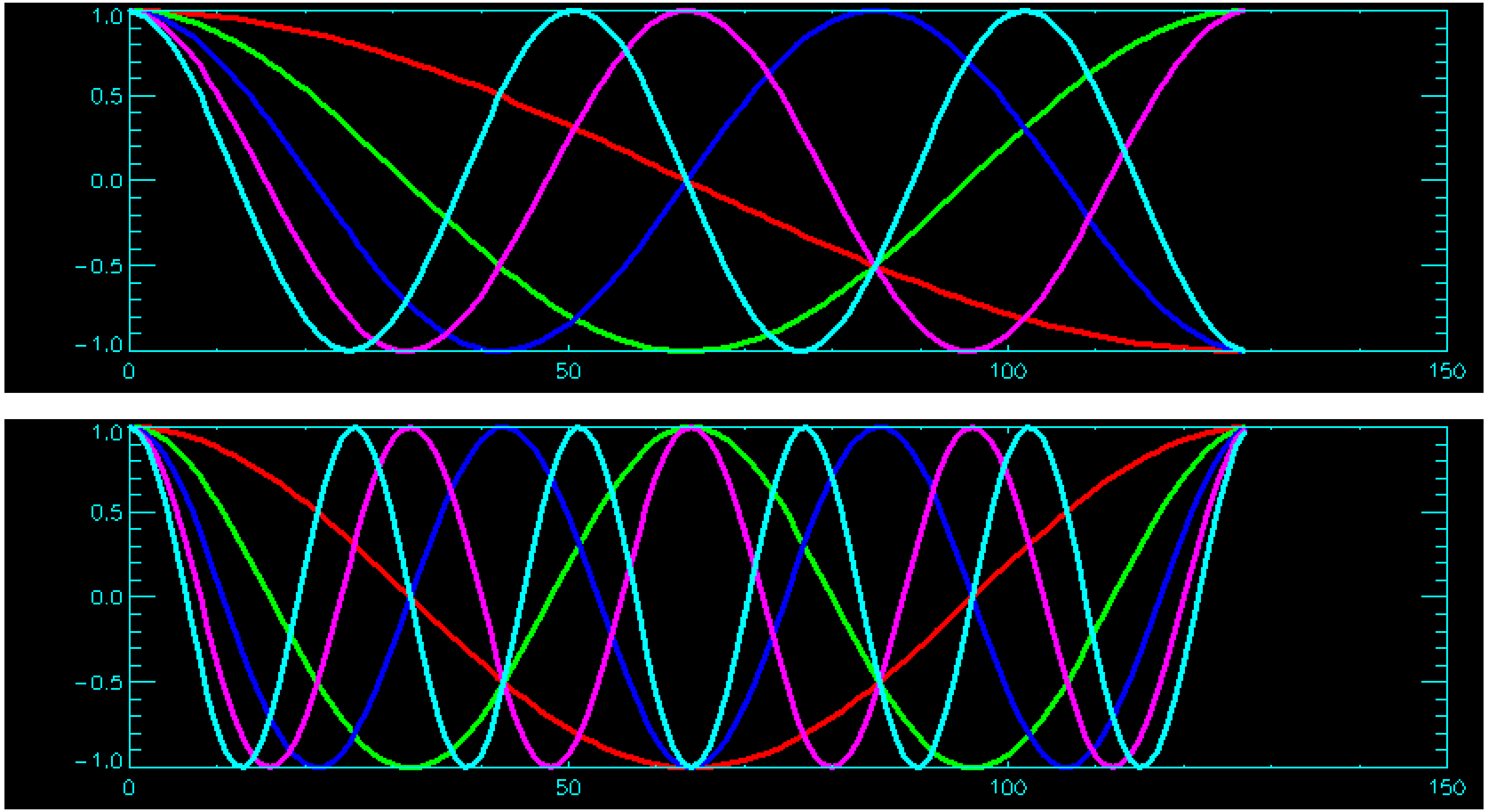
- Basisfunktionen sind alle halben und alle vollen Frequenzen der Cosinustransformation

$$C(u) = \sum_{m=1}^{M-1} f(m)\cos\left(\frac{mu\pi}{M}\right)$$

- Meist (z.B. auch bei der jpeg-Transformation) werden die Transformationswerte aus den Funktionswerten zwischen den Abtastpunkten gemittelt, d.h., m wird um 0.5 verschoben

