

# Graphische Datenverarbeitung II

## Bildkompression & Dateiformate II

Prof. Dr. Elke Hergenröther

## Komperssionsverfahren

Nicht verlustbehaftete Verfahren:

- Run Length Encoding (RLE)
- Lempel-Ziv-Welch (LZW)
- Huffman Kodierung

Verlustbehaftete Verfahren:

- **Fraktale Kompression**
- JPEG
- JPEG 2000 („Wavelet-Kompression“)

Prof. Dr. Elke Hergenröther 2

## Fraktale von F. Musgrave



Prof. Dr. Elke Hergenröther 3

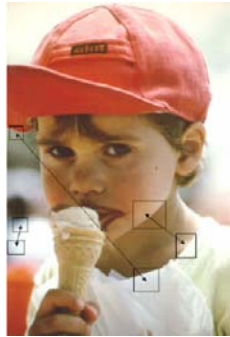
## Basis der Fraktalen Kompression: Collage-Theorem

**Collage-Theorem** (Michael Barnsley):

- Jede beliebige Figur eines Bildes kann man aus verkleinerten Figuren ihrer selbst zusammensetzen.
- Man muss sich nur die entsprechenden **Transformationen** merken.

Prof. Dr. Elke Hergenröther 4

## Fraktale Komprimierung: Codierung



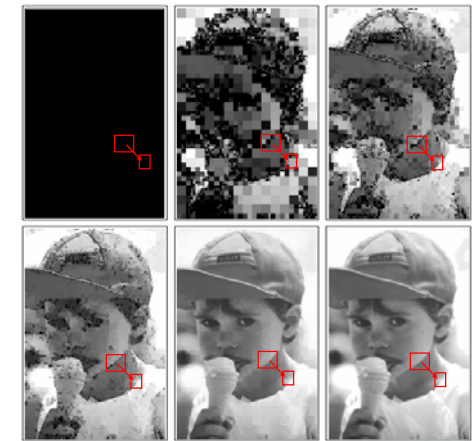
Suche von Selbstähnlichkeiten in Bildern:

- Dazu muss man zunächst per Suchverfahren die korrespondierenden Flächen ermitteln (Tafel).
- **Sehr Zeitaufwendig**

Affine Transformationen speichern:

- Abbildungsvorschriften entwickeln, wie flächenmäßig **größere Bildbereiche auf flächenmäßig kleinere Bildbereiche transformiert** werden: Helligkeit, Rotation, Skalierung, Translation (Verschiebung)

## Fraktale Komprimierung: Decodierung



Asynchrones Verfahren:  
Decodierung geht sehr schnell



### Fraktale Kompression:

Gute Bildqualität bis zu einer Kompression von 1:35

Auch bei geringen Kompressionsraten werden einfarbige Flächen mit dunklen und hellen Flecken bedeckt

Das schränkt die Anwendung ein:  
Nicht geeignet für Röntgenbilder und CT.

## Fraktale Kompression: **Vorteile**

- Hohe Qualität bei hohen Kompressionsraten
- Unterstützung von Zooming
- Funktionensystem interpoliert statt zu pixeln
- Skalierbarkeit
- Anzahl der Dekompressionsschritte steuert die Bildqualität

## Fraktale Kompression: **Nachteile**

- Lange Kompressionsdauer
- Stark asymmetrisches Verfahren
- Suchstrategien der Blockpaare sind aufwändig
- Blockartefakte mit Informationsverlust
- Nicht gut geeignet für beispielsweise Text, Linien, etc.
- Bei ungeeigneten Bildern: Schlechtere Qualität als JPEG bei niedrigeren Kompressionsraten

## Komperssionsverfahren

Nicht verlustbehaftete Verfahren:

- Run Length Encoding (RLE)
- Lempel-Ziv-Welch (LZW)
- Huffman Kodierung

Verlustbehaftete Verfahren:

- Fraktale Kompression
- **JPEG**
- JPEG 2000 („Wavelet-Kompression“)

## JPEG

Seit 1992 ist das JPEG-Verfahren zum weltweiten Standard in der Kompression von Farb- und Graustufenbildern geworden.

