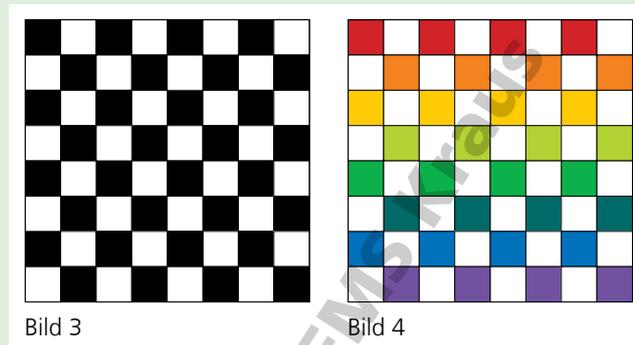
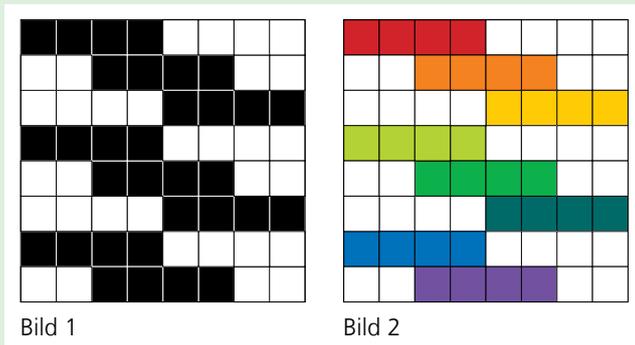


# Themen

	Seite	
Datenkompression und Datenreduktion	A-2	1
Laufängencodierung	A-4	2
Huffman-Codierung	A-8	3
Auflösung und Farbtiefe	A-13	4
Samplingrate und Samplingtiefe	A-16	5
Grafikformate GIF, PNG und JPG	A-19	6
Audioformat MP3	A-26	7

# Laufhängencodierung



Digitale Bilder bestehen aus unzähligen kleinen, einfarbigen Kästchen, die Pixel genannt werden. Sie sind so klein, dass wir sie normalerweise gar nicht wahrnehmen. Eine Bilddatei enthält für jedes dieser Kästchen die entsprechende Farbinformation.

1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0
0	0	0	0	1	1	1	1
1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0
0	0	0	0	1	1	1	1
1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0

Der Inhalt von Bild 1 lässt sich ganz einfach notieren: eine 1 für jedes schwarze und eine 0 für jedes weiße Kästchen. Für jedes der 64 Kästchen wird so ein Bit Speicherplatz benötigt. Insgesamt also 64 Bit bzw. 8 Byte.

Für die farbige Variante in Bild 2 wird analog für jedes Kästchen die Farbe als RGB-Farbe notiert. Eine RGB-Farbe wird mit drei Werten für Rot, Grün und Blau zwischen 0 und 255 angegeben. Für jeden dieser Werte werden 8 Bit, für die RGB-Farbe insgesamt  $3 \times 8 \text{ Bit} = 24 \text{ bit}$  Speicherplatz benötigt. Das gesamte Bild 2 ist also  $64 \times 24 \text{ Bit} = 1536 \text{ Bit}$  bzw. 192 Byte groß.

Wenn wir die Bilder 1 und 2 zeilenweise „lesen“, folgen stets mehrere Pixel einer Farbe aufeinander. Derartige Wiederholungen von Zeichen, Symbolen oder Zahlen finden sich häufig in Dateien. Es bietet sich also an, Folgen identischer Symbole durch deren Anzahl und das Symbol zu ersetzen.

Die Folgen identischer Symbole wurden zunächst englisch als „Run“ bezeichnet. Das auf dieser Grundidee basierende Datenkompressionsverfahren erhielt den englischen Namen run-length encoding, oder kurz RLE. Davon wurden später die deutschen Bezeichnungen „Lauf“ und Laufhängencodierung abgeleitet.

Das Verfahren wird technisch als Vorcodierungsschritt bei der Bildkompression (z.B. JPEG) eingesetzt.

Wenden wir die Laufhängencodierung auf unser Beispielbild 1 an, erhalten wir

4 x 1   6 x 0   4 x 1   6 x 0   4 x 1  
 4 x 1   6 x 0   4 x 1   6 x 0   4 x 1  
 4 x 1   6 x 0   4 x 1   2 x 0

bzw. in Binärschreibweise

100 1   110 0   100 1   110 0   100 1  
 100 1   110 0   100 1   110 0   100 1  
 100 1   110 0   100 1   010 0

Für die Codierung jedes einzelnen der 14 Läufe werden 3 Bit + 1 Bit benötigt. Das gesamte Bild benötigt folglich  $14 \times 4 \text{ Bit} = 56 \text{ Bit}$  bzw. 7 Byte Speicherplatz.

