

Übung zur Vorlesung

Digitale Medien

Vorlesung: Heinrich Hußmann

Übung: Renate Häuslschmid

Ludwig-Maximilians-Universität München

Wintersemester 2016/17

Bildgröße und Auflösung

Beispiel:

Bild mit 2560 x 1920 Pixel Größe.

$$\text{Breite [px]} = \text{Breite [in]} * \text{Auflösung [ppi]}$$

$$1 \text{ Inch (Zoll)} = 2,54\text{cm}$$

Anzeige auf dem Bildschirm mit 72 ppi:

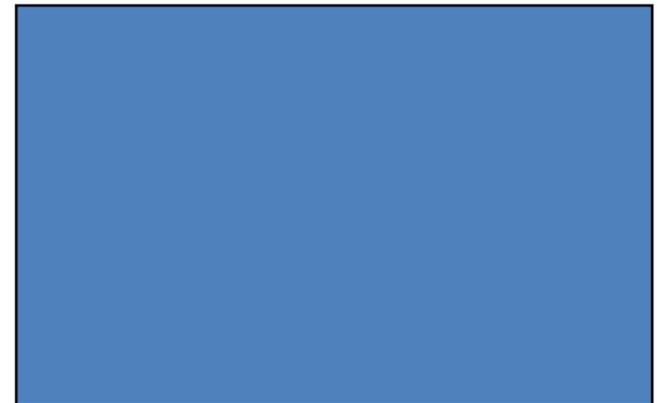
$$\text{Breite[px]} = \text{Breite[in]} * \text{Auflösung[ppi]}$$

$$2560 = \text{Breite[in]} * 72 \text{ ppi}$$

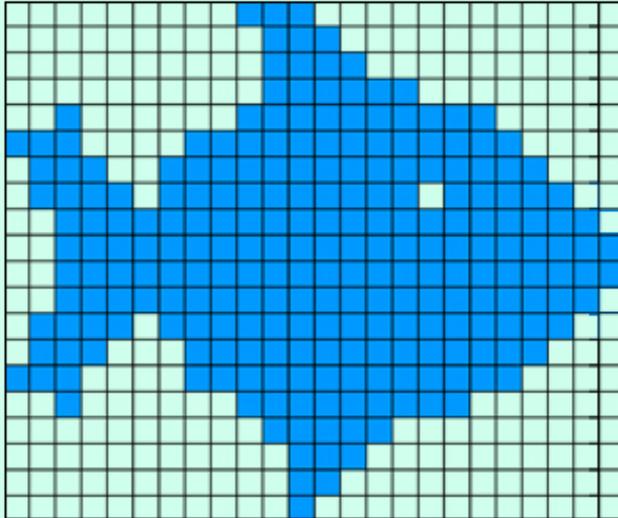
$$\text{Breite[in]} = 2560 / 72 = 35,55 \text{ inch}$$

$$\text{Breite[cm]} = \text{Breite[in]} * 2,54$$

$$\text{Breite[cm]} = 35,55 * 2,54 = 90,31\text{cm}$$



Bildgröße und Auflösung



} 1 Zoll = 2,54 cm (die Darstellung am Monitor kann abweichen)

Dieses Bild enthält 20 x 24 Pixel = 480 Pixel.
Das ist die absolute Auflösung.

Dieses Bild enthält 4 Pixel pro Zoll (Zoll = inch).
Damit beträgt die relative Auflösung dieses Bildes = 4 ppi.

Die Angabe „4 ppi“ ist für sich alleine ohne Wert,
denn sie sagt nichts über die vorhandene Pixelmenge aus.
Erst durch die Angabe von Länge und Breite ist der Wert vollständig.

Im Beispiel ist das Bild 5 Zoll hoch und 6 Zoll breit.
Die vollständige Angabe der relativen Auflösung muss also lauten:
5 x 6 Zoll mit 4 ppi.

Quelle: wikipedia.org

Bildgröße und Auflösung

Beispiel:

Bild mit **2560 x 1920** Pixel Größe.

$$\text{Breite [px]} = \text{Breite [in]} * \text{Auflösung [ppi]}$$

1 Inch (Zoll) = 2,54cm

Anzeige auf 19" Bildschirm, 40cm Bildbreite.