Die wichtigsten Anforderungen bei der Entwicklung des JPEG-Formats lauteten:

- Das Format soll sich für möglichst viele unterschiedliche Einsatzbereiche eignen.
- Die Bildqualität und der Kompressionsfaktor soll die Grenze des während der Entwicklung technisch Machbaren darstellen.
- Die Kompression muss unabhängig von Lage, Größe oder Farbanzahl des Bildes sein.
- Die mathematische Komplexität des Kompressionsalgorithmus soll gleichzeitig so gering wie möglich sein.
- Der Algorithmus muss auch in Hardware einfach zu realisieren sein (z.B. für Kompressionschips).
- Die Bildqualität soll durch die Auswahl eines Kompressionsfaktors einstellbar sein.

## JPEG

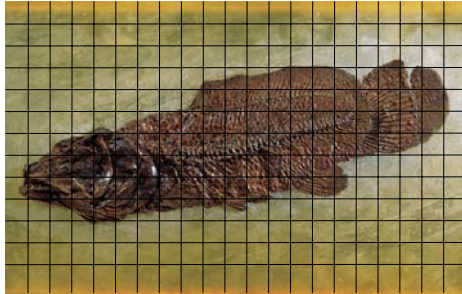
**JPEG ist gleichzeitig der Name**

- eines Standardisierungsorganisation,
- eines Grafikformates und
- eines Bildkompressionsverfahren

JPEG = Joint Photographic Experts Group

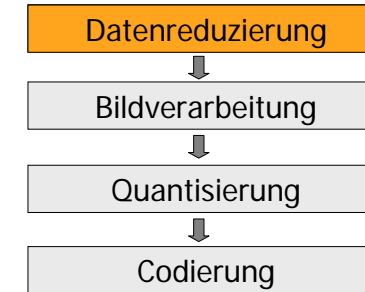
## JPEG: Blockorientiertes Verfahren

Zerlegen des Originalbildes in 8 x 8 große Blöcke



## JPEG

Die Komprimierung der Blöcke erfolgt in vier Schritten:



## JPEG

### 1. Schritt: Datenreduzierung

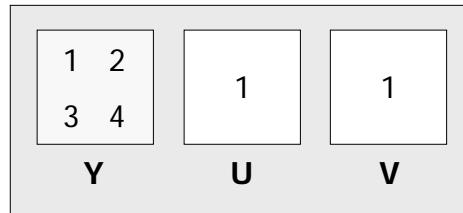
- Umwandlung: RGB Farbmodell in YUV Farbmodell
- entspricht einer Datenreduzierung von 50%

## JPEG: YUV- oder YIQ-Modell

- Das menschliche Auge besitzt mehr Rezeptoren für Helligkeitswerte als für Farbänderung.
- Daher können Farbwerte in geringerer Auflösung als Helligkeitswerte vorliegen.
- YUV- oder YIQ-Modelle unterstützen diese Eigenschaft:  
Y: Helligkeit (Grauwerte)  
I,Q bzw. U, V: Farbcodierung (Chrominanz)

## JPEG: YUV- oder YIQ-Modell

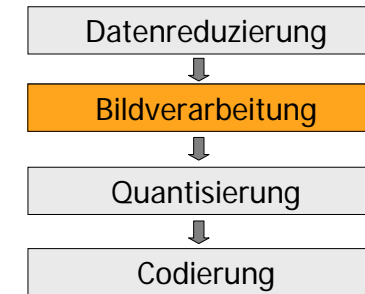
Codierung: 4:1:1-Farbraum im YUV-Modell



Komprimierungsrate: 50% (6 statt 12 Werte für 4 Pixel)

## JPEG

Die Komprimierung der Blöcke erfolgt in vier Schritten:



## JPEG

### 2. Schritt: Diskrete Kosinus-Transformation

- Abbildung in den Frequenzraum (Ortsfrequenzbereich)
- Abbildung ist bijektive (verlustfrei).

