

5 JPEG

5.1 Bayer Filter

5.2 Überblick

5.3 Diskrete Cosinus-Transformation

5.4 Bildmodell

5.5 Codierung

5.6 Übertragungsmodi

5.7 Beispiel von Wallace

5.8 JPEG 2000

Bayer Filter:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| G01 | R02 | G03 | R04 | G05 | R06 | G07 | R08 |
| B09 | G10 | B11 | G12 | B13 | G14 | B15 | G16 |
| B17 | R18 | G19 | R20 | G21 | R22 | G23 | G24 |
| B25 | B26 | B27 | G28 | B29 | G30 | B31 | G32 |
| G33 | R34 | G35 | R36 | G37 | R38 | G39 | R40 |
| B41 | G42 | B43 | G44 | B45 | G46 | B47 | G48 |
| G49 | R50 | G51 | R52 | G53 | R54 | G55 | R56 |
| B57 | G58 | B59 | G60 | B61 | G62 | B63 | G64 |

Es existieren viele Möglichkeiten, die nicht direkt gemessenen Werte zu bestimmen, z. B. durch einfache Interpolation.

Beispiele:

Grüne Pixel: $G13 = (G05+G14+G21+G12)/4$

Rote Pixel: $R10 = (R02+R18)/2$

$R13 = (R04+R06+R22+R20)/4$

Blaue Pixel: Berechnung analog zu den roten Pixeln.

Bayer Filter:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| G01 | R02 | G03 | R04 | G05 | R06 | G07 | R08 |
| B09 | G10 | B11 | G12 | B13 | G14 | B15 | G16 |
| B17 | R18 | G19 | R20 | G21 | R22 | G23 | R24 |
| B25 | B26 | B27 | G28 | B29 | G30 | B31 | G32 |
| G33 | R34 | G35 | R36 | G37 | R38 | G39 | R40 |
| B41 | G42 | B43 | G44 | B45 | G46 | B47 | G48 |
| G49 | R50 | G51 | R52 | G53 | R54 | G55 | R56 |
| B57 | G58 | B59 | G60 | B61 | G62 | B63 | G64 |

Berücksichtigung der Farbänderungen in der Nachbarschaft:

Die fehlenden grünen Farbwerte werden über bilineare Interpolation gewonnen.

Beispiele für blaue Ergänzungen:

$$B10 = G10 * (B09/G09 + B11/G11) / 2$$

$$B18 = G18 * (B09/G09+B11/G11+B27/G27+B25/G25) / 4$$

JPEG = Joint Photographic Experts Group
Offiziell: ISO/IEC JTC1 SC29 WG1

Überblick: Komprimierung

Bild

Vorbehandlung

8x8-Blöcke

DCT
(= Discrete Cosine Transform)

Vergrößerung

Entropie-Codierung

8x8-Blöcke

Nachbehandlung

komprimiertes Bild

Bemerkung: Die Vergrößerung, auch Quantisierung genannt, und die Entropie-Codierung werden über Tabellen spezifiziert.

Überblick: Dekomprimierung

komprimiertes Bild

Vorbehandlung

8x8-Blöcke

Entropie-Decodierung

Dequantisierung

IDCT

8x8-Blöcke

Nachbehandlung

Bild

Bemerkung: Die Dequantisierung dient nur dazu, die vor der Quantisierung herrschenden Größenverhältnisse wiederherzustellen.

Neben der verlustbehafteten Bildkompression existiert auch eine verlustfreie JPEG-Variante.

Vorhersage des Pixelwertes X:

| | | | | |
|--|--|---|---|--|
| | | | | |
| | | C | B | |
| | | A | X | |

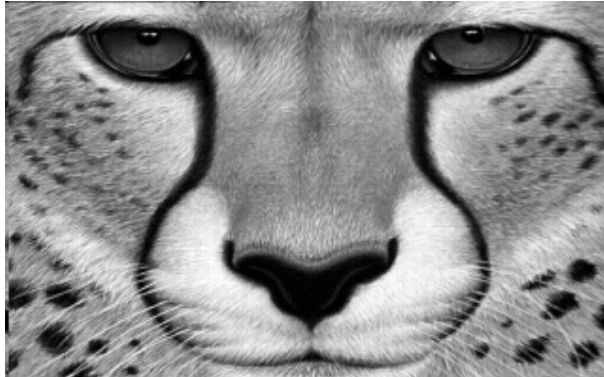
Vorhersageverfahren:

| | |
|---|---------------------|
| 0 | keine Vorhersage |
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | $A + B - C$ |
| 5 | $A + ((B - C) / 2)$ |
| 6 | $B + ((A - C) / 2)$ |
| 7 | $(A + B) / 2$ |

Bemerkung: Die Differenz zum Vorhersagewert sollte klein sein. Damit erhält man eine gute Entropie-Codierung.