Wieviel ppi werden benötigt, um das Bild bildschirmfüllend anzuzeigen?

$$\text{Breite[in]} = \text{Breite[cm]} / 2,54$$

$$\text{Breite[in]} = 40\text{cm} / 2,54 = 15,75 \text{ in}$$

$$\text{Breite[px]} = \text{Breite[in]} * \text{Auflösung[ppi]}$$

$$2560 = 15,75 * \text{Auflösung[ppi]}$$

$$\text{Auflösung[ppi]} = 2560 / 15,75 = 162,54 \text{ ppi}$$

PPI oder DPI

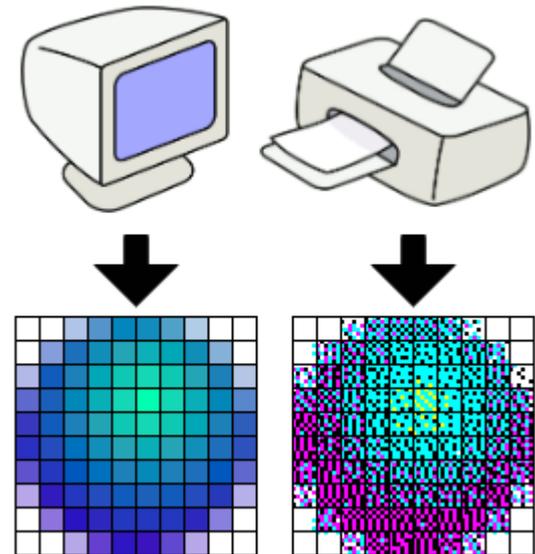
ppi = Wie viele Pixel pro Zoll (Inch) werden angezeigt?

dpi = Wie viele Punkte (dots) werden pro Zoll angezeigt?

Je nach Ausgabemedium werden oft mehrere dots zur Darstellung eines Pixels verwendet.

Beispiele: Drucker (z.B. 4 Farbpunkte pro Pixel), Monitor (z.B. 3 Bildpunkte pro Pixel).

Ob man DPI oder PPI verwendet hängt also davon ab, was man sagen möchte.



Quelle: wikipedia.org

Aufgabe 1

- a) Eine Bilddatei mit einer Auflösung von 640x480 Pixeln sei vorgegeben. Berechnen Sie den benötigten Speicherplatz in Kilobytes (1 KByte = 1.000 Byte), falls die Datei mit folgenden Werten unkomprimiert und ohne Farbpalette gespeichert wird:
- a. Farbauflösung 8 Bit pro Farbe (RGB), 72 ppi
 - b. Farbauflösung 4 Bit pro Farbe (RGB), 72 ppi
 - c. Farbauflösung 2 Bit pro Farbe (RGB), 72 ppi
 - d. Farbauflösung 2 Bit pro Farbe (RGB), 300 ppi

Welche Auswirkung haben ppi auf die Dateigröße? Erklären Sie warum das so ist.

- b) Sie wollen die Datei nun in der Größe 12,7 x 12,7cm drucken. Wie viele ppi können maximal ohne Skalierung des Bildes erreicht werden (Anmerkung: 1 Zoll (Inch) = 2,54 cm)? Erläutern Sie wie Sie auf diesen Wert kommen.

Lösung zu Aufgabe 1 a)

- a. 24 Bit, 72 ppi: $640 * 480 * 24 = 7.372.800 \text{ Bit} = 921.600 \text{ Bytes} = 921,6 \text{ KBytes}$
- b. 12 Bit, 72 ppi: $640 * 480 * 12 = 3.686.400 \text{ Bit} = 460.800 \text{ Bytes} = 460,8 \text{ KBytes}$
- c. 6 Bit, 72 ppi: $640 * 480 * 6 = 1.843.200 \text{ Bit} = 230.400 \text{ Bytes} = 230,4 \text{ KBytes}$
- d. 6 Bit, 300 ppi: 230,4 Kbytes (Dateigröße unabhängig von ppi)

Die ppi Anzahl hat keine Auswirkung auf die Dateigröße. Sie gibt lediglich an, wie dicht die Pixel gesetzt werden und verändert aber nicht die Pixelanzahl.

Lösung zu Aufgabe 1 b)