$$C(u) = \sum_{m=1}^{M-1} f(m)\cos\left(\frac{(m+0.5)u\pi}{M}\right)$$

Reelle Basisfunktionen DCT / DFT



Basisfunktionen der 2-d DCT

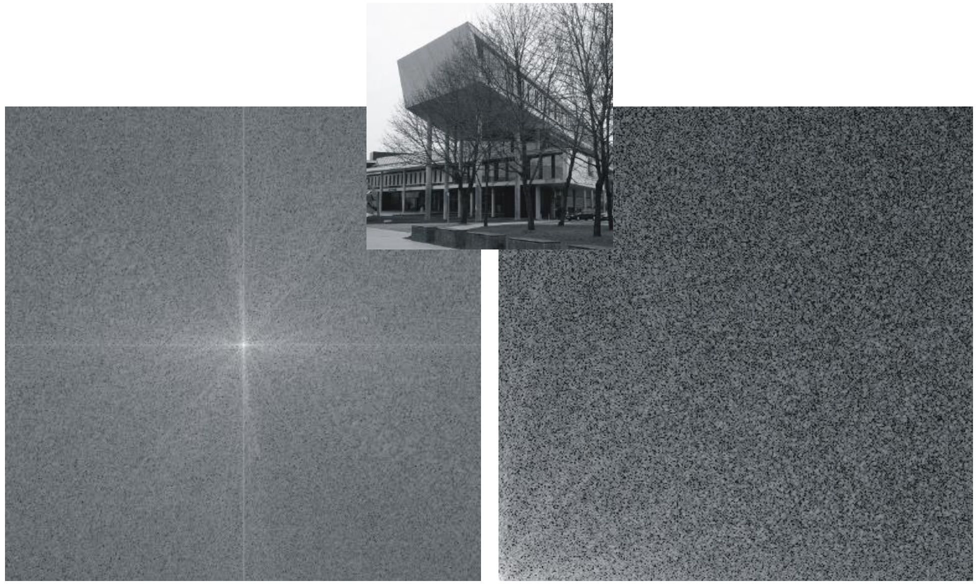
$$C(u,v) = \alpha(u)\alpha(v)\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\cos\left(\frac{(m+0.5)u\pi}{M}\right)\cos\left(\frac{(n+0.5)v\pi}{N}\right)$$

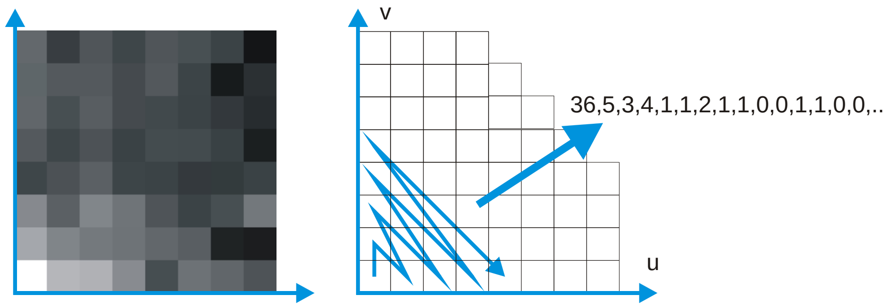
$$f(m,n) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)C(u,v)\cos\left(\frac{(u+0.5)m\pi}{M}\right)\cos\left(\frac{(v+0.5)n\pi}{N}\right).$$

$$\alpha(u) = \begin{cases} \sqrt{1/M} & \text{, für } u = 0 \\ \sqrt{2/M} & \text{, für } u \neq 0 \end{cases} \quad \text{und} \quad \alpha(v) = \begin{cases} \sqrt{1/N} & \text{, für } v = 0 \\ \sqrt{2/N} & \text{, für } v \neq 0 \end{cases}$$