Wie stark ein Bild im Vergleich zur Originalgröße verkleinert wurde, gibt die Kompressionsrate an. Sie wird mit dieser Formel berechnet:

$$\text{Kompressionsrate} = 100 \times \left(1 - \frac{\text{komprimiert}}{\text{Originalgröße}}\right) \%$$

$$\text{Kompressionsrate} = 100 \times \left(1 - \frac{7 \text{ Byte}}{8 \text{ Byte}}\right) = 12,5 \%$$

Für die Codierung der 14 Läufe im farbigen Bild 2 werden jeweils 3 Bit für die Kästchenzahl und 24 Bit für die Farbe benötigt. Insgesamt ergibt das  $14 \times 27 \text{ Bit} = 378 \text{ Bit}$  bzw. 47,25 Byte. Daraus ergibt sich eine Kompressionsrate von gut 75 %. Diese hohe Kompressionsrate resultiert vor allem daraus, dass pro Lauf die Farbe nur einmal codiert werden muss. Dadurch werden jeweils 9 bzw. 15 Byte Speicherplatz eingespart.

Die Laufhängencodierung führt jedoch nicht in jedem Fall zu einem geringeren Speicherbedarf. In Dateien, die nur wenige Wiederholungen enthalten, erhält man nur eine sehr geringe Kompressionsrate (z. B. 3,5 % bei Bild 4).

In ungünstigen Fällen wird die Datei durch die Laufhängencodierung sogar deutlich größer; im Bild 3 beispielsweise mehr als doppelt so groß.

# Laufängencodierung

## Aufgabe 1

Beschreibe, welche Grundidee der Laufängencodierung zugrunde liegt und wie man bei der Codierung vorgeht.

Bei der Laufängencodierung wird ausgenutzt, dass in Dateien häufig mehrere identische Pixel, Zeichen, Symbole oder Zahlen aufeinanderfolgen.

Bei der Codierung werden diese Folgen identischer Symbole durch deren Anzahl und das Symbol ersetzt.

## Aufgabe 2

Für welche Art Bilder liefert das Verfahren der Laufängencodierung hohe Kompressionsraten?

Die Laufängencodierung liefert bei farbigen Bildern mit wenigen Farbwechseln die größten Kompressionsraten.

## Aufgabe 3

Für welche Art Bilder sind die mit der Laufängencodierung komprimierten Bilder größer als das Original?

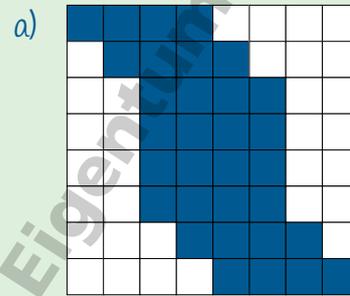
Schwarz-weiß-Bilder mit vielen Wechseln zwischen schwarz und weiß und farbige Bilder mit wenigen aufeinanderfolgenden identisch gefärbten Pixeln sind mit der Laufängencodierung meist größer als das Original.

## Aufgabe 4

Das sind die Daten eines acht Pixel breiten Bildes:

```
11110000011110000011110000111100
00111100001111000001111000001111
```

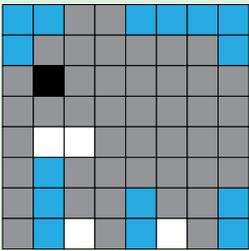
- Zeichne das Bild.
- Berechne die Datenmenge der Bilddaten.
- Codiere die Bilddaten mit der Laufängencodierung.
- Berechne die Datenmenge der komprimierten Bilddaten.
- Wie groß ist die Kompressionsrate?