$$12.7 \text{ cm} = 12.7 / 2,54 \text{ in} = 5 \text{ in}$$

$$\text{Breite[px]} = \text{Breite[in]} * \text{Auflösung[ppi]}$$

Breite: $640 \text{ px} = 5 \text{ in} * \text{Auflösung}$

$$\text{Auflösung} = 640 \text{ px} / 5 \text{ in} = 128 \text{ ppi}$$

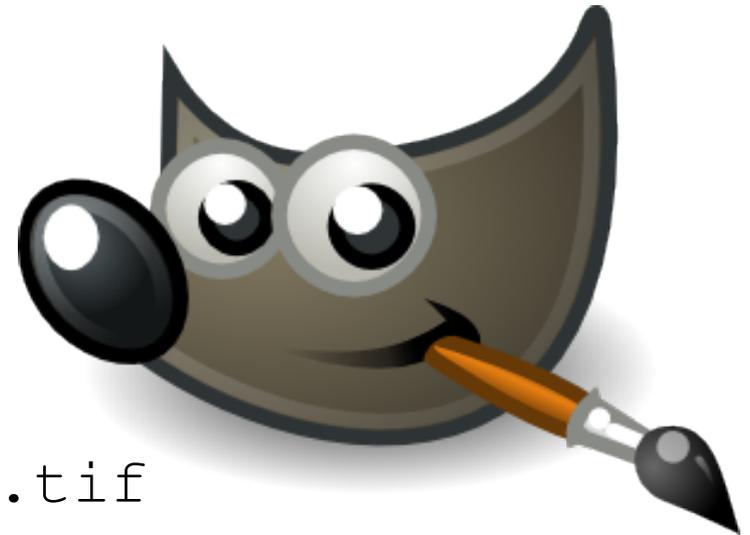
Höhe: $480 \text{ px} = 5 \text{ in} * \text{Auflösung}$

$$\text{Auflösung} = 480 \text{ px} / 5 \text{ in} = 96 \text{ ppi}$$

→ Maximal 128 ppi in der Breite und 96 ppi in der Höhe.

Ohne Skalierung werden 140 px in der Breite abgeschnitten.

GIMP



Gimp starten

Beispielbild:

`/home/proj/mi_dm/img/lena.tif`

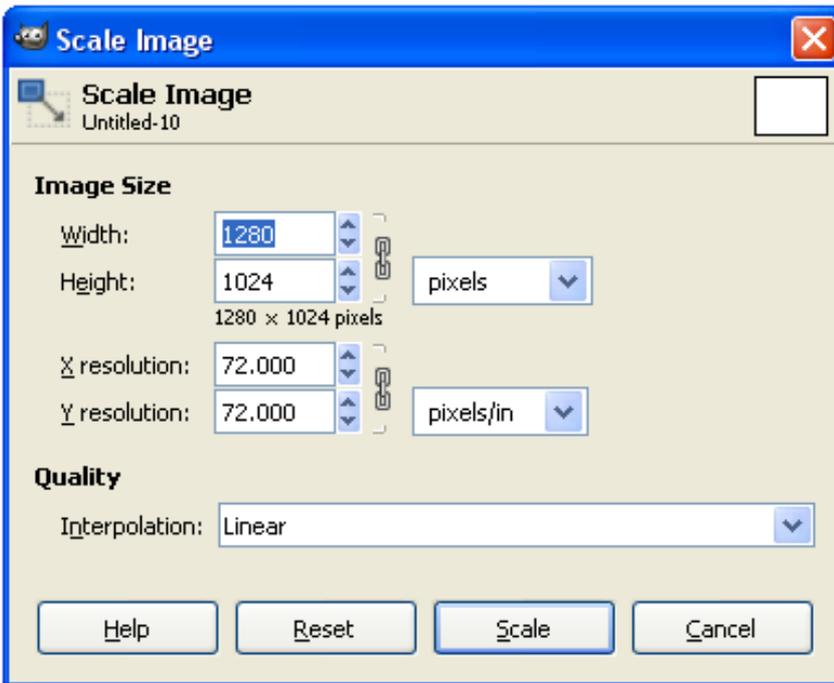
oder

`uebung8.zip` auf der Webseite

GIMP

Wichtige Funktionen in GIMP (Version 2.8.6):

Skalierung: Image → Scale Image



Auflösung in Bildpunkten

Anzeigaauflösung (z.B. in ppi)

Interpolationseinstellung

GIMP

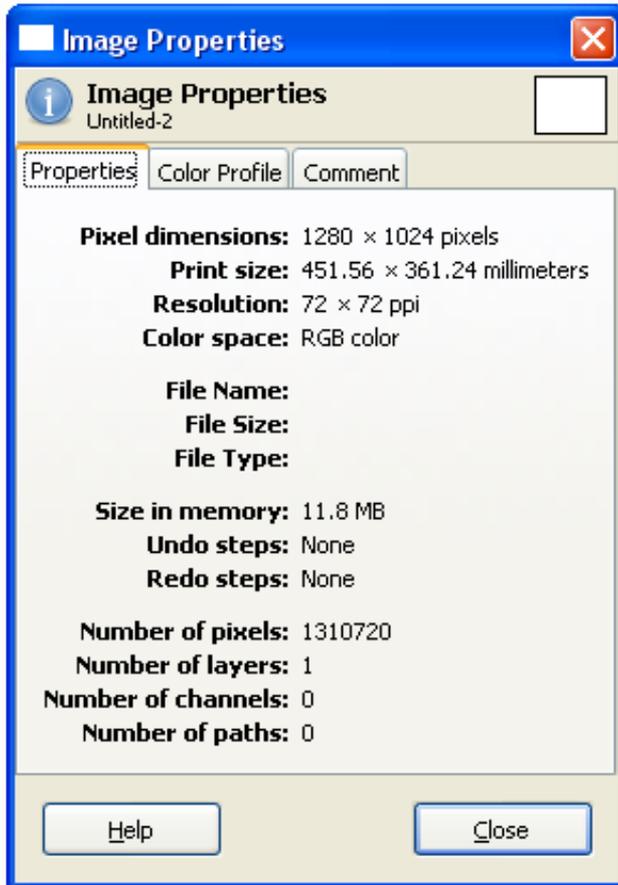
Wichtige Funktionen in GIMP:

Info-Fenster:

Image → Image Properties

Informationen über das Bild:

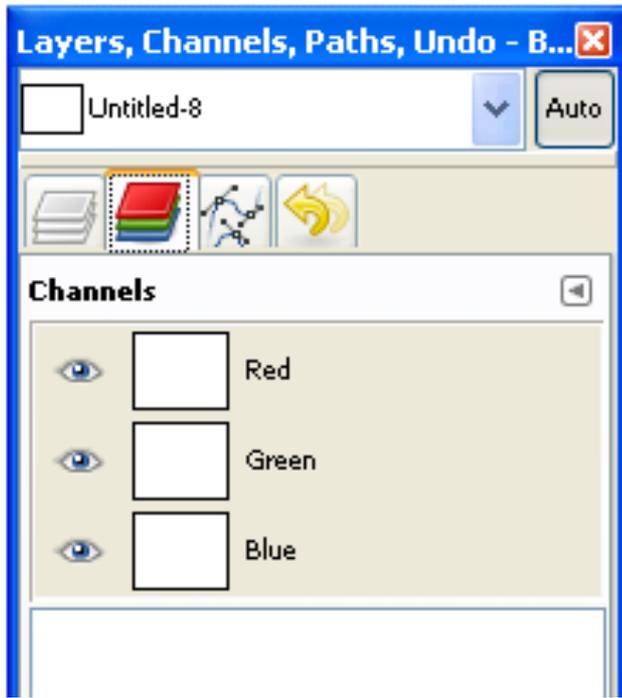
Größe und Auflösung, Farbtiefe



GIMP

Wichtige Funktionen in GIMP:

Farbkanäle: Windows → Dockable Dialogs → Channels



Einzelne Farbkanäle des Bildes

Zu-/Abschaltbar per Auge-Icon

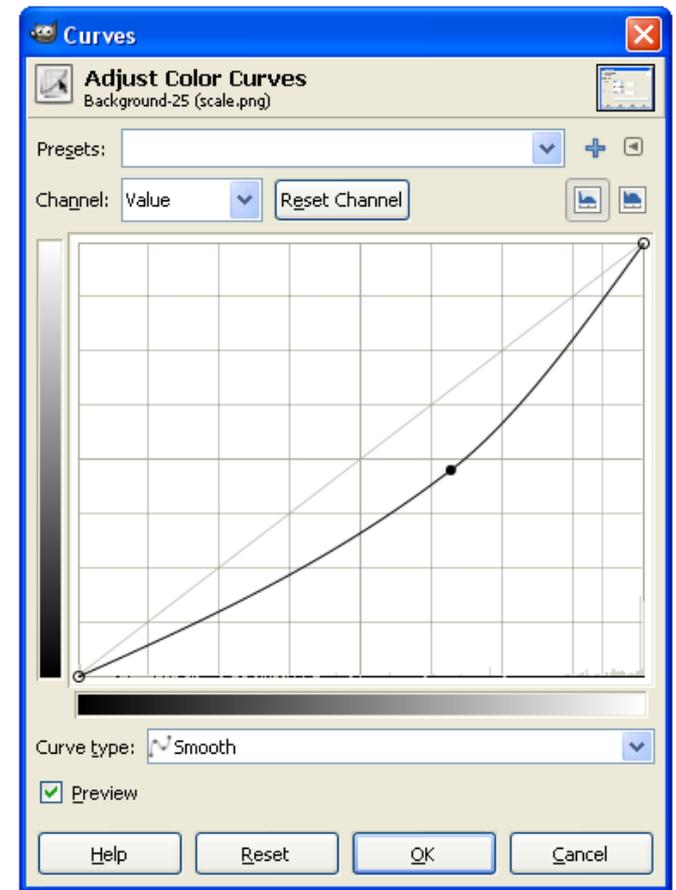
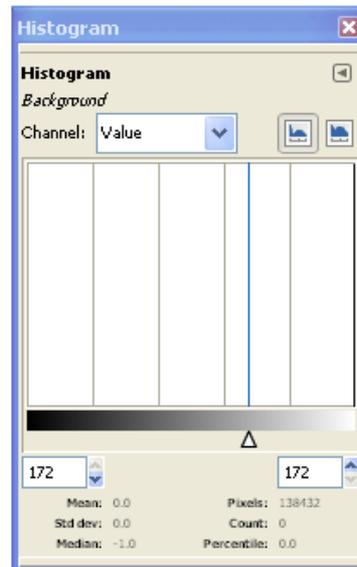
GIMP

