

6. Licht, Farbe und Bilder

- 6.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 6.2 Farbmodelle
- 6.3 Raster-Bilddatenformate 
 - Grundbegriffe für Bildspeicherung und -Bearbeitung 
 - Bitmap-Formate
 - Verlustfrei komprimierende Formate
- 6.4 Verlustbehaftete Kompression bei Bildern
- 6.5 Weiterentwicklungen bei der Bildkompression

Literatur:

Quelle für Informationen zu diversen Dateiformaten: <http://www.wotsit.org>

John Miano: Compressed Image File Formats, Addison-Wesley 1999

Klassifikation von Bild-Dateiformaten

- Vektorgrafik
 - Enthält *keine* Rasterdaten, sondern Beschreibung von Einzelobjekten
 - Beispiele: SVG (Scalable Vector Graphics) und div. proprietäre Formate
- Rastergrafik (Bitmap) 
 - Speicherung der Rasterdaten eines Bildes
 - Einfache verlustfreie Kompression
 - Beispiele: BMP, TIFF
- „Meta-Files“
 - Kombination von Vektorgrafik und Rasterdaten
 - Beispiele: WMF (Windows Meta File), Macintosh PICT
- Erweiterte Bitmap-Formate
 - Bessere Kompression, zusätzliche Funktionalität (z.B. Animation)
 - Beispiele: GIF, PNG
- Stark komprimierende Formate
 - (Auch) verlustbehaftete Kompression
 - Beispiel: JPEG

Hinweis:
Mehr Informationen
zu SVG/Vektorgrafik
später in der Vorlesung

Bildgröße und Auflösung

- *Pixel (picture element)*: Kleinste Einheit eines Bildes, Bild„punkt“
 - Die tatsächliche Größe eines Pixels hängt vom Ausgabegerät ab.
 - Seitenverhältnis (*pixel aspect ratio*) muss nicht 1 sein
- Bildgröße für Bildschirmdarstellung in Pixel
 - Beispiel: Bild der Größe 131 x 148 Pixel
- *Auflösung*: Anzahl der Pixel, die auf einer bestimmten Strecke zur Darstellung zur Verfügung stehen
 - Angabe in *ppi (pixel per inch)*, Standardwert 72 ppi (d.h. 1 Pixel = 1 Pica-Punkt)
- Zusammenhang Abmessungen / Auflösung / Pixelgröße:
 - $\text{Breite [px]} = \text{Breite [in]} * \text{Auflösung [ppi]}$
- *Skalierung*: Konversion des Bildes auf andere Auflösung (*resampling*)
 - Abwärtsskalierung, „Herunterrechnen“: Bildung von Mittelwerten
 - Aufwärtsskalierung, „Hochrechnen“:
 - » Nur eingeschränkt automatisch möglich
 - » Diverse Interpolationsalgorithmen (z.B. "Bi-kubische Interpolation")



Anschaulich: Anzahl der Pixel im Bild



100



50



25

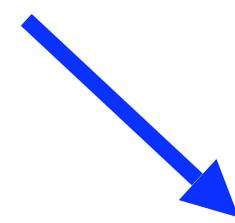


12

Farbtiefen und Farbkanäle

- *Farbtiefe (color resolution)*: Anzahl der Farben, die pro Pixel gespeichert werden können
 - Typische Werte:
 - » 2 Farben (1 bit) = schwarz-weiss
 - » 16 Farben (4 bit)
 - » 256 Farben (8 bit)
 - » 16,7 Millionen Farben (24 bit)
 - „True Color“:
 - » 24 bit Farbtiefe
 - » 1 Byte je Grundfarbe (R, G, B)
- *Farbkanal*: Teil der gespeicherten Information, der sich auf eine der Primärkomponenten des gewählten Farbmodells bezieht
 - Bei Rohdaten meist: Rot, Grün und Blau (RGB-Modell)
 - bei Druckvorbereitung auch CMY bzw. CMYK („Vierfarbdruck“)

Anschaulich: RGB Farbkanäle



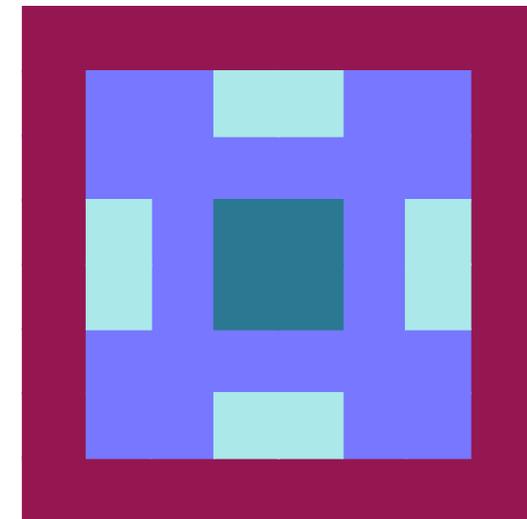
Farbpaletten und indizierte Farben

- *Farbpalette*: Die Menge der in einem konkreten Bild tatsächlich enthaltenen Farben
 - Teilmenge der insgesamt möglichen Farben
- *Indizierte Speicherung*:
 - Farbpalette (Tabelle) enthält die im Bild vorkommenden Farben
 - Pro Pixel wird nur der Index in die Palettentabelle gespeichert

1	1	1	1	1	1	1	1
1	2	2	3	3	2	2	1
1	2	2	2	2	2	2	1
1	3	2	0	0	2	3	1
1	3	2	0	0	2	3	1
1	2	2	2	2	2	2	1
1	2	2	3	3	2	2	1
1	1	1	1	1	1	1	1

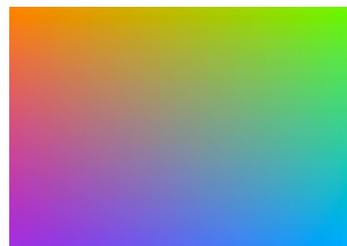
	R	G	B	
0	35	101	128	
1	128	0	64	
2	99	92	254	
3	156	227	227	

Speicherplatz:
 $8 \times 8 \times 2 \text{ bit} +$
 $4 \times 3 \times 8 \text{ bit} + 4 \times 2 \text{ bit} =$
 232 bit
 (statt 1536 bit)



Dithering

- Farbverläufe sind bei Reduzierung auf wenige Palettenfarben schlecht darstellbar
- Dithering: Darstellung von Verläufen durch Punktmuster höherer Auflösung, wobei die einzelnen Punkte Farben aus der verkleinerten Palette haben
 - Bekanntester Algorithmus: Floyd-Steinberg
 - Wird oft auch von Anzeigeprogrammen (z.B. Browsern) durchgeführt



16,7 Mio Farben

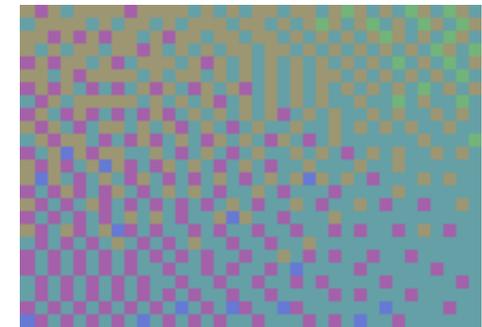


16 Indexfarben
ohne Dithering



16 Indexfarben
mit Dithering

Ausschnitt-
vergrößerung:



Beispiel: 2, 4, 256 Helligkeitsstufen je Farbkanal



256



4



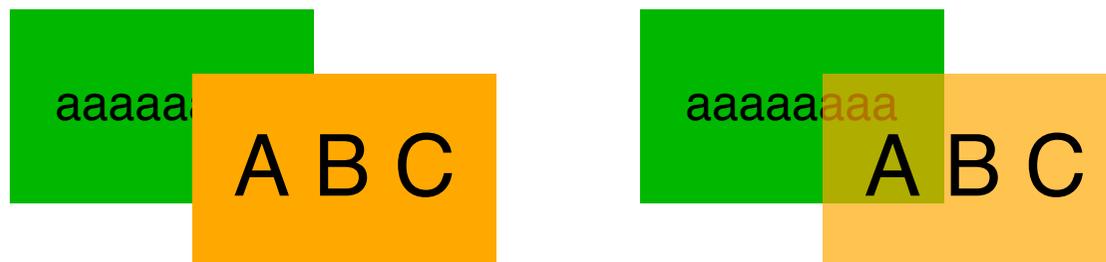
2



2 +
dither

Transparenz: Alphakanal

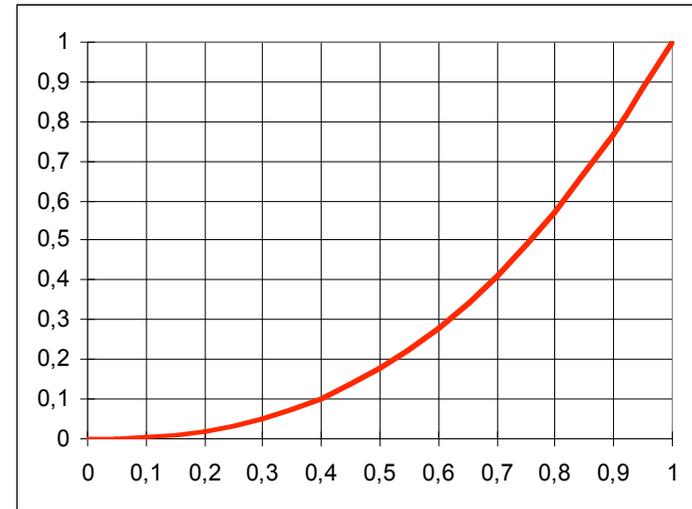
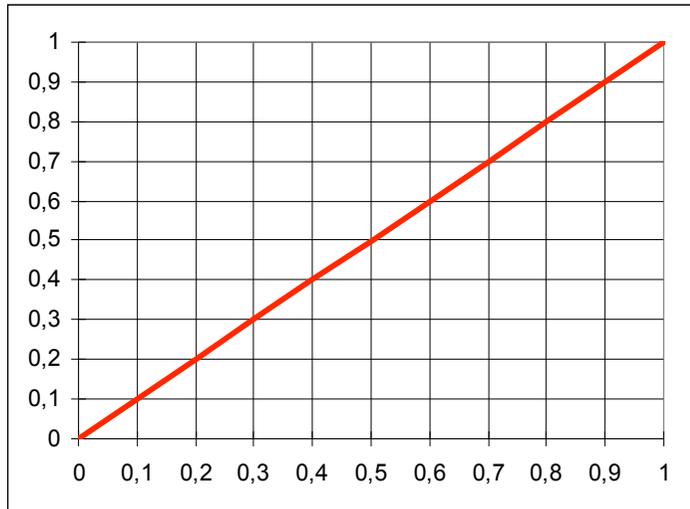
- Bilder bestehen oft aus verschiedenen Elementen, die sich überlagern
 - z.B. Hintergrund, Vordergrund
- Zum Überlagern müssen oft Objekte „transparent“ (durchscheinend) werden
- *Alphakanal*: Zusätzliche Information zu einem Bild, die den Grad der Transparenz angibt
 - Sinnvoll vor allem bei der Zusammensetzung eines Bildes aus verschiedenen Ebenen (*layers*)
 - Häufig verwendet in Bildbearbeitungsprogrammen
 - Technisch: Zusätzliche Information pro Pixel („RGBA-Farbmodell“)



Gamma-Korrektur

- Jedes Ausgabegerät zeigt Farben geringfügig anders an
 - Phosphortypen bei Monitoren
 - Grafik-Subsysteme verschiedener Betriebssysteme (z.B. Macintosh vs. Windows)
- *Gammawert*: beschreibt das Verhältnis zwischen den gespeicherten und auf dem aktuellen Anzeigegerät angezeigten Farben
 - im Wesentlichen Änderung der Helligkeit, Details gleich!
- *Gammakorrektur*: Anpassung der angezeigten Farben an die beabsichtigten Farben
 - Bei der Anzeige auf einem bestimmten Monitor (ohne Monitoreinstellungen zu verändern)
 - Bei der Anzeige von Dateien aus „fremden“ Plattformen, deren Gammawert bekannt ist
- Gammakorrektur wird realisiert in:
 - Bildbearbeitungsprogrammen
 - Anzeigeprogrammen für bestimmte Dateiformate (z.B. PNG)

Mathematische Beschreibung: Gamma



- Röhrenmonitor mit Phosphor als Leuchtstoff:
 - Steuerspannung u von 0,0 bis $1,0 * u_{\max}$
 - Helligkeit h von 0,0 bis $1,0 * h_{\max}$
- Verhältnis von Helligkeit eines Punktes zur Steuerspannung ist nicht linear, sondern exponentiell:
 $h = u^\gamma$ wobei γ (sprich: Gamma) = 2,5

6. Licht, Farbe und Bilder

- 6.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 6.2 Farbmodelle
- 6.3 Raster-Bilddatenformate 
 - Grundbegriffe für Bildspeicherung und -Bearbeitung
 - Bitmap-Formate 
 - Verlustfrei komprimierende Formate
- 6.4 Verlustbehaftete Kompression bei Bildern
- 6.5 Weiterentwicklungen bei der Bildkompression

Literatur:

Quelle für Informationen zu diversen Dateiformaten: <http://www.wotsit.org>

Windows BMP-Format

- Standardformat aus Microsoft DOS und Windows
- Rasterformat mit zulässigen Farbtiefen 1, 4, 8 und 24 bit
- Verwendet eine Farbpalette (*color table*) (bei niedrigeren Farbtiefen als 24 bit)
- Besteht aus:
 - Kopfinformation
 - Farbtabelle
 - Daten
- Datenablage zeilenweise
- 4- und 8-bit-Variante unterstützen Lauflängen-Kompression:
 - RLE4 und RLE8
 - Zwei Bytes (RLE8) bzw. Halbbytes (RLE4) als Einheit:
 - » Erstes Byte: Anzahl der beschriebenen Pixel
 - » Zweites Byte: Index in Farbtabelle für diese Pixel
- Spezielle Variante mit Alphakanal: „BMP4“

Tagged Image File Format TIFF

- Entwickelt ca. 1980 von Aldus
(Firma Aldus inzwischen von Adobe übernommen)
 - Portabilität, Hardwareunabhängigkeit, Flexibilität
- Unterstützt ca. 80 verschiedene Varianten zur Datenspeicherung und deren Kombination
 - z.B. schwach aufgelöstes „Preview“-Bild und hochaufgelöstes Bild
 - Farbmodell explizit angegeben
- Kann Metainformation (z.B. über Ursprungshardware) speichern
- Grundstruktur:
 - Header
 - Liste von *Image File Directories*
 - » Image File Directory: Liste von Tags
(jeweils pro Tag: Typ, Datentyp, Länge, Zeiger auf Daten)
 - Datenbereich

6. Licht, Farbe und Bilder

- 6.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 6.2 Farbmodelle
- 6.3 Raster-Bilddatenformate 
 - Grundbegriffe für Bildspeicherung und -Bearbeitung
 - Bitmap-Formate
 - Verlustfrei komprimierende Formate 
- 6.4 Verlustbehaftete Kompression bei Bildern
- 6.5 Weiterentwicklungen bei der Bildkompression

Literatur:

John Miano: Compressed Image File Formats, Addison-Wesley 1999

GIF-Format: Allgemeines

- GIF = Graphics Interchange Format
 - eingeführt von CompuServe 1987 („GIF87a“)
 - Heute verwendete Version von 1989 („GIF89a“) mit kleinen Modifikationen
- Verlustfreie Kompression (mit LZW)
- Kleiner Farbumfang (max. 256 Farben in einem Bild)
- Flexible Anzeigeeoptionen (z.B. *interlaced* und Animation)
- Optimal für kleinere Grafiken und Gestaltungselemente
- Wenig geeignet für hoch auflösende Bilder (z.B. Fotos)
- Patent-Streit:
 - Unisys hat Patent auf den verwendeten LZW-Algorithmus
 - 1999: Ankündigung von Lizenzforderungen für GIF-Grafiken
 - Initiativen zum Ersatz von GIF (z.B. durch PNG)
- Im folgenden: Weit gehend komplette Darstellung des Formats (als typisches Beispiel)
 - Abgrenzungszeichen, Blocklängenfelder etc. hier nicht beschrieben

BNF-Grammatik für das GIF-Format

<GIF Data Stream> ::= Header <Logical Screen> <Data>* Trailer
<Logical Screen> ::= Logical Screen Descriptor [Global Color Table]
<Data> ::= <Graphic Block> | <Special-Purpose Block>
<Graphic Block> ::= [Graphic Control Extension] <Graphic-Rendering Block>
<Graphic-Rendering Block> ::= <Table-Based Image> | Plain Text Extension
<Table-Based Image> ::= Image Descriptor [Local Color Table] Image Data
<Special-Purpose Block> ::= Application Extension | Comment Extension



Global Color Table in GIF

- Eine GIF-Datei kann mehrere Bilder enthalten.
- Farbtabellen (Paletten)
 - entweder global für alle enthaltenen Bilder (Global Color Table)
 - oder lokal je Bild
- Lokale Farbtabelle hat Vorrang vor globaler Tabelle
- Hintergrundfarbe für Gesamtdarstellung möglich, wenn globale Farbtabelle existiert
- Sortierung der globalen Farbtabelle:
 - Reihenfolge der Farben in globaler Farbtabelle nach Häufigkeit sortiert

GIF-Elemente im Detail (1)

- Header:
 - Signatur „GIF“
 - Version (87a oder 89a)
- Logical Screen Descriptor:
 - Breite und Höhe des „logischen Bildschirms“ in Pixel
 - Global Color Table Flag (ist eine solche Tabelle vorhanden?)
 - Color Resolution (Welche Auflösung hatte das Ursprungsbild?)
 - Sort Flag (Ist die globale Farbtabelle sortiert?)
 - Size of Global Color Table s (maximale Anzahl im Bild benutzter Farben)
 - Background Color Index (Hintergrundfarbe - Index in die globale Farbtabelle)
 - Pixel Aspect Ratio (Seitenverhältnis der Pixel)
- Global Color Table (optional)
 - $3 * 2^{s+1}$ Bytes, jeweils Rot-, Grün- und Blau-Wert

Transparenzfarbe in GIF

- In GIF (89) kann eine Farbe der Tabelle als „transparent“ gekennzeichnet werden.
 - Pixel dieser Farbe werden nicht angezeigt, statt dessen Hintergrund
 - Das ist keine echte Transparenz im Sinne eines Alphakanals!



GIF-Elemente im Detail (2)

- Graphic Block:
 - entweder Bild (Table-Based Image) oder Textblock
 - Textblöcke in der Praxis kaum verwendet
- Graphic Control Extension (nur in GIF89):
 - Disposal Method (Was passiert nach abgeschlossener Anzeige?)
 - » Optionen: Keine Aktion, Hintergrundfarbe, früheres Bild
 - User Input Flag (Benutzereingaben erwartet? - kaum verwendet)
 - Transparency Flag (Wird eine Transparenzfarbe benutzt?)
 - Delay Time (Wartezeit bis nächstes Bild angezeigt wird)
 - Transparency Index (Index in Farbtabelle für Transparenzfarbe)

GIF-Elemente im Detail (3)

- Image Descriptor:
 - Image Top Position (Lage der linken oberen Ecke auf dem log. Bildschirm)
 - Image Width & Height (Breite und Höhe des Einzelbildes)
 - Local Color Table Flag (Gibt es eine lokale Farbtabelle?)
 - Interlace Flag (Wird Interlacing verwendet?)
 - Sort Flag (Ist die lokale Farbtabelle sortiert?)
- Local Color Table:
 - Analog zur Global Color Table
- Table Based Image Data
 - LZW Minimum Code Size: Startlänge der LZW-Codes
 - Image Data: Bilddaten, LZW-komprimiert, strukturiert in 255-Byte-Blöcke

Interlacing in GIF

- Ziel: Kürzere empfundene Ladezeit für Betrachter, z.B. bei Web-Grafik
- Bild wird schrittweise in Zeilen aufgebaut
 - 1. Durchlauf: Jede 8. Zeile beginnend in Zeile 0
 - 2. Durchlauf: Jede 8. Zeile beginnend in Zeile 4
 - 3. Durchlauf: Jede 4. Zeile beginnend in Zeile 2
 - 4. Durchlauf: Jede 2. Zeile beginnend in Zeile 1



LZW-Algorithmus beim GIF-Format

- In den Datenbereich eingetragen werden:
 - Indizes in die aktuelle Farbtabelle (Länge meist 8 bit) als Repräsentation von Einzel-Pixeln
 - Weitere Indizes (Länge zwischen Pixel-Indizes+1 und 12 bit) als Repräsentation von Pixelfolgen (zeilenweise)
- Startbelegung der LZW-Code-Tabelle
 - ist implizit mit der Farbtabelle gegeben
- Rücksetzen der LZW-Codierung
 - Spezieller Reset-Code (*clear code*) erlaubt völligen Neustart der Codierung
 - Im Prinzip an jeder Stelle möglich, v.a. am Beginn eines neuen Bildes
- Packen von Bitcodes in Bytes
 - Codes werden in Bytes (8-bit-Worte) gepackt
 - Platzersparnis gegenüber 1 Byte (oder mehr) je Codewert

Animated GIF

- GIF-Datei mit mehreren Bildern als einfacher „Film“
 - Bilder enthalten verschiedene Stadien der Animation
 - Anzeigeprogramm zeigt zyklisch die verschiedenen Bilder an, mit definierter Wartezeit dazwischen
- Praktische Bedeutung:
 - Eine der einfachsten Formen, Besucher von Web-Seiten vom eigentlichen Inhalt abzulenken ...
 - Heutzutage sehr schwach im Vergleich zu Animationstechniken wie Macromedia Flash (sh. später)
 - Dennoch: Einfach handzuhaben und plattformübergreifend stabil implementiert



GIF



Special Purpose Blocks in GIF

- Application Extension
 - Möglichkeit für Hersteller, nach Registrierung bei CompuServe spezielle Informationen zur Aktivierung von Programmfeatures zu hinterlegen
 - Weit verbreitet:
 - » Netscape 2.0 Loop Extension
 - » Realisiert Animation von GIFs
- Comment Extension
 - Möglichkeit zur Ablage von nicht angezeigter Metainformation
 - z.B. Autor, Copyright, Kontakt

Portable Network Graphics PNG („Ping“)

- Geschichte:
 - Ausgelöst durch Comuserve's Ankündigung, auf das GIF-Format Lizenzgebühren zu erheben (1994)
 - Arbeitsgruppe beim W3C für PNG, standardisiert 1996
 - PNG ist offen und lizenzfrei
- Ziel:
 - Besserer Ersatz für GIF, teilweise auch Ersatz für JPEG
 - Mäßige praktische Verbreitung
- Farbtiefen:
 - 24 oder 48 bit „TrueColor“, 8 oder 16 bit Graustufen, Paletten bis 256 Farben (optional)
- Hauptvorteile:
 - Völlig verlustfrei ("Deflate"-Algorithmus: Lempel-Ziv- + Huffman-Kompression)
 - Echter Alpha-Kanal
 - Gamma-Korrektur (Gamma-Wert der Quellplattform speicherbar)
 - Verbessertes Interlacing (7-Pass-Algorithmus „Adam7“)
 - Bessere Kompression (Kompressionsfilter)
 - Integritätstest für Dateien (*magic signature*, CRC-32)

PNG: Beispiel



Echter Alpha-Kanal in PNG

- Alpha-Werte pro Pixel gespeichert
 - 4 Bytes pro Pixel: „RGBA“-Farbmodell
 - Ermöglicht elegante Schatten und Übergänge zwischen Grafik und Hintergrund
- Vermeidet Wechselwirkungen zwischen Anti-Aliasing und Transparenzfarbe
 - Bei „binärer Transparenz“ wie in GIF oft „weißer Rand“ um transparente Grafiken aufgrund von Anti-Aliasing (erzeugt nicht-transparente Farben)



Kompressionsverbesserung durch Filter in PNG

- Beispiel:
 - Wertfolge 1, 2, 3, 4, 5, 6, 7, 8, 9, ...
 - Komprimiert extrem schlecht mit LZ-artigen Algorithmen
- Filter (Prädiktion):
 - Ersetze alle Zahlen (außer der ersten) durch die Differenz zur vorhergehenden
 - Wertfolge: 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
 - Komprimiert exzellent! (viele Wiederholungen)
- Filter in PNG:
 - Sub: Differenz zum linksstehenden Byte
 - Up: Differenz zum darüberstehenden Byte
 - Average: Differenz zum Durchschnitt der Sub- und Up-Bytes
 - Paeth: Differenz zum *Paeth-Prediktor*
 - » Benutzt linksstehendes, darüberstehendes und "links oben" stehendes Byte
 - Heuristiken zur Wahl des passenden Filters

Welches Format wofür?

- Für Web-Grafiken (klein, geringe Farbanzahl)
 - GIF oder PNG
- Für Bilderzeugung mit Scanner oder Austausch über diverse Geräte hinweg:
 - TIFF
- Für hochauflösende Bilder mit vielen Farben (Fotos)
 - JPEG (wegen wesentlich besserer Kompression)
 - Bei grossen einheitlichen Farbflächen evtl. auch PNG (beste Qualität)