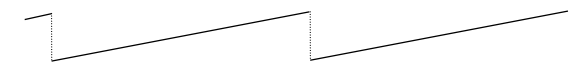
### Transformation vom Orts- in den Ortsfrequenzbereich

- Transformation der Grauwerte erfolgt zeilenweise
- Vor der Transformation der Farbbilder: Trennung der Kanäle
- Bildinformation wird in eine „andere“ Darstellungsweise überführt

Bildzeile



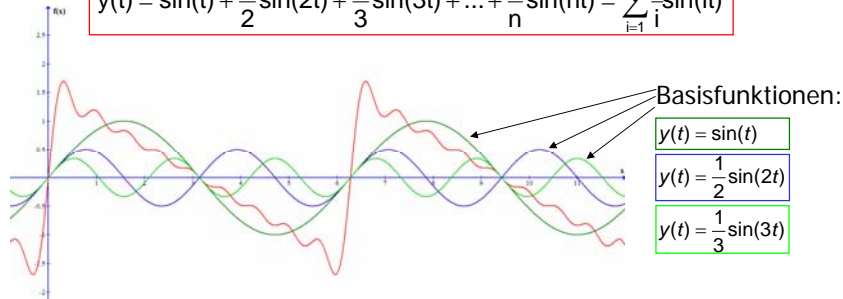
Funktion, die den Grauwertverlauf entlang dieser Bildzeile darstellt



## Transformation vom Orts- in den Ortsfrequenzbereich

Jede Funktion lässt sich als Überlagerung von Sinus- bzw. Kosinus-schwingungen darstellen:

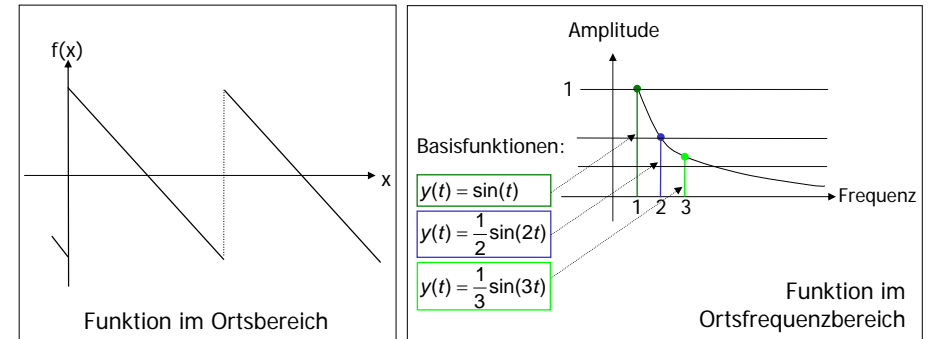
$$y(t) = \sin(t) + \frac{1}{2} \sin(2t) + \frac{1}{3} \sin(3t) + \dots + \frac{1}{n} \sin(nt) = \sum_{i=1}^n \frac{1}{i} \sin(it)$$



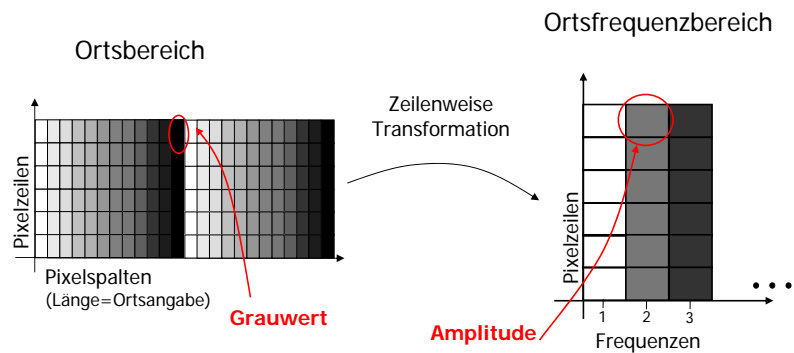
Probieren Sie es selbst aus: <http://padowan.dk/graph/>

## Transformation vom Orts- in den Ortsfrequenzbereich

**Frequenzspektrum der Funktion:**  $y(t) = \sin(t) + \frac{1}{2} \sin(2t) + \frac{1}{3} \sin(3t) + \dots + \frac{1}{n} \sin(nt) = \sum_{i=1}^n \frac{1}{i} \sin(it)$