Wichtige Funktionen in GIMP:

Farbhistogramm: Colors → Info → Histogram

Übersicht über die Farbverteilung

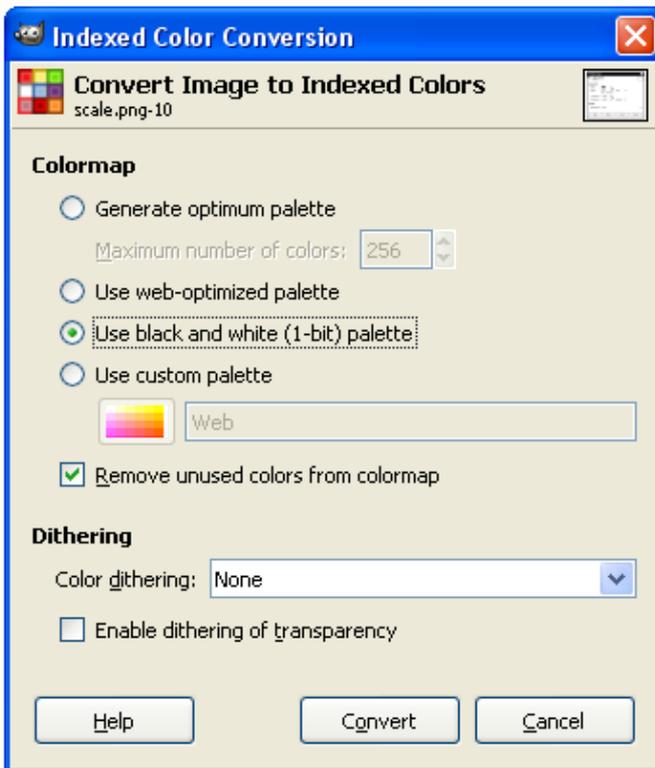
Farbkurven: Colors → Curves
Verstärkung/Abschwächung von einzelnen Farbanteilen



GIMP

Wichtige Funktionen in GIMP:

Farben reduzieren: Image → Mode → Indexed



Reduktion der enthaltenen Farben
auf kleineren Umfang

Auswahl verschiedener Paletten

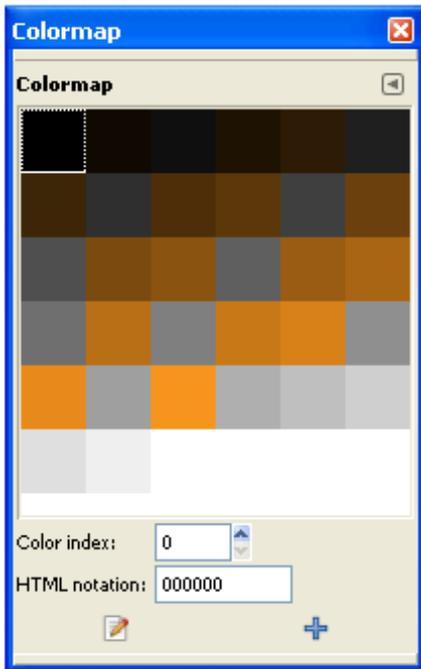
und Rasterungs- (Dithering) Algorithmen

GIMP

Wichtige Funktionen in GIMP:

Farbtabelle: (nur falls Indizierte Farben)

Windows → Dockable Dialogs → Colormap



Enthaltene Farben des Bildes
sind direkt manipulierbar

Farbtabelle

Speicherplatzberechnung für Bilder mit Farbtabelle am Beispiel:

Bild (3x3 Pixel) mit 4
Verschiedenen Farben (0-3):

2	2	1
1	1	0
3	3	3

9 Pixel x 2 Bit = **18 Bit**

Da sowohl die Tabelle als auch die Pixel-
daten in der Datei gespeichert werden
müssen ergibt sich dadurch eine Datei-
größe von 18 Bit + 104 Bit = **122 Bit**

Indizes
2 Bit/Index,
da 4 Indizes
Vorhanden
($2^2 = 4$)

Farbtabelle:

RGB-Werte

0	00 00 00
1	00 00 FF
2	00 FF 00
3	FF 00 00

4 Farben x 24 Bit = 96 Bit = 12 Byte

4 Indizes x 2 Bit = 8 Bit

→ benötigt 96 Bit + 8 Bit = **104 Bit**

Aufgabe 2

Ein Bild mit 12 Farben (von 0 bis 11) ist gegeben.