- $64 \times 1 \text{ Bit} = 64 \text{ Bit}$
- $4 \times 1, 5 \times 0, 4 \times 1, 5 \times 0,$   
 $4 \times 1, 4 \times 0, 4 \times 1, 4 \times 0,$   
 $4 \times 1, 4 \times 0, 4 \times 1, 5 \times 0,$   
 $4 \times 1, 5 \times 0, 4 \times 1$
- 15 Läufe, maximal 5 lang, 3 Bit, Farbe 1 Bit  
 $15 \times (3 + 1) \text{ Bit} = 60 \text{ Bit}$
- Kompressionsrate 6,25 %

# Laufhängencodierung

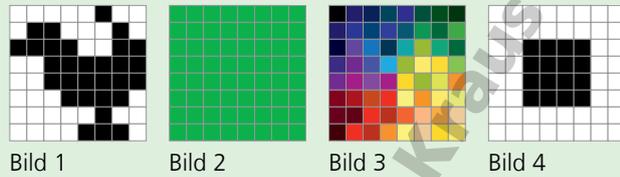
## Aufgabe 5



- Berechne die Datenmenge des Bildes.
- Wie groß ist die Datenmenge, nachdem das Bild mit der Laufhängencodierung komprimiert wurde?  
Beachte, wie viel Bit du für die Speicherung der maximalen Kästenanzahl der Läufe benötigst.
- Berechne die erzielte Kompressionsrate.

- 64 Kästchen  $\times$  24 Bit = 1536 Bit
- 2 x blau, 2 x grau, 5 x blau, 6 x grau,  
1 x blau, 1 x grau, 1 x schwarz, 15 x grau,  
2 x weiß, 6 x grau, 1 x blau, 7 x grau,  
1 x blau, 2 x grau, 1 x blau, 2 x grau,  
1 x blau, 1 x grau, 1 x blau, 1 x weiß,  
1 x grau, 1 x blau, 1 x weiß, 1 x grau, 1 x blau  
  
25 Läufe, maximal 15 lang, 4 Bit,  
Farbe 24 Bit  
 $25 \times (4 + 24) \text{ Bit} = 700 \text{ Bit}$
- Kompressionsrate: 54,4 %

## Aufgabe 6



- Schätze ein, ob die Laufhängencodierung bei den vier Bildern zu einer Verringerung der Datenmenge führen wird?
- Berechne die Datenmenge der vier Bilder.
- Wie groß ist die Datenmenge der vier Bilder, nachdem sie mit der Laufhängencodierung komprimiert wurden?
- Berechne die erzielten Kompressionsraten.  
War deine Einschätzung aus Aufgabe a) richtig?

### Bild 1

- $64 \times 1 \text{ Bit} = 64 \text{ Bit}$

- ```

00000100
01100010
11110011
00111111
00111111
00011110
00000100
00001110

```

5 x 0, 1 x 1, 3 x 0, 2 x 1, 3 x 0, 1 x 1, 1 x 0,  
4 x 1, 2 x 0, 2 x 1, 2 x 0, 6 x 1, 2 x 0, 6 x 1,  
3 x 0, 4 x 1, 6 x 0, 1 x 1, 6 x 0, 3 x 1, 1 x 0

21 Läufe, maximal 6 lang, 3 Bit, Farbe 1 Bit  
 $21 \times (1 + 3) \text{ Bit} = 84 \text{ Bit}$

- das „komprimierte“ Bild ist größer

# Laufhängencodierung

## Bild 2

---

- b)  $64 \text{ Kästchen} \times 24 \text{ Bit} = 1536 \text{ Bit}$
- c) 1 Lauf, 64 lang, 7 Bit, Farbe 24 Bit  
 $1 \times (7 + 24) \text{ Bit} = 31 \text{ Bit}$
- d) Kompressionsrate: 98 %

## Bild 3

---

- b)  $64 \text{ Kästchen} \times 24 \text{ Bit} = 1536 \text{ Bit}$
- c) 64 Läufe, 1 lang, 1 Bit, Farbe 24 Bit  
 $64 \times (1 + 24 \text{ Bit}) = 1600 \text{ Bit}$
- d) das „komprimierte“ Bild ist größer

## Bild 4

---

- b)  $64 \times 1 \text{ Bit} = 64 \text{ Bit}$
- c) 00000000  
 00000000  
 00111100  
 00111100  
 00111100  
 00111100  
 00000000  
 00000000
- 18 x 0, 4 x 1, 4 x 0, 4 x 1,  
 4 x 0, 4 x 1, 4 x 0, 4 x 1, 18 x 0
- 9 Läufe, maximal 18 lang, 6 Bit, Farbe 1 Bit  
 $9 \times (6 + 1) \text{ Bit} = 63 \text{ Bit}$
- d) Kompressionsrate: 1,56 %