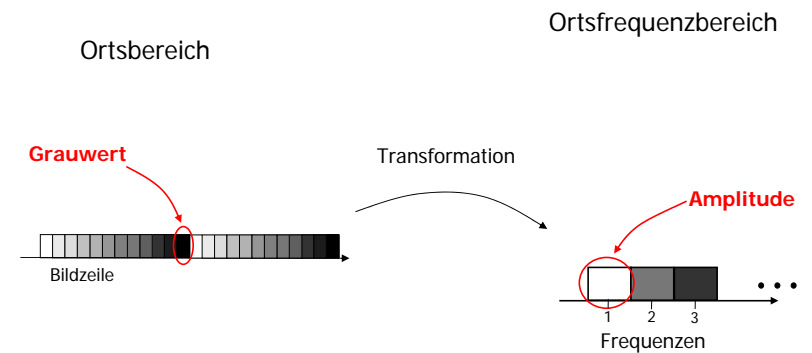


## Transformation vom Orts- in den Ortsfrequenzbereich

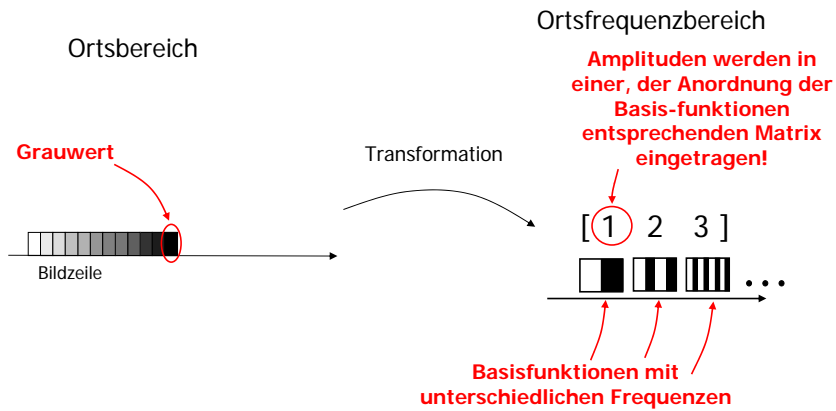


Achtung: Hier stellt der Grauwert die Amplitude der Basisfunktion dar!!!