Beispiele für verschiedene Qualitäten bei JPEG:

**Bild eines Geparden, Original: 64.000 Byte,
JPEG-Qualität 100, Bildgröße: 50.440 Byte:**



JPEG-Qualität 75, Bildgröße: 16.180 Byte:



JPEG-Qualität 50, Bildgröße = 10.481 Byte:



JPEG-Qualität 25, Bildgröße = 6.526 Byte:



JPEG-Qualität 15, Bildgröße = 4.371 Byte:



JPEG-Qualität 4, Bildgröße = 1.415 Byte:



JPEG-Qualität 5, Bildgröße = 1.733 Byte:



JPEG-Qualität 1, Bildgröße = 933 Byte:



Eindimensionale DCT:

Ein Bild wird durch eine Folge P_x kleiner ganzer Zahlen beschrieben.

Die Elemente G_j für $j = 0, 1, \dots, 7$ der diskreten Cosinus-Transformation sind für 8 Elemente definiert durch:

$$G_j = \frac{1}{2} C_j \sum_{x=0}^7 P_x \cos\left(\frac{(2x+1)j\pi}{16}\right)$$

$$C_j = \frac{1}{\sqrt{2}} \text{ falls } x=0, \text{ sonst } 1.$$

Die Werte P_x für $x = 0, 1, \dots, 7$ gewinnt man zurück über die inverse Transformation:

$$P_x = \frac{1}{2} \sum_{j=0}^7 C_j G_j \cos\left(\frac{(2x+1)j\pi}{16}\right)$$

Anwendung:

$P_x = 11; 13; 15; 17; 19; 21; 23; 25.$

$G_j = 50,91; -12,88; -7,105e-015; -1,347;$
 $1,776e-015; 0,4018; -8,882e-015; -0,1014.$

Eine typische Folge von Pixelwerten für ein JPEG-Bild sei gegeben durch:

211; 213; 214; 216; 215; 213; 210; 210.

Die diskrete Cosinus-Transformation dieser Werte liefert:

601,7; 2,113; -5,385; -0,6451; 0,7071; -0,6801;
-0,06566; -0,8105.

Hierbei ist die überwiegende Zahl der Werte recht klein.

Die Rücktransformation liefert natürlich die Originalwerte:

211; 213; 214; 216; 215; 213; 210; 210.

Fordert man als Ergebnis der Cosinus-Transformation nur Ganzzahlen, dann erhält man:

601; 2; -5; 0; 0; 0; 0; 0.

Die rücktransformierten Zahlen weisen nur geringe Veränderungen gegenüber den Originalzahlen aus:

211; 212; 213; 214; 214; 212; 210; 209.

Ersetzt man auch die 2 durch eine 0, ist das Ergebnis der Rücktransformation auch noch akzeptabel.

210; 211; 213; 214; 214; 213; 211; 210.

Zweidimensionale DCT:

Ein Bild werde beschrieben durch einen Pixelarray P_{xy} .

Die Elemente G_{ij} der diskreten Cosinus-Transformation sind definiert durch

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} P_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

wobei $C_i = \frac{1}{\sqrt{2}}$ falls $x=0$, sonst 1.

Inverse zweidimensionale diskrete Cosinus-Transformation:

Die Rückgewinnung der Pixelelemente erfolgt über

$$P_{xy} = \frac{1}{4} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} C_i C_j G_{ij} \cos\left(\frac{(2x+1)i\pi}{2n}\right) \cos\left(\frac{(2y+1)j\pi}{2n}\right)$$

wobei $C_i = \frac{1}{\sqrt{2}}$ falls $x=0$, sonst 1.