# Auflösung und Farbtiefe



**Bild 1**  
Auflösung 300 ppi / 300 dpi  
Farbtiefe 16 Bit  
Dateigröße 1,39 Megabyte



**Bild 2**  
Auflösung 300 ppi / 300 dpi  
Farbtiefe 8 Bit  
Dateigröße 731 Kibibyte



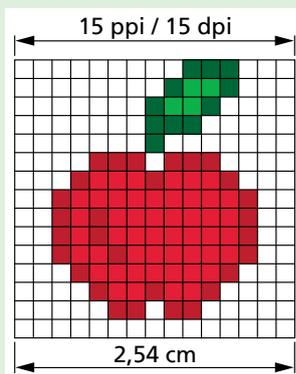
**Bild 3**  
Auflösung 72 ppi / 72 dpi  
Farbtiefe 16 Bit  
Dateigröße 106 Kibibyte



**Bild 4**  
Auflösung 72 ppi / 72 dpi  
Farbtiefe 8 Bit  
Dateigröße 65,5 Kibibyte

## Auflösung

Digitale Bilder bestehen aus kleinen, einfarbigen Kästchen, den Pixeln. Für die Qualität eines Bildes ist es unter anderem wichtig, wie dicht die Pixel angeordnet sind. Befinden sich viele Pixel auf engem Raum, spricht man von einer hohen Auflösung.



Die Auflösung eines Bildes wird in dpi oder ppi angegeben. dpi wird für Druckdaten verwendet und bedeutet Dots per Inch, also Punkte pro Inch. Inch ist eine Maßeinheit des angloamerikanischen Maßsystems und entspricht 2,54 cm.

Die in den Bildern 1 und 2 verwendete Auflösung von 300 dpi entspricht der üblichen Bildauflösung von Bildern für den Druck.

Bei Bildern für digitale Medien wird die Auflösung eher in ppi angegeben. Das bedeutet Pixel pro Inch.

## Farbtiefe

Die drei Farbkanäle Rot, Grün und Blau in farbigen Bildern nach dem RGB-Farbschema können in gewöhnlichen Bildern Werte zwischen 0 und 255 annehmen. Damit stehen pro Farbkanal  $2^8$  Abstufungen zur Verfügung. Daraus ergeben sich  $(2^8)^3$ , also über 16 Millionen unterschiedliche Farben.

Für die Anzahl der verfügbaren Farben für ein Pixel wird der Begriff Farbtiefe verwendet.

Genau genommen bezeichnet der Begriff aber nicht die Anzahl der Farben, sondern den Speicherplatz, der nötig ist, um die Farbinformation dieser Farben zu speichern. Für  $2^8$  Abstufungen pro Farbkanal sind das 8 Bit und damit beträgt auch die Farbtiefe 8 Bit.

## Dateigröße

Die Größe einer Bilddatei entspricht der Summe des Speicherbedarfs der Farbinformationen aller Pixel und ein paar weiteren Informationen zu Format, Abmessung und Name der Datei. Die Dateigröße lässt sich daher mit dieser Formel näherungsweise berechnen:

Dateigröße  $\approx$  Pixelanzahl  $\times$  Farbtiefe  $\times$  Farbkanäle  
Datei Apfel  $\approx$  225 Pixel  $\times$  8 Bit  $\times$  3  $\approx$  5400 Bit

Da aus einer höheren Auflösung meist eine größere Pixelanzahl resultiert, haben sowohl die Auflösung als auch die Farbtiefe einen direkten Einfluss auf die Größe einer Bilddatei. Anhand der Bilder oben ist leicht zu sehen, dass eine Verdoppelung der Farbtiefe die Dateigröße annähernd verdoppelt. Im Gegenzug bewirkt eine geringere Auflösung – also weniger Pixel bei gleicher Bildbreite – eine drastische Reduktion der Dateigröße.

Möchte man die Größe eines Bildes reduzieren, beispielsweise, um es für eine Website zu verwenden, kann man eine Farbtiefe von 8 Bit anstelle von beispielsweise 16 Bit verwenden und die Auflösung verringern. Während die geringere Farbtiefe dabei vermutlich kaum auffällt, verändert die verlustbehaftete Verringerung der Pixelanzahl bei gleichbleibender Bildabmessung das Aussehen des Bildes möglicherweise deutlich.

Foto Burg Eltz: Walter Kärcher (Pixabay)

# Auflösung und Farbtiefe

## Aufgabe 1

Ein Bild ist 22 × 22 cm groß und enthält 1 299 × 1 299 Pixel.

Wie groß ist die Auflösung?