## Transformation vom Orts- in den Ortsfrequenzbereich

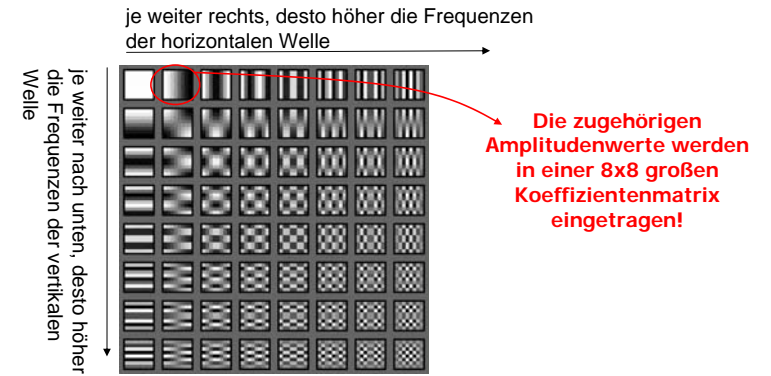


## Transformation vom Orts- in den Ortsfrequenzbereich

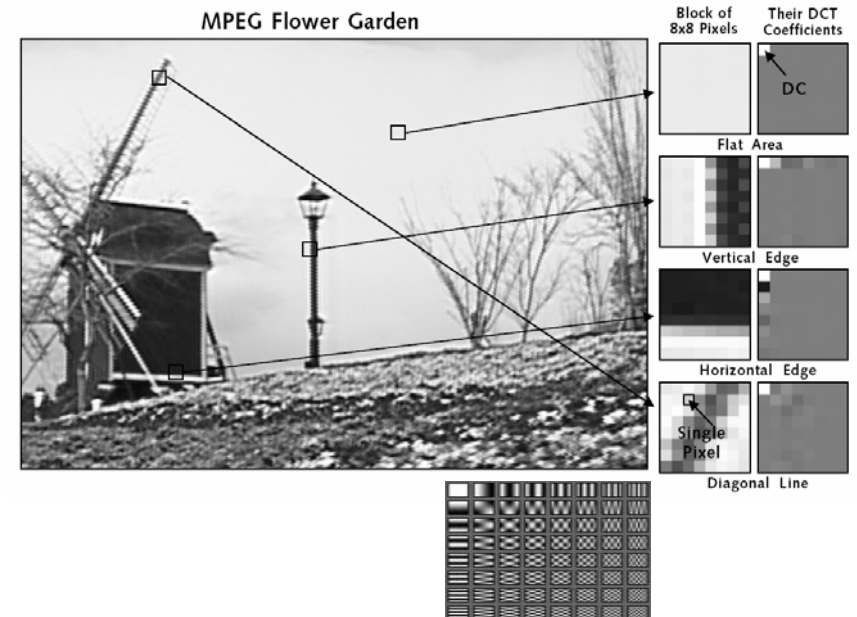
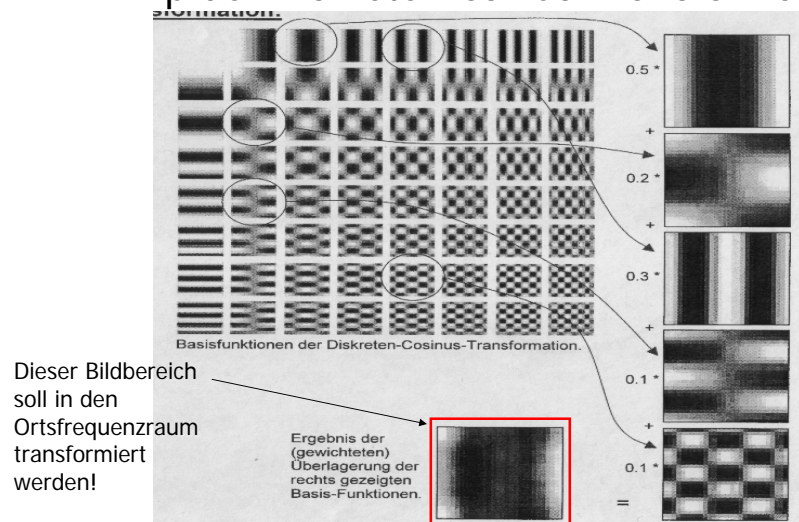