DCT vs DFT

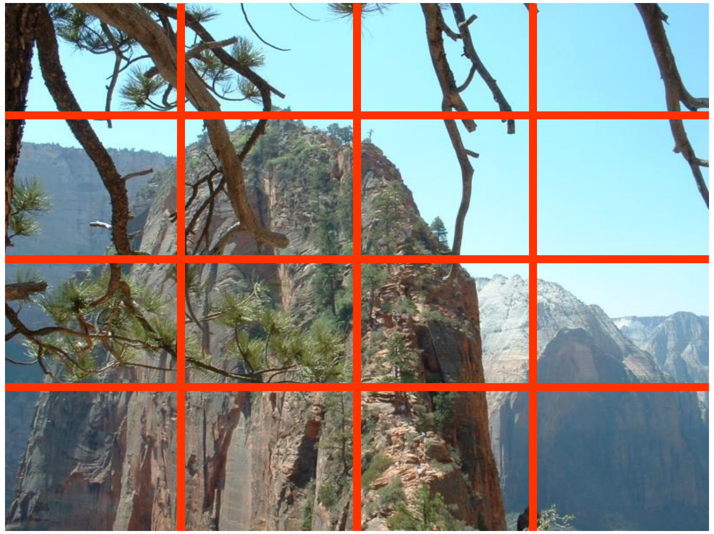
- DCT hat nur reelle Komponenten, deren Wert mit Abstand vom Ursprung abnimmt.
- Kompression durch Quantisierung:
  - Werte entlang der Diagonale auslesen,
  - Quantisieren,
  - nur bis zum letzten von Null verschiedenen Wert speichern.





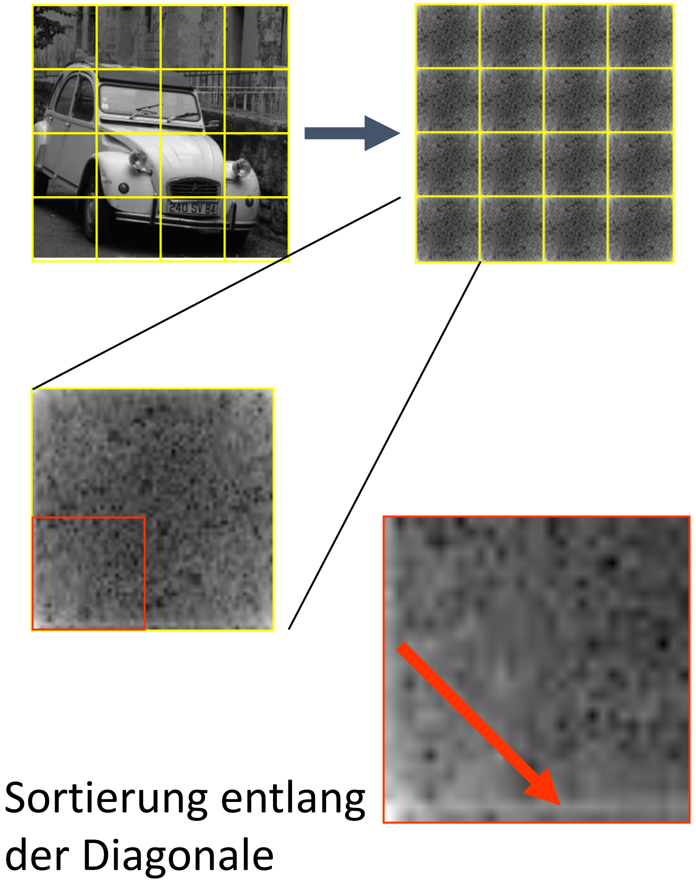
Zerlegung in Teilbilder

- Anzahl der Information tragenden Koeffizienten ist geringer, je geringer die Anzahl der Kanten ist.
- Zerlegung in Teilbilder
  - Anzahl der Funktionswerte im Frequenzraum entspricht der Anzahl der Pixel des Teilbilds.
  - Wahrscheinlichkeit von Kanten in Teilbildern sinkt mit deren Größe.
  - Teilbildgröße: Kompromiss zwischen durchschnittlicher Anzahl der Information tragenden Frequenzen und Gesamtanzahl der Teilbilder.



JPEG Kompression von Teilbildern

- Farbraumtransformation von RGB nach YCbCr
- Tiefpassfilterung und Unter-abtastung der Farbabweichungs-signale Cb und Cr.
- Einteilung in 8×8-Blöcke und DCT auf dem Y-Kanal
- Quantisierung der DCT-Koeffizienten.
- Umsortierung und Entropiekodierung (Huffman).