Das Bild soll einmal mit und einmal ohne Farbtabelle gespeichert werden. Vergleichen Sie den benötigten Speicherplatz, falls 24-Bit-Farben verwendet werden.

4	0	8	1	4
10	1	9	4	2
8	9	11	6	8
7	11	5	10	5
3	2	7	3	11

Lösung zu Aufgabe 2

Ohne Farbpalette:

25 Pixel, jeweils 24 Bit: $25 * 24 = 600 \text{ Bit}$

Mit Farbpalette:

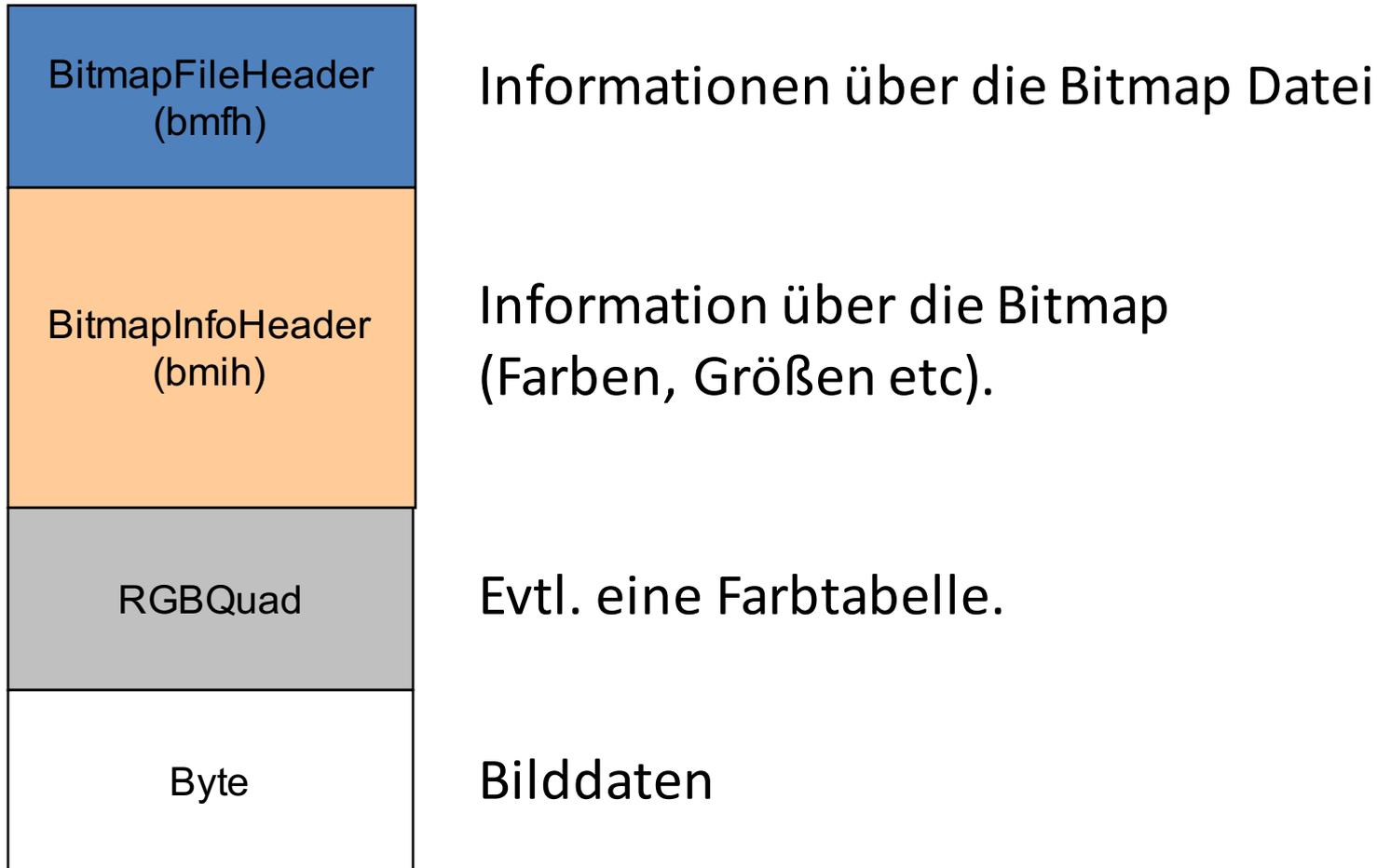
Zur Referenzierung einer Farbe benötigt man mindestens 4 Bit (größter Wert in der Palette 11).