## Diskrete Kosinus-Transformation: Zusammenfassung

1. Bild wird in 8x8 Pixel große Blöcke aufgeteilt
2. Jeder Block wird in 64 vordefinierte Basisfunktionen zerlegt:

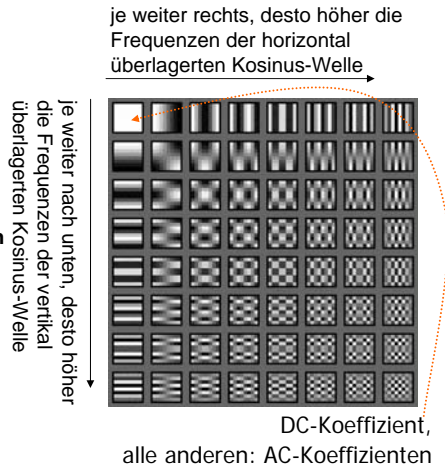


## Prinzip der Diskreten-Kosinus-Transformation

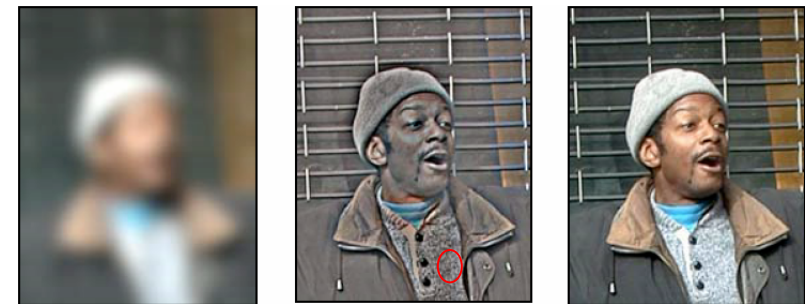


### Diskrete Kosinus-Transformation:

- Bild wird in 8x8 Pixel große Blöcke aufgeteilt
- Jeder Block des Bildes wird in den Ortsfrequenzbereich transformiert:
  - Bildausschnitt wird zerlegt in 64 vordefinierte Basisfunktionen.
  - Bildinformation wird als Überlagerung der Basisfunktionen definiert.
  - **Basisfunktionen sind Basisvektoren eines 64D-Raums.**
- Dies ist für jeden Farbkanal durchzuführen
- **Die Abbildung in den Frequenzraum ist bijektiv (verlustfrei).**



### Zerlegung eines Bildes in seine unterschiedlichen Frequenzen



niedrige Frequenzen      hohe Frequenzen      Überlagerungen

Bilder: Koll, MPEG-Video-Kompression

Relativ uninteressanter Bildinhalt!

### Diskrete-Kosinus-Transformation: „Codierung“

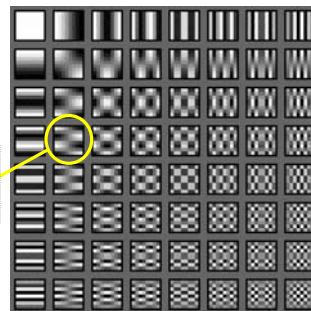
$$F(u,v) = \frac{1}{4} C(u)C(v) \sum_{j=0}^7 \sum_{k=0}^7 f(j,k) \cdot \cos\left(\frac{(2j+1)u\pi}{16}\right) \cdot \cos\left(\frac{(2k+1)v\pi}{16}\right)$$

Koeffizient (Amplitude) der Basisfunktion in der u. Zeile und der v. Spalte

Wert des Pixels in der j. Zeile und der k. Spalte des zu transformierenden Blocks

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } u, v = 0 \\ 1 & \text{sonst} \end{cases}$$

u = 3  
v = 1



Warum genügen diese 64 Basisfunktionen?

### Diskrete Kosinus-Transformation

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

Grauwertverteilung in einem 8x8 großen Block

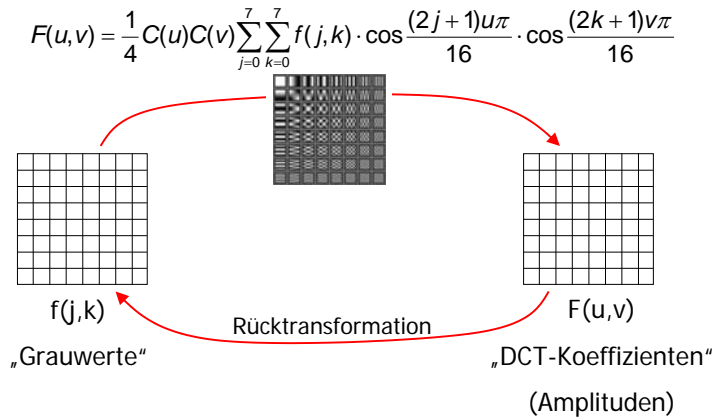
DCT

Mittelwert der Blockhelligkeit

1260	-1	-12	-5	2	-2	-3	1
-23	-17	-6	-3	-3	0	0	-1
-11	-9	-2	2	0	-1	-1	0
-7	-2	0	1	1	0	0	0
-1	-1	1	2	0	-1	1	1
2	0	2	0	-1	1	1	-1
-1	0	0	-1	0	2	1	-1
-3	2	-4	-2	2	1	-1	0