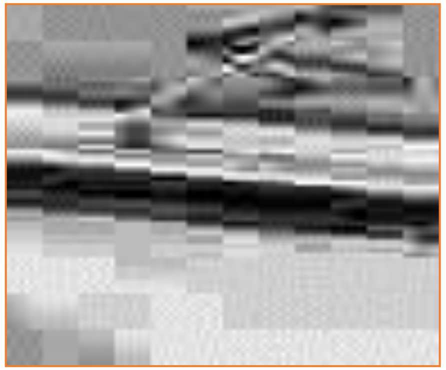


Quantisierungseffekte

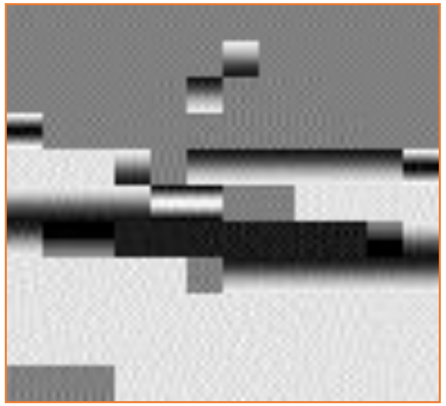
Zerlegung in 8x8-Blöcke, Quantisierung und Rücktransformation