Umrechnung Seitenlänge in die Einheit Inch:  
 $22 \text{ cm} : 2,54 = 8,66 \text{ Inch}$

Pixelanzahl durch Seitenlänge in Inch teilen:  
 $1299 \text{ Pixel} : 8,66 \text{ Inch} = 150 \text{ ppi}$

Die Auflösung beträgt 150 ppi bzw. 150 dpi.

## Aufgabe 2

Wie viele Farben stehen bei einem RGB-Bild für jedes einzelne Pixel zur Verfügung?

- a) bei 8 Bit Farbtiefe
- b) bei 16 Bit Farbtiefe
- c) bei 4 Bit Farbtiefe

Farbtiefe    Farben

- a)     8 Bit     $(2^8)^3 = 16\,777\,216$
- b)     16 Bit    $(2^{16})^3 = 281\,474\,976\,710\,656$
- c)     4 Bit     $(2^4)^3 = 4\,096$

## Aufgabe 3

Berechne die ungefähre Dateigröße für ein RGB-Bild mit 2 500 × 2 000 Pixeln und 16 Bit Farbtiefe.

Dateigröße =  $2500 \times 2000 \times 16 \text{ Bit} \times 3$

Dateigröße = 240 000 000 Bit  
 = 30 000 000 Byte  
 = 29 296,875 Kibibyte  
 = 28,61 Mebibyte

## Aufgabe 4

Auf wie viel Prozent kann die Größe einer RGB-Bilddatei ungefähr reduziert werden, wenn sie in ein Graustufenbild umgewandelt wird?

Hinweis: Überleg zuerst, wie viele Farbkanäle ein Graustufenbild hat.

Die Dateigröße wird mit dieser Formel berechnet:

Dateigröße = Pixelanzahl × Farbtiefe × Anzahl Farbkanäle

Graustufenbilder haben nur einen Farbkanal für Schwarz mit der entsprechenden Farbtiefe des Bildes, z.B. 8 Bit, also 256 Abstufungen von schwarz bis weiß.

Dateigröße RGB-Farben = Pixelanzahl × Farbtiefe × 3

Dateigröße Graustufen = Pixelanzahl × Farbtiefe × 1

Die Dateigröße eines RGB-Bildes kann durch Umwandeln in ein Graustufenbild auf etwa ein Drittel reduziert werden.

# Auflösung und Farbtiefe

## Aufgabe 5

Eine Digitalkamera hat einen Bildsensor mit 20 Megapixeln und arbeitet mit 8 Bit Farbtiefe und Standard-RGB-Farben.

Wieviel Speicherplatz benötigt jedes Bild ungefähr, das mit der Kamera aufgenommen wird?

$$20 \text{ Megapixel} = 20 \text{ Millionen Pixel}$$

$$\begin{aligned} \text{Speicherplatz} &= 20\,000\,000 \times 8 \times 3 \\ &= 480\,000\,000 \text{ Bit} \\ &= 60\,000\,000 \text{ Byte} \\ &= 60\,000 \text{ Kilobyte (58\,593,75 Kibibyte)} \\ &= 60 \text{ Megabyte (57,22 Mebibyte)} \end{aligned}$$

## Aufgabe 6

Du hast ein RGB-Bild mit 8 Bit Farbtiefe vorliegen. Es ist  $5\,000 \times 3\,300$  Pixel groß.

Du möchtest es mit einer Breite von 20 cm und einer Auflösung von 300 dpi drucken.

Die Bilddatei soll dafür so klein wie möglich sein.

Um wie viel Prozent kannst du die Dateigröße vor dem Druck reduzieren?

Größe Ausgangsdatei:

$$5\,000 \times 3\,300 \times 8 \times 3 = 396\,000\,000 \text{ Bit}$$

Endmaß in Inch:

$$\text{Breite } 20 \text{ cm} = 7,874 \text{ Inch}$$

$$\text{Breite } 7,874 \text{ Inch} \times 300 \text{ dpi} = 2362 \text{ Pixel}$$

Berechnung der Bildhöhe:

$$\frac{2\,362 \text{ Pixel}}{5\,000 \text{ Pixel}} = \frac{\text{Bildhöhe } x}{3\,300 \text{ Pixel}}$$

$$\text{Bildhöhe } x = \frac{2\,362 \times 3\,300}{5\,000} = 1559 \text{ Pixel}$$

$$2\,362 \times 1559 \times 8 \times 3 = 88\,376\,592 \text{ Bit}$$

$$100 \times \left( 1 - \frac{88\,376\,592 \text{ Bit}}{396\,000\,000 \text{ Bit}} \right) = 100 \times (1 - 0,22317321) = 77,7 \%$$