Koeffizientenmatrix

### Diskrete-Kosinus-Transformation



### Diskrete-Kosinus-Transformation: „Decodierung“

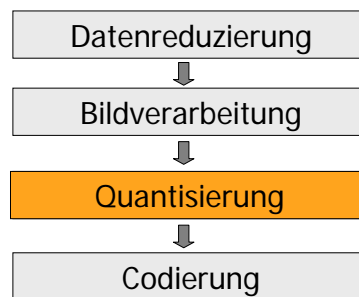
Verlustfreie Rücktransformation:

$$f(i,j) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v) \cdot F(u,v) \cdot \cos\left(\frac{(2j+1)u\pi}{16}\right) \cdot \cos\left(\frac{(2k+1)v\pi}{16}\right)$$

$$C(u), C(v) = \begin{cases} 1 & \text{für } u, v = 0 \\ \frac{1}{\sqrt{2}} & \text{sonst} \end{cases}$$

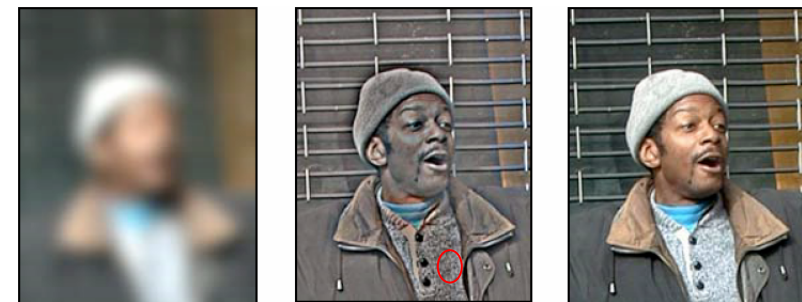
### JPEG

Die Komprimierung der Grafikdaten erfolgt in vier Schritten:



### JPEG

3. Schritt: Quantisierung



niedrige Frequenzen

hohe Frequenzen

Überlagerungen

Durch die Quantisierung wird hochfrequente Bildinformation gelöscht!

Relativ uninteressanter Bildinhalt!

## JPEG

### 3. Schritt: Quantisierung:

- Ganzzahlige Division der Koeffizienten durch vorgegebene Zahl (entweder für alle gleich oder aus Tabelle, die mitübertragen wird)
- Dadurch werden hochfrequente Anteile kontrollierte beseitigt.
- **Löschung von Information!**

## JPEG

1260	-1	-12	-5	2	-2	-3	1
-23	-17	-6	-3	-3	0	0	-1
-11	-9	-2	2	0	-1	-1	0
-7	-2	0	1	1	0	0	0
-1	-1	1	2	0	-1	1	1
2	0	2	0	-1	1	1	-1
-1	0	0	-1	0	2	1	-1
-3	2	-4	-2	2	1	-1	0

Beispiel:  
 $1260 / 16 = 78,75$

79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantisierung erfolgt durch die folgende Quantisierungstabelle:

16	11	10	16	24	40	51	61
14	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

## JPEG

### 3. Schritt: Beispiel für Quantisierungsmatrizen

16	11	10	16	24	40	51	61
14	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Luminanz (Y)

Chrominanz (UV)

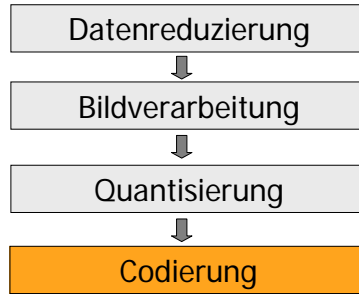
## Quantisierung

Durch die Quantisierung in JPEG werden die hochfrequenten Bildanteile kontrolliert beseitigt.



## JPEG

Die Komprimierung der Grafikdaten erfolgt in vier Schritten:

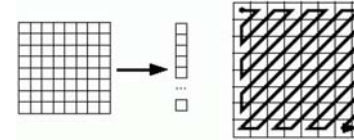


## JPEG

### 4. Schritt: Entropiecodierung

Codierung mit verlustfreier Kompression

- Werte werden in Zickzack Reihenfolge ausgelesen:



Beispiel einer ausgelesenen Koeffizientenmatrix (F(u,v)-Werte):



- durch **Laufängen-Codierung** (je mehr Koeffizienten == 0 sind, desto mehr Platz spart man!) und anschließender Huffman- bzw. Arithmetischer-Codierung komprimiert