1:30

1:45



1:75



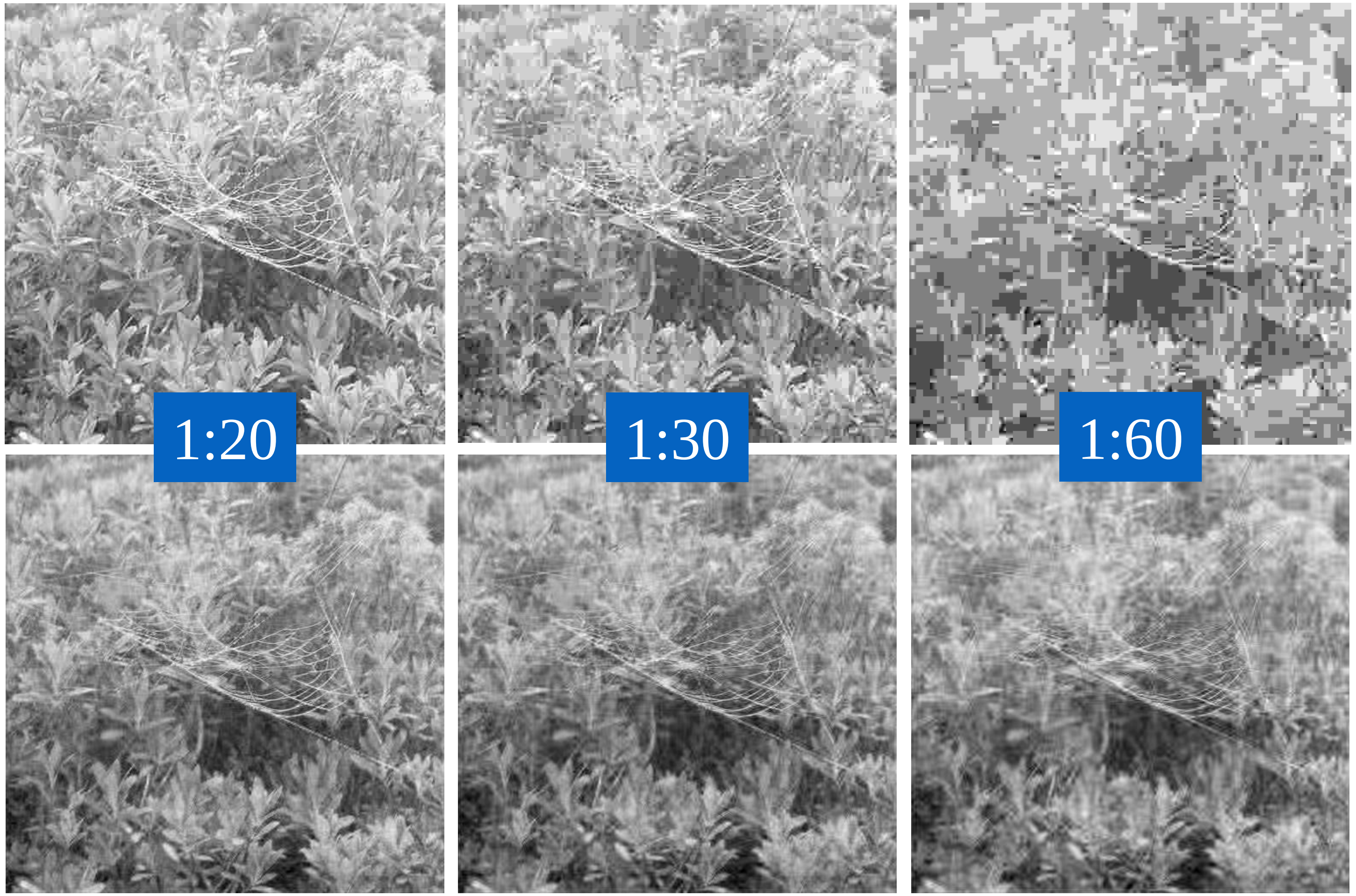
JPEG2000: Wavelet-Koeffizienten

- Wavelet-Transformation zerlegt das Bild in lokal wirkende Wellen unterschiedlicher Frequenz.
- Keine explizite Blockung notwendig.
- Kompression wird umso besser sein, umso kompakter die Wavelets in Orts- und Frequenzraum sind.
- Wavelet-Kompression durch Daubechies-Wavelets höherer Ordnung (Wavelet-Kompression ist Teil des JPEG2000-Standards).

Vergleich Wavelet vs DCT

DCT-basierte Kompression

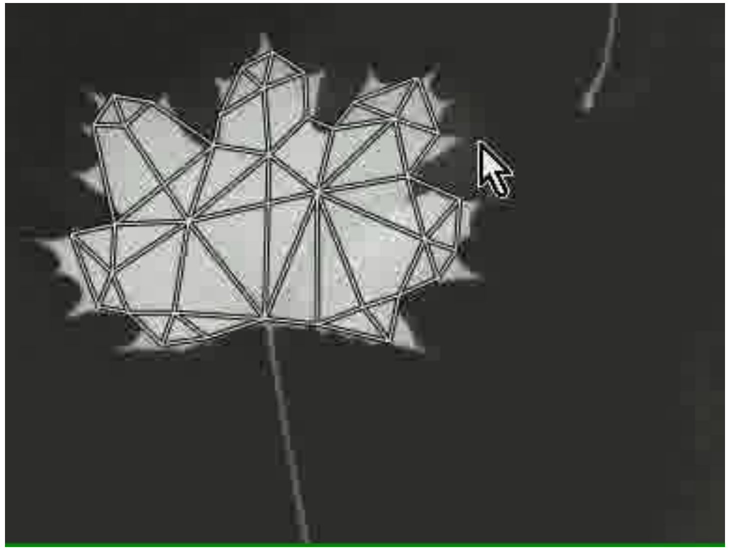
Wavelet-Kompression



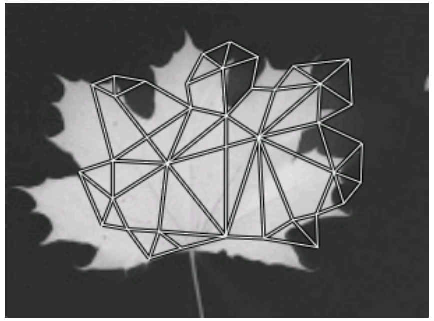
Was sollten Sie heute gelernt haben?

- Kompression entfernt Redundanz aus Bildern
- Redundanz: Kodierungs-, Interpixel-, Psychovisuelle Redundanz.
- Kodierungs- und Interpixelredundanz ist Datenredundanz (=verlustfreie Kompression).
- Psychovisuelle Redundanz ist Informationsredundanz (=verlustbehaftete Kompression).

Famous Last Question

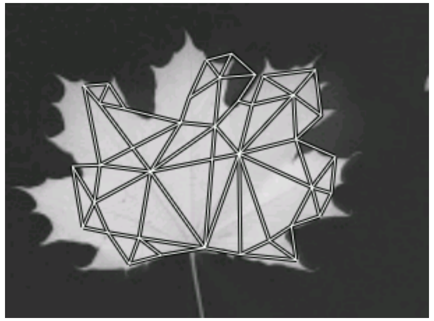


Wie könnte man eine Sequenz von Filmbildern besonders gut *verlustfrei* komprimieren?



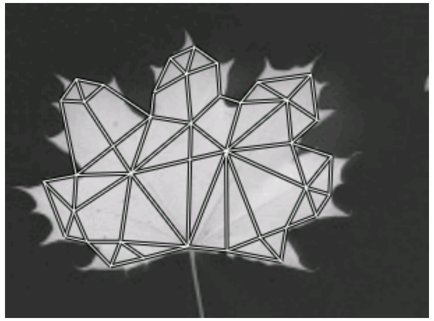
…

Bild 1



…

Bild 41



…

Bild 81