Beispiel zu DCT:

Eingabe:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 136 | 131 | 135 | 139 | 135 | 138 | 139 | 145 |
| 140 | 146 | 132 | 146 | 135 | 133 | 138 | 134 |
| 148 | 145 | 140 | 144 | 148 | 132 | 134 | 149 |
| 149 | 145 | 142 | 132 | 137 | 137 | 139 | 143 |
| 149 | 140 | 132 | 139 | 150 | 146 | 145 | 130 |
| 141 | 134 | 144 | 145 | 131 | 133 | 134 | 149 |
| 132 | 143 | 146 | 146 | 130 | 146 | 144 | 135 |
| 139 | 131 | 143 | 144 | 132 | 134 | 135 | 143 |

Nach DCT (gerundet auf 3 Nachkommastellen):

| | | | | | | | |
|---------|--------|--------|---------|--------|--------|--------|--------|
| 1116,37 | 4,819 | 5,413 | -5,594 | 7,375 | 5,331 | 0,329 | -4,691 |
| -0,559 | 0,938 | 4,416 | 5,724 | 7,082 | -7,197 | -7,994 | 1,463 |
| -10,837 | -7,677 | -4,806 | -12,214 | -2,101 | 6,882 | 2,852 | -7,162 |
| -4,161 | -7,926 | -3,300 | 2,982 | -4,149 | 5,567 | 2,046 | -2,954 |
| -2,875 | -4,480 | 2,809 | 10,921 | 0,125 | 0,633 | 3,842 | 6,797 |
| 2,798 | -3,693 | 3,920 | -6,992 | -8,019 | -2,116 | 6,394 | 0,445 |
| 0,295 | -3,355 | 5,102 | -10,723 | 16,159 | -2,784 | 8,806 | 4,810 |
| 7,141 | -2,595 | -9,081 | 11,326 | -0,516 | -1,629 | -4,707 | 1,962 |

Nach Inverser DCT:

136 131 135 139 135 138 139 145
140 146 132 146 135 133 138 134
148 145 140 144 148 132 134 149
149 145 142 132 137 137 139 143
149 140 132 139 150 146 145 130
141 134 144 145 131 133 134 149
132 143 146 146 130 146 144 135
139 131 143 144 132 134 135 143

Dies ist die Ursprungsmatrix.

Rechenfehler werden durch die Rundung zu Ganzzahlen maskiert.

Ein zweites Beispiel:

Originalmatrix:

136 131 135 139 135 138 139 145
140 146 132 146 135 133 138 134
148 145 140 144 148 132 134 149
149 145 142 132 137 137 139 143
149 140 132 139 150 146 145 130
141 134 144 145 131 133 134 149
132 143 146 146 130 146 144 135
139 131 143 144 132 134 135 143

Nach DCT (gerundet zu Ganzzahlen):

1116 5 5 -6 7 5 0 -5
-1 1 4 6 7 -7 -8 1
-11 -8 -5 -12 -2 7 3 -7
-4 -8 -3 3 -4 6 2 -3
-3 -4 3 11 0 1 4 7
3 -4 4 -7 -8 -2 6 0
0 -3 5 -11 16 -3 9 5
7 -3 -9 11 -1 -2 -5 2

Nach IDCT (gerundet zu Ganzzahlen):

136 131 135 139 135 138 139 145
140 146 132 146 135 133 138 134
147 145 140 144 148 132 134 149
149 145 142 132 137 137 139 143
149 140 132 139 150 146 145 130
141 134 144 145 131 133 134 149
132 143 146 146 130 146 144 135
139 131 143 144 132 134 135 143

Originalmatrix:

```
136 131 135 139 135 138 139 145
140 146 132 146 135 133 138 134
148 145 140 144 148 132 134 149
149 145 142 132 137 137 139 143
149 140 132 139 150 146 145 130
141 134 144 145 131 133 134 149
132 143 146 146 130 146 144 135
139 131 143 144 132 134 135 143
```

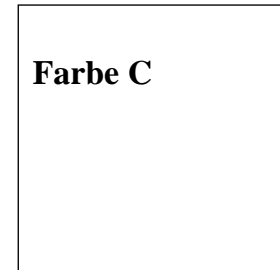
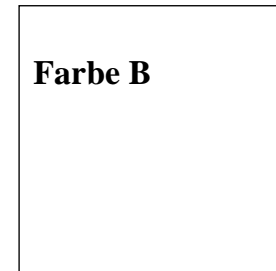
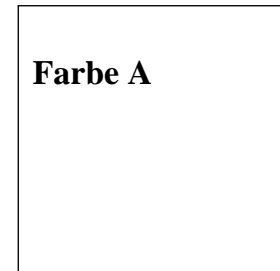
Nach DCT und setzen aller Elemente, deren Betrag kleiner als 11 ist, auf 0:

```
1116 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
-11 0 0 -12 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 11 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 -11 16 0 0 0
0 0 0 11 0 0 0 0
```

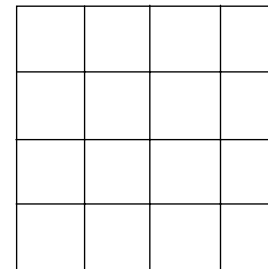
Nach IDCT:

```
138 137 138 140 138 135 136 140
134 142 143 137 135 139 141 138
142 138 139 143 142 137 137 144
143 142 139 139 142 145 143 138
147 141 134 136 145 151 144 133
138 139 143 146 140 132 137 148
137 141 140 136 137 142 142 135
137 137 139 140 137 134 136 141
```

Zum Bildmodell von JPEG, ein Dreikomponentenbild:



Für die einzelnen Komponenten erlaubt der JPEG-Standard ein Größenverhältnis von bis zu 1 : 16.

