

## 4. Speicher- und Austauschformate

Grundlegende Unterscheidung: Vektor- vs. Rastergrafik

*Vektorgrafik* (älteres Verfahren): basiert auf "Polylinien" (vgl. Pfade in PostScript), d.h. attributierte Geradenstücke. Vorteil: beliebig skalierbar (Ähnlichkeitsabbildung, vgl. `scale` in PostScript).

*Rastergrafik*: Einzelpunkte in regelmäßigem Gitter (meist Rechteckgitter), Speicherung in Form von Matrizen. Vorteil: Nähe zur Display-Hardware (CRT, LCD).

Skalierungsverfahren für Rastergrafiken:

Verkleinern: z.B. wähle nur alle Bildpunkte mit geradzahligem Indexpaar (problematisch)

Vergrößern: Setze alle Bildpunkte mit geradzahligem Indexpaar nach dem Original, andere Bildpunkte ergeben sich aus rechtem (in gerader Zeile) oder unterem (in ungerader Zeile) Nachbarn.

⇒ Informationsverlust beim Verkleinern; beim Vergrößern kein Genauigkeitsgewinn.

Zur Farbspezifikation in Rastergrafik-Formaten:

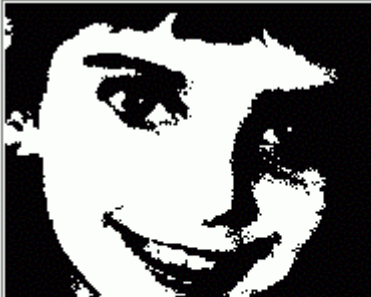
Angabe der "*Farbtiefe*" in Bit pro Pixel.

Fotorealistische Bilder: 24 Bit pro Bildpunkt, d.h. je 8 Bit für Rot, Grün und Blau ⇒ 256 Rot-, 256 Grün- und 256 Blautöne ⇒ ca. 16,8 Millionen Farben (die aber alle im Gamut des Monitors bzw. Druckers liegen!). "True Color", kontinuierl. Farbverläufe. Beispiel: (FF, 00, FF) = (255, 0, 255) = (R,G,B) ist Magenta. (Zum Vergleich: Gleichzeitig auf dem Monitor sind bei einer Auflösung von 1024 x 1024 Pixeln ca. 1 Million Bildpunkte.)

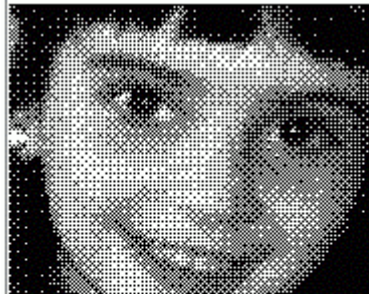
Mittlere Qualität: 8 Bit pro Bildpunkt. Feste Zuordnung: 3 Bit für Rot, 3 Bit für Grün, 2 Bit für Blau ⇒ 256 feste Farben. Flexibler: variable Farbabbildung über 3-spaltige *Color Lookup Table* (CLUT) mit 256 Einträgen, editierbar. ("Pseudo-Color")

Geringe Qualität: 4 Bit pro Bildpunkt für Grauwertbilder (16 Graustufen), 1 Bit pro Bildpunkt für s/w-Bilder (Bitmaps).

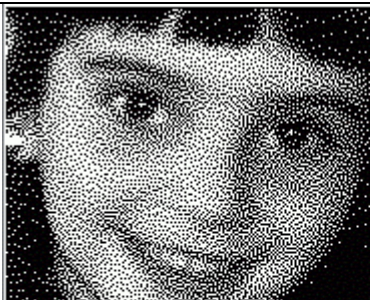
Durch Rasterung können auch mit Bitmaps Grauwert-Eindrücke erzeugt werden. Beispiele:



Strichzeichnung  
nur schwarz oder weiß, keine durch Raster dargestellte Übergänge



Autotypisch gerastertes Bitmap  
verschiedene Helligkeitswerte werden durch verschieden große Rasterpunkte dargestellt



Frequenzmoduliert gerastertes Bitmap  
verschiedene Helligkeitswerte werden durch Anhäufungen gleich großer, im Abstand variabler Rasterpunkte erzeugt



Sonstige Rasterungen  
Verschiedene Helligkeitswerte werden für besondere gestalterische Gesichtspunkte durch verschieden starke geometrische Gebilde erzeugt (hier Linien). Möglich sind Linien, Wellen, kreisförmige Raster ...



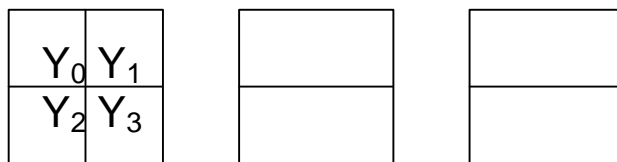
*Bild mit echten Graustufen* (aus Dünnbier 2001)

Der menschliche Gesichtssinn kann ca. 60 Graustufen unterscheiden.

Alpha-Kanal: In einigen Anwendungen und Formaten werden True Color- (24 Bit-) Bilder mit einem zusätzlichen Byte codiert (d.h. mit 32 Bit / Pixel) – Information über die Transparenz im zusätzlichen Byte (nicht standardisiert).

Farb-Subsampling:

In den Farbmodellen mit Helligkeits-Koordinate (YIQ, YUV) wird oft die Farbinformation ungenauer gespeichert als die Helligkeit Y. Dafür werden die Farbwerte (I, Q) benachbarter Pixel zusammengefasst zu "Makropixeln". Z.B. Subsampling-Schema "4:2:2": Je zwei horizontal aufeinanderfolgende Farbinformationen werden zusammengefasst, auf 4 Pixel Luminanzinformation kommen je 2 Makropixel Farbinformation. 33 % des Speicherplatzes werden eingespart.



## Bildkompressionsverfahren

(hier nur knapper Überblick, Kompression ist Thema der *Bildverarbeitung*)

Größe eines Bildes (oder einer Bildserie):

Breite x Höhe x Farbtiefe ( x Bilder pro Sekunde)

Beisp.: Film mit 320 x 200 Pixeln, 24 Bit Farbtiefe, Bildrate 25

Bilder pro Sek.: Datenvolumen von 4,6 MB/Sek. = 16,1 GB/h

⇒ Datenkompression erforderlich.

*Kompression*: Entfernung von Redundanz

Redundanz durch:

- Eigenschaften der vorliegenden Codierung des Bildes (z.B. überflüssige Bits)
- Regelmäßigkeiten in der Bildstruktur (z.B. Nachbarschaftsrelationen, große leere Flächen...)
- Wahrnehmungsschwächen

2 Typen von Kompression:

- verlustfrei (lossless): "Qualität vor Geschwindigkeit"
- verlustbehaftet (lossy): "Geschwindigkeit (bzw. Speicherplatz-Ersparnis) vor Qualität"

Kompressionsphasen:

1. Vorverarbeitung (z.B. Reduktion der Farbtiefe, Subsampling; s.o.)
2. Diskrete Basistransformation (z.B. Kosinustransformation, s.u.; Wavelet-Transformation)
3. Quantisierung (Entfernung der als redundant betrachteten Anteile durch Rundungsoperationen)
4. Codierung

Bei verlustfreien Verfahren wird nur codiert.

Gebräuchliche Codierungsverfahren:

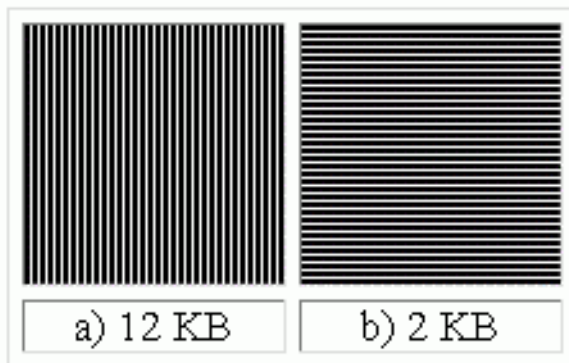
1. *Laufängencodierung* (RLE: Run Length Encoding; auch VLC: Variable Length Encoding)

Wiederholtes Auftreten von Werten wird durch die Angabe der Häufigkeit ersetzt.

- Ausnutzung der Redundanz benachbarter Pixel



Beispiel: Das Windows-BMP-Format bietet zeilenweise RLE an.



⇒ keine Kompression bei Vertikalstreifen.

## 2. Differenzcodierung (mit nachträglicher RLE)

einfachste Variante: Farbwerte von zwei Zeilen werden voneinander abgezogen. Hiermit würde auch bei obigem Bsp. a) eine Kompression erreicht.

## 3. Huffman-Codierung

Benutzt Statistik der Bilddaten.

### Huffman Code

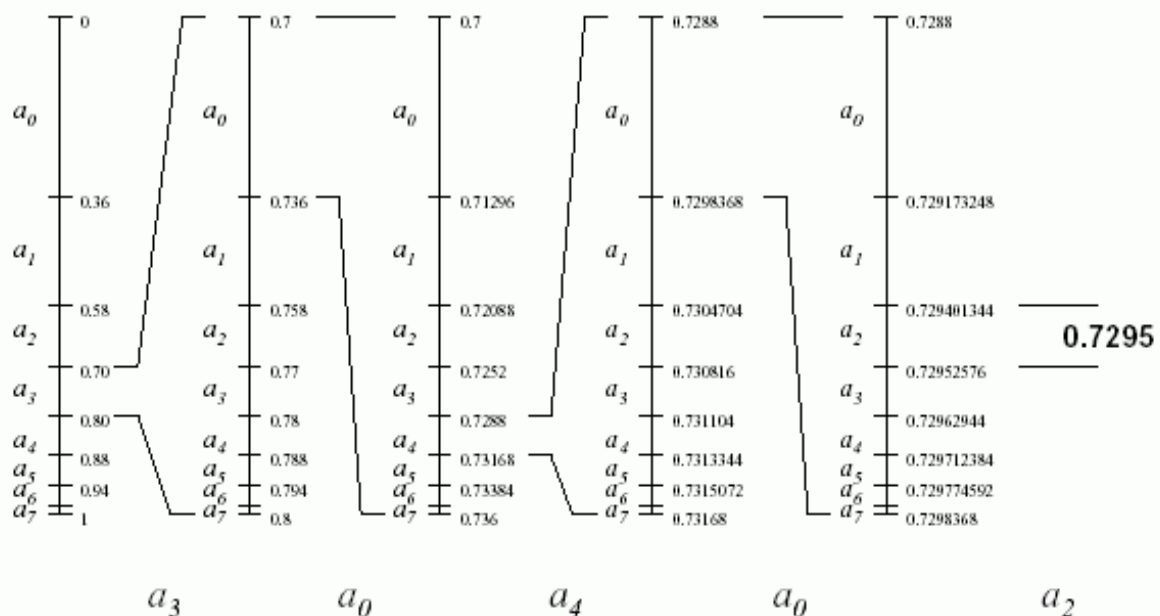
- Codewörter variabler Länge  
(ganzzahlige Anzahl von Bits)
- Repräsentation von Zeichen mit größerer Wahrscheinlichkeit durch kürzere Codewörter
- Codierung durch Erzeugen eines Huffman-Codebaumes
- Präfix-Eigenschaft  
→ eindeutige Decodierung

## 4. Arithmetische Codierung

Variante der Huffman-Codierung, bei der Zeichen durch Intervalle reeller Zahlen codiert werden. Intervall-Längen nach Häufigkeiten gewählt.

### Arithmetischer Code

- Codierung der gesamten Nachricht als ein Codewort
  - ➔ keine Diskrepanz zwischen Codewortlänge und Informationsgehalt
- Repräsentation eines Codewortes durch ein Intervall reeller Zahlen zwischen 0.0 und 1.0
- Codierung durch sukzessive Verkleinerung des Intervalls



Das Verfahren ist patentiert und darf nicht ohne Lizenzierung verwendet werden. Der Code nähert sich bei sehr langen Nachrichten einer optimalen Codierung.

In der Praxis werden binäre Gleitkommazahlen für obere und untere Intervallgrenzen verwendet und der obere Wert nach der ersten Stelle abgebrochen, welche vom unteren Wert verschieden ist.

## 5. LZW-Codierung (nach Lempel, Ziv und Welch)

(Varianten hiervon auch in PKZIP und GZIP.)

Prinzip: Codetabelle aus statischem und dynamischem Teil.

Übertragen wird nur der statische Teil und der transformierte String; die dynamische Tabelle wird bei der Decodierung rekonstruiert. Bei der Codierung wird mehrfaches Auftreten von Teilstrings ausgenutzt.

Siehe Beispiel bei H. Rechner,

<http://homepages.uni-tuebingen.de/student/horst.rechner/Ausarbeitung.html>

## Formate für den Austausch von Bilddaten

Links zu Spezifikationen:

[http://www.cica.indiana.edu/graphics/image\\_specs/](http://www.cica.indiana.edu/graphics/image_specs/)

### A. Vektorbasierte Dateiformate

#### 1. *HPGL* (und *HPGL2*)

Hewlett Packard Graphics Language

Grafik-Sprache für Stiftplotter, max. 8 Stifte (d.h. 3 Bit Farbtiefe), ASCII-Dateien (einzeilig).

Befehlskürzel: z.B.

AA = Arc Absolute

AF = Advance Full Page

CI = Circle

PD = Pen Down

PU = Pen Up

PR = Plot Relative

usw., jeweils mit numerischen Parametern.

#### 2. *PostScript*

(Adobe Systems Inc. 1982)

Seitenbeschreibungssprache mit voller Mächtigkeit einer universellen Programmiersprache erlaubt auch Einbeziehung von Rasterdaten

- siehe Übung -

### 3. DXF

Drawing Interchange Files

(verwandt: DWF)

Transfersprache von AutoCAD (Firma Autodesk)

ASCII-Dateien aus 4 Sektionen:

Header Section – allgemeine Konfigurations-Spezifikationen

Tables Section – 6 Tabellen mit Spezifikationen zu Linienarten, Text-Styles, sichtbarer Ausschnitt, Benutzerkoordinaten-

Umrechnung

Blocks Section – Blockdefinitions-Einträge für Gruppierungen (Blöcke) von Elementen

Entities Section – eigentliche Zeichenbefehle

Gruppen-Struktur der Datei: je zwei aufeinanderfolgende Zeilen bilden eine "Gruppe"; 1. Zeile: Codewort (pos. Integer), z.B. 10 = "x-Koordinate des Startpunkts eines Zeichenelements", 62 = "Farbnummer", 6 = "Name eines Linientyps". 2