Palette:	12 Farben, jeweils 24 Bit	$12 * 24 = 288 \text{ Bit}$
Bild:	25 Pixel, jeweils 4 Bit (max 11)	$25 * 4 = 100 \text{ Bit}$
Zuordnung:	Palettenfarben zu den 4-Bit-Folgen	$12 * 4 \text{ Bit} = 48 \text{ Bit}$
Zusammen:	$288 + 100 + 48 = 436 \text{ Bit}$	

Bitmap Format

für mehr Informationen:

https://de.wikipedia.org/wiki/Windows_Bitmap



BitmapFileHeader

Adresse		Größe (Byte)	Zweck
00	bfType	2	Immer "BM"
02	bfSize	4	Dateigröße in Bytes
06	bfReserved1	2	Immer 0
08	bfReserved2	2	Immer 0
10	bfOffBits	4	Offset vom Anfang der Datei zu den eigentlichen Bitmap Daten

BitmapInfoHeader

Adresse		Größe (Byte)	Zweck
14	biSize	4	Größe des BmlH in Bytes
18	biWidth	4	Bildbreite in Pixel
22	biHeight	4	Bildhöhe in Pixel
26	biPlanes	2	Anzahl der Flächen (immer 1)
28	biBitCount	2	Anzahl der Bits pro Pixel
30	biCompression	4	Kompression (0 = keine)
34	biSizeImage	4	Größe der Bilddaten in Bytes
38	biXPelsPerMeter	4	X Pixel Pro Meter (meistens 0)
42	biYPelsPerMeter	4	Y Pixel Pro Meter (meistens 0)
46	biClrUsed	4	Anzahl der Farben
50	biClrImportant	4	Anzahl der wichtigen Farben

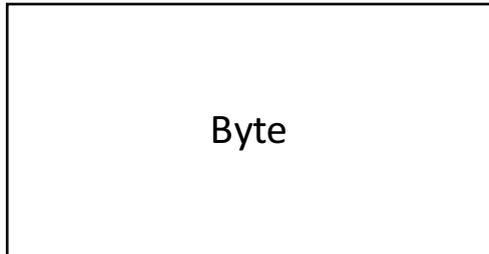
Farbtabelle



Eintrag für eine Farbe:

Adresse innerhalb der Farbe	Größe in Bytes	Name	Beschreibung
0	1	rgbBlue	Blauer Farbanteil
1	1	rgbGreen	Grüner Farbanteil
2	1	rgbRed	Roter Farbanteil
3	1	rgbReserved	Immer 0

Bilddaten



Je nach Einstellungen entweder BGR Werte oder Referenz auf die Tabelle.

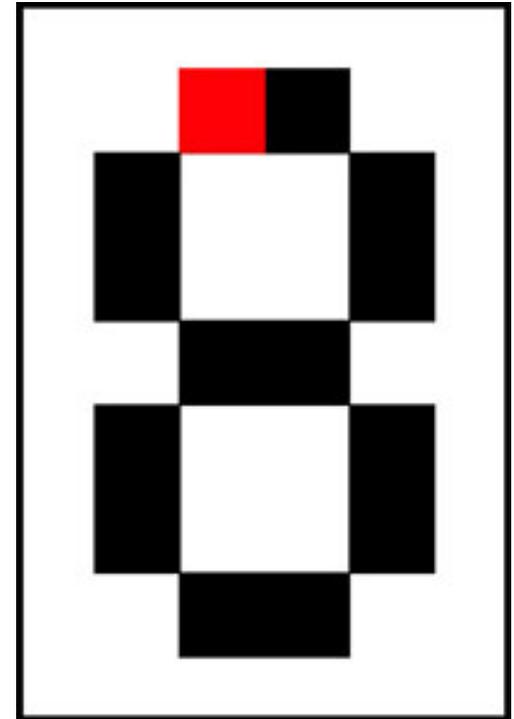
Die erste Datenzeile in der Datei repräsentiert die unterste Zeile im Bild.

Zeilen deren Bytes nicht durch 4 teilbar sind werden mit beliebigen Bytes aufgefüllt.

Aufgabe 3 a) und b)

Im ZIP-Archiv zur Aufgabe finden Sie die Datei "aufgabe3.bmp".

- a) Welche Bildgröße in Pixel hat die Grafik "aufgabe3.bmp"? Geben Sie auch die Stelle im Hexcode und den entsprechenden Hexadezimal-Code mit an.
- b) In dem Bitmap ist eine geheime Nachricht versteckt. Finden Sie diese Nachricht und beschreiben Sie, warum dies möglich ist ohne Auswirkungen auf das Bild zu haben. Ihre Lösung muss sowohl die geheime Nachricht als auch die Erklärung enthalten.



Lösung zu Aufgabe 3 a) und b)

- a) Breite: Offset 18 – 4 Byte → 06 00 00 00 → 6 Pixel
Höhe: Offset 22 – 4 Byte → 09 00 00 00 → 9 Pixel

- b) Nachricht: „DM ROCKT“

Im Bitmap Format werden Reihen, bei denen die benötigte Anzahl von Bytes kein Vielfaches von 4 sind solange mit zusätzlichen Bytes aufgefüllt bis diese Eigenschaft eintrifft. Diese Bytes können beliebigen Wert haben. Daher haben diese Änderungen keine Auswirkung auf die Datei.

6px pro Reihe, 3 Byte pro Farbe → $6 \cdot 3$ Byte pro Reihe = 18
→ 2 Byte müssen zur 20 aufgefüllt werden.

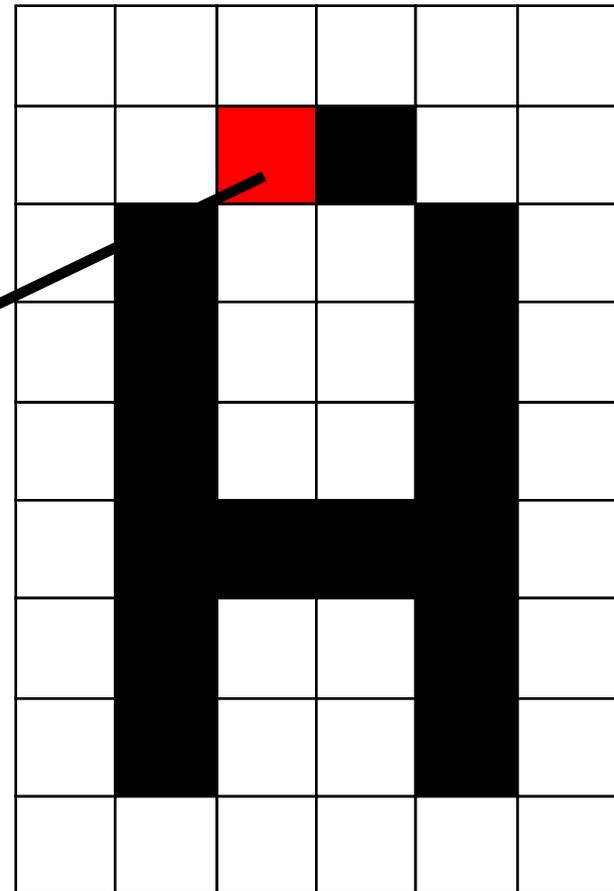
Aufgabe 3 c)

Nutzen Sie einen Hex-Editor, um die Bitmap Datei zu manipulieren. Fügen Sie den ersten Buchstaben Ihres Vornamens in das Bitmap ein. Der äußere Rand der Grafik muss weiß bleiben. Der Buchstabe soll bis auf einen Pixel schwarz sein. Der oberste linke Pixel des Buchstabens soll in einem vollen Rot dargestellt werden analog zum Beispiel rechts, das eine 8 zeigt, welche in die Grafik eingefügt wurde. Nutzen Sie dazu ausschließlich den Hex-Editor. Die geheime Nachricht darf dabei nicht verloren gehen. Fügen Sie Ihrer Abgabe sowohl die Beschreibung der Herangehensweise als auch das fertige Bild bei.

Lösung zu Aufgabe 3 c)

Beispiel A:

```
42 4DEC 00 00 0000 0000 00 36 00 00
00 28 00 00 0006 0000 00 09 00 00 00 01
00 18 00 00 0000 00B6 00 00 00 12 0B
00 00 12 0B 00 0000 0000 00 0000 00 00
FF FF
FF FF FF FF FF FF 44 4D FF FF FF 00 00
00
FF FF FF FF FF FF 0000 00 FF FF FF 12
00 FF FF FF 00 0000 FF FF FF FF FF FF
00 0000 FF FF FF 52 4F FF FF FF 00 00
00 0000 00 0000 0000 00 00 FF FF FF
43 4B FF FF FF 00 0000 FF FF FF FF FF
FF 00 0000 FF FF FF 00 00 FF FF FF 00
00 00 FF FF FF FF FF FF 00 00 FF FF
FF 54 00 FF FF FF 00 00 00 FF FF FF FF
FF FF 00 00 00 FF FF FF 00 00 FF FF FF
FF FF FF 00 00 FF 00 00 00 FF FF FF FF
FF FF 00 00 FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF 00 00 00 00
```



Wichtig: Daten werden „kopfüber“ und Farben als BGR gespeichert

Übungsblatt 8

- Übungsblatt 8:
<https://www.medien.ifi.lmu.de/lehre/ws1617/dm/>
- Abgabe bis Freitag den 23.12.2016, 09:00 Uhr
morgens in [UniWorX](#)