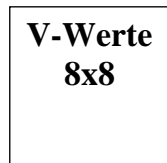
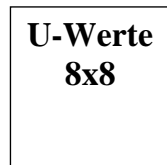
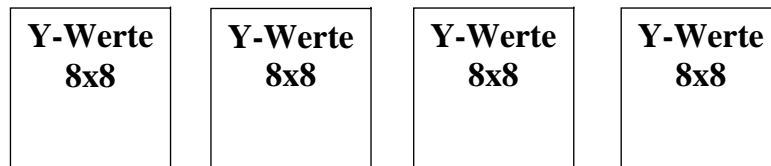


Eine Komponente wird in 8x8-Matrizen aufgeteilt.

Zum Bildmodell von JPEG:

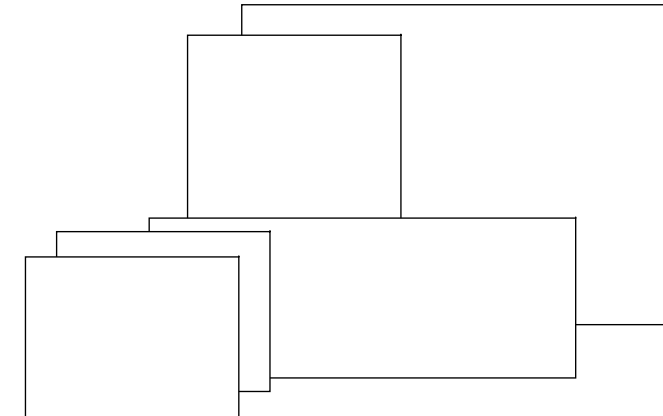
Beim Einsatz von JPEG transformiert man ein Bild häufig in einen luminanzbasierten Farbraum, wie zum Beispiel YUV. Hierbei lassen sich die Datenmengen der U- und V-Komponenten fast ohne Qualitätseinbuße um den Faktor 2 oder 4 reduzieren.

So gehören bei einer Reduktion 4 : 1 bezüglich U und V zu einer 8x8-Umatrix eine 8x8-Vmatrix und vier 8x8-Ymatrizen.



Beschrieben werden muß noch die Zuordnung der Y-, U- und V-Werte.

Bildmodell von JPEG:



Eine Folge von Bildkomponenten.

Ein JPEG-Bild besteht aus 1 bis 255 Bildkomponenten. Jede Komponente ihrerseits ist ein rechteckiger Array von x Zeilen und y Spalten. Die Arraywerte sind aus dem Bereich $[0, 2^P - 1]$ mit $P = 8$ oder $P = 12$ für DCT-basierte Bilder. Alle Komponenten eines Bildes benutzen die gleiche Präzision. Die Dimensionen der einzelnen Komponenten können voneinander abweichen, aber nicht zu stark. Als Verhältniszahlen für die Zeilen- oder Spaltenzahl zweier Komponenten sind die Ganzzahlen 1, 2, 3 und 4 zugelassen.

Ein Beispiel: Ein Dreikomponentenbild mit

$$K_1: x_1 = 512, y_1 = 256,$$

$$K_2: x_2 = 256, y_2 = 512,$$

$$K_3: x_3 = 128, y_3 = 256.$$

Man speichert für ein Bild nur die Maxima der Zeilenzahlen und Spaltenzahlen der Komponenten X und Y. Für das Beispiel sind dies $X = 512$ und $Y = 512$. Für jede Komponente berechnet man horizontale und vertikale Abtastraten. Für das Beispiel ergibt sich:

$$K_1: H_1 = 4, V_1 = 1,$$

$$K_2: H_2 = 2, V_2 = 2,$$

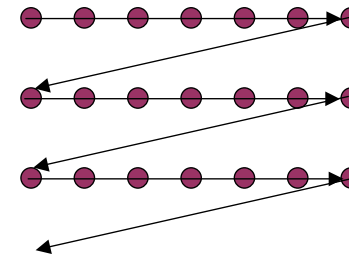
$$K_3: H_3 = 1, V_3 = 1.$$

Man bestimmt nun $H_{\max} = 4$ und $V_{\max} = 2$; zu jeder Komponente werden nur die H- und V-Werte gespeichert. Der Dekompressor berechnet die Zeilenzahl der einzelnen Komponente i nach der Formel $z_i = \lceil X * (H_i / H_{\max}) \rceil$ und die Spaltenzahl analog $s_i = \lceil Y * (V_i / V_{\max}) \rceil$.

Bemerkung: Ein JPEG-Bild enthält keine Informationen über das Seitenverhältnis von Pixeln, den Farbraum oder die Bildentstehung.

Verschränkung der Komponenten:

Die normale Lesart einer Komponente ist von oben nach unten und von links nach rechts wie im Bild demonstriert.



Hat man mehrere Tabellen mit verschiedenen Dimensionen, dann definiert der JPEG-Standard eine kleinste zu verschachtelnde Einheit MCU (= Minimum Coded Unit), sie wird bestimmt durch die Abtastraten H und V.

Betrachten wir ein Beispiel mit den vier Komponenten K_1, K_2, K_3 und K_4 .

Die entsprechenden H- und V-Werte seien

$$H_1 = 2, V_1 = 2,$$

$$H_2 = 2, V_2 = 1,$$

$$H_3 = 1, V_3 = 2,$$

$$H_4 = 1, V_4 = 1.$$

Beispiel einer Huffman-Tabelle zur Codierung der DC-Kategorien:

| Kategorie | Codelänge | Codewort |
|-----------|-----------|-----------|
| 0 | 2 | 00 |
| 1 | 3 | 010 |
| 2 | 3 | 011 |
| 3 | 3 | 100 |
| 4 | 3 | 101 |
| 5 | 3 | 110 |
| 6 | 4 | 1110 |
| 7 | 5 | 11110 |
| 8 | 6 | 111110 |
| 9 | 7 | 1111110 |
| 10 | 8 | 11111110 |
| 11 | 9 | 111111110 |

Die Folge der AC-Koeffizienten wird durch eine Tripel-
folge beschrieben, Länge einer Nullfolge, Bitzahl des
nächsten Nicht-Null-Werts, Wertangabe. Die Wertan-
gabe erfolgt wie bei der DC-Codierung, zur Codierung
des Paares Lauflänge, Bitzahl nutzt man eine Huffman-
tabelle, die für jedes Bild anzugeben ist. Es werden
Lauflängen von 0 bis 15 codiert. Da es Lauflängen größer
als 15 gibt, dient die Angabe (15, 0) als Verlängerungs-
symbol, die Angabe (0, 0) schließt einen Block ab.

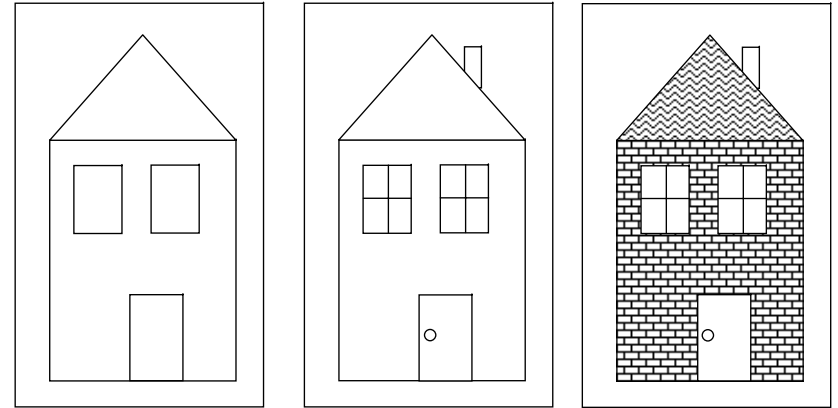
Ausschnitt aus einer Codierungstabelle für AC-Koeffizienten:

| Lauf/Bitzahl | Codelänge | Codewort |
|--------------|-----------|------------------|
| 0/0 (EOB) | 4 | 1010 |
| 0/1 | 2 | 00 |
| 0/2 | 2 | 01 |
| 0/3 | 3 | 100 |
| 0/4 | 4 | 1011 |
| 0/5 | 5 | 11010 |
| 0/6 | 7 | 1111000 |
| 0/7 | 8 | 11111000 |
| 0/8 | 10 | 1111110110 |
| 0/9 | 16 | 111111110000010 |
| 0/A | 16 | 1111111110000011 |
| 1/1 | 4 | 1100 |
| 1/2 | 5 | 11011 |
| 1/3 | 7 | 1111001 |
| 1/4 | 9 | 111110110 |
| 1/5 | 11 | 1111110110 |
| 1/6 | 16 | 111111110000100 |
| 1/7 | 16 | 111111110000101 |
| 1/8 | 16 | 111111110000110 |
| 1/9 | 16 | 111111110000111 |
| 1/A | 16 | 1111111110001000 |
| 2/1 | 5 | 11100 |
| 2/2 | 8 | 11111001 |
| 2/3 | 10 | 1111110111 |
| 2/4 | 12 | 111111110100 |
| 2/5 | 16 | 1111111110001001 |
| 2/6 | 16 | 1111111110001010 |

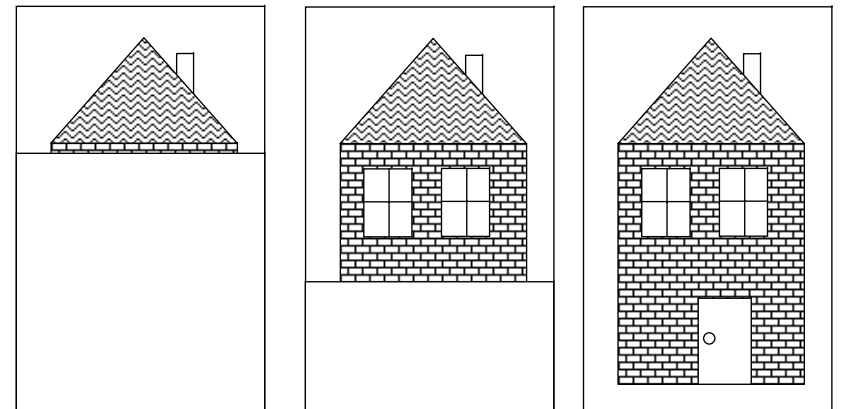
| | | |
|-----|----|-----------------|
| 2/7 | 16 | 111111110001011 |
| 2/8 | 16 | 111111110001100 |
| 2/9 | 16 | 111111110001101 |
| 2/A | 16 | 111111110001110 |
| 3/1 | 6 | 111010 |
| 3/2 | 9 | 111110111 |
| 3/3 | 12 | 11111110101 |
| 3/4 | 16 | 111111110001111 |
| 3/5 | 16 | 111111110010000 |
| 3/6 | 16 | 111111110010001 |
| 3/7 | 16 | 111111110010010 |
| 3/8 | 16 | 111111110010011 |
| 3/9 | 16 | 111111110010100 |
| 3/A | 16 | 111111110010101 |

Übertragungsmodi:

Man unterscheidet drei Übertragungsmodi, den sequentiellen, den progressiven und den hierarchischen.



Progressiver Modus



Sequentieller Modus

Übertragungsmodi:

Hierarchische Zerlegung:

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |

| | |
|--|--|
| | |
| | |

| |
|--|
| |
|--|

Progressive Übertragung:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 8 |
| 3 | 4 | 4 | 5 | 6 | 7 | 8 | 8 |
| 4 | 5 | 5 | 6 | 6 | 7 | 8 | 8 |
| 5 | 6 | 6 | 6 | 7 | 8 | 8 | 8 |
| 6 | 7 | 7 | 7 | 8 | 8 | 8 | 8 |
| 7 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |

Beispiel von Wallace:

Originalmatrix:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 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 |

DC-Transformierte:

| | | | | | | | |
|--------|--------|--------|-------|-------|-------|-------|-------|
| 1260 | -1,03 | -12,08 | -5,20 | 2,13 | -1,67 | -2,71 | 1,32 |
| -22,59 | -17,48 | -6,24 | -3,16 | -2,86 | -0,07 | 0,43 | -1,19 |
| -10,95 | -9,26 | -1,58 | 1,53 | 0,20 | -0,94 | -0,57 | -0,06 |
| -7,08 | -1,91 | 0,23 | 1,45 | 0,90 | -0,08 | -0,04 | 0,33 |
| -0,63 | -0,83 | 1,47 | 1,56 | -0,13 | -0,66 | 0,61 | 1,28 |
| 1,75 | -0,20 | 1,62 | -0,34 | -0,78 | 1,48 | 1,04 | -0,99 |
| -1,28 | -0,36 | -0,32 | -1,46 | -0,49 | 1,74 | 1,08 | -0,76 |
| -2,6 | 1,55 | -3,76 | -1,85 | 1,87 | 1,21 | -0,57 | -0,45 |

Originalmatrix verschoben um -128:

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 11 | 16 | 21 | 25 | 27 | 27 | 27 | 27 |
| 16 | 23 | 25 | 28 | 31 | 28 | 28 | 28 |
| 22 | 27 | 32 | 35 | 30 | 28 | 28 | 28 |
| 31 | 33 | 34 | 32 | 32 | 31 | 31 | 31 |
| 31 | 32 | 33 | 34 | 34 | 27 | 27 | 27 |
| 33 | 33 | 33 | 33 | 32 | 29 | 29 | 29 |
| 34 | 34 | 33 | 35 | 34 | 29 | 29 | 29 |
| 34 | 34 | 33 | 33 | 35 | 30 | 30 | 30 |

DC-Transformierte der angepaßten Originalmatrix:

| | | | | | | | |
|--------|--------|--------|-------|-------|-------|-------|-------|
| 235,6 | -1,03 | -12,08 | -5,20 | 2,15 | -1,67 | -2,71 | 1,32 |
| -22,59 | -17,48 | -6,24 | -3,16 | -2,86 | -0,07 | 0,43 | -1,19 |
| -10,95 | -9,26 | -1,58 | 1,53 | 0,20 | -0,94 | -0,57 | -0,06 |
| -7,08 | -1,91 | 0,22 | 1,45 | 0,90 | -0,08 | -0,04 | 0,33 |
| -0,63 | -0,84 | 1,47 | 1,56 | -0,13 | -0,66 | 0,61 | 1,28 |
| 1,75 | -0,20 | 1,62 | -0,34 | -0,76 | 1,48 | 1,04 | -0,99 |
| -1,28 | -0,36 | -0,32 | -1,46 | -0,49 | 1,74 | 1,08 | -0,76 |
| -2,6 | 1,55 | -3,76 | -1,85 | 1,87 | 1,21 | -0,57 | -0,45 |

Quantisierte DC-Transformierte:

| | | | | | | | |
|----|----|----|---|---|---|---|---|
| 15 | 0 | -1 | 0 | 0 | 0 | 0 | 0 |
| -2 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| -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 |

Dequantisierte DC-Transformierte:

| | | | | | | | |
|-----|-----|-----|---|---|---|---|---|
| 240 | 0 | -10 | 0 | 0 | 0 | 0 | 0 |
| -24 | -12 | 0 | 0 | 0 | 0 | 0 | 0 |
| -14 | -13 | 0 | 0 | 0 | 0 | 0 | 0 |
| -14 | 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 |

IDC-Transformierte:

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 11 | 16 | 21 | 25 | 27 | 27 | 27 | 27 |
| 16 | 23 | 25 | 28 | 31 | 28 | 28 | 28 |
| 22 | 27 | 32 | 35 | 30 | 28 | 28 | 28 |
| 31 | 33 | 34 | 32 | 32 | 31 | 31 | 31 |
| 31 | 32 | 33 | 34 | 34 | 27 | 27 | 27 |
| 33 | 33 | 33 | 33 | 32 | 29 | 29 | 29 |
| 34 | 34 | 33 | 35 | 34 | 29 | 29 | 29 |
| 34 | 34 | 33 | 33 | 35 | 30 | 30 | 30 |

Rekonstruierte Ausgangsmatrix:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 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 |

Quantisierte DC-Transformierte an einer Stelle geändert:

| | | | | | | | |
|----|----|----|---|---|---|---|---|
| 15 | 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 |

Dequantisierte DC-Transformierte:

| | | | | | | | |
|-----|-----|-----|---|---|---|---|---|
| 240 | 0 | -10 | 0 | 0 | 0 | 0 | 0 |
| -24 | -12 | 0 | 0 | 0 | 0 | 0 | 0 |
| -14 | -13 | 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 |

IDC-Transformierte:

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 16 | 18 | 21 | 24 | 26 | 28 | 28 | 28 |
| 20 | 22 | 24 | 26 | 28 | 28 | 28 | 28 |
| 27 | 28 | 29 | 30 | 30 | 30 | 28 | 27 |
| 32 | 33 | 33 | 34 | 33 | 31 | 29 | 27 |
| 35 | 35 | 36 | 35 | 34 | 32 | 29 | 28 |
| 35 | 35 | 36 | 36 | 34 | 32 | 30 | 29 |
| 32 | 33 | 34 | 34 | 34 | 33 | 31 | 30 |
| 30 | 31 | 33 | 33 | 34 | 33 | 31 | 30 |

Rekonstruierte:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 144 | 146 | 149 | 152 | 154 | 156 | 156 | 156 |
| 148 | 150 | 152 | 154 | 156 | 156 | 156 | 156 |
| 155 | 156 | 157 | 158 | 158 | 158 | 156 | 155 |
| 160 | 161 | 161 | 162 | 161 | 159 | 157 | 155 |
| 163 | 163 | 164 | 163 | 162 | 160 | 157 | 156 |
| 163 | 163 | 164 | 164 | 162 | 160 | 158 | 157 |
| 160 | 161 | 162 | 162 | 162 | 161 | 159 | 158 |
| 158 | 159 | 161 | 161 | 162 | 161 | 159 | 158 |

Codierung der letzten quantisierten DC-Transformierten:

Annahme: DC-Koeffizient des Vorgängers ist 12.

Folge: (2)(3), (1, 2)(-2), (0, 1)(-1), (0, 1)(-1),
(0, 1)(-1), (2, 1)(-1), (0, 0)

Bitfolge: 011 11 11011 01 00 0 00 0 00 0 11100 0 1010

Hier: 31 Bit dienen der Codierung von 64
Ausgangswerten.

Codierung der vorletzten quantisierten DC-Transformierten:

Annahme: DC-Koeffizient des Vorgängers ist 12.

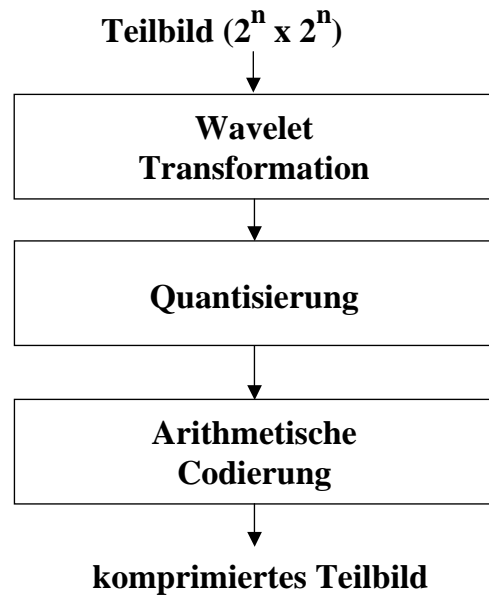
Folge: (2)(3), (1, 2)(-2), (0, 1)(-1), (0, 1)(-1),
(0, 1)(-1), (2, 1)(-1), (0, 1)(-1), (0, 0)

Bitfolge: 011 11 11011 01 00 0 00 0 00 0 11100 0 00 0 1010

Hier: 34 Bit dienen der Codierung von 64
Ausgangswerten.

JPEG 2000:

Überblick über Architektur:



Bemerkungen:

- (i) Bei gleicher Bildqualität komprimiert JPEG 2000 etwa doppelt so gut wie das einfache JPEG.
- (ii) Verlustfreie und verlustbehaftete Komprimierung nutzen die gleiche Architektur.

Ein YCbCr-Modell:

Bildung einer Luminanz $Y = a \cdot R + b \cdot G + c \cdot B$ und neuen Farben Cr, Cb, Cg mit

$$\begin{aligned} Cr &= R - Y, \\ Cb &= B - Y, \\ Cg &= G - Y. \end{aligned}$$

Es gilt $Cr + Cb + Cg = \text{konstant}$, daher genügen zwei der drei Chromakomponenten.

$$\begin{aligned} Y &= 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B, \\ Cb &= 0,564 \cdot (B - Y), \\ Cr &= 0,713 \cdot (R - Y). \end{aligned}$$

Rücktransformation:

$$\begin{aligned} R &= Y + 1,402 \cdot Cr, \\ G &= Y - 0,344 \cdot Cb - 0,714 \cdot Cr, \\ B &= Y + 1,772 \cdot Cb. \end{aligned}$$

Bemerkung: Die Y-Transformation nutzt man auch bei der Wandlung eines Buntbildes in ein Graubild.

Ein zweites YCbCr-Modell:

Man setzt:

$$\begin{aligned} Y &= (77/256)*R + (150/256)*G + (29/256)*B, \\ Cb &= -(44/256)*R - (87/256)*G + (131/256)*B + 128, \\ Cr &= (131/256)*R - (110/256)*G + (21/256)*B + 128. \end{aligned}$$

Bemerkung: Der Wertebereich für Y ist 16 ... 235.

Rücktransformation:

$$\begin{aligned} R &= Y + 1,371 * (Cr - 128), \\ G &= Y - 0,698 * (Cr - 128) - 0,336 *(Cb - 128), \\ B &= Y + 1,772 * Cb. \end{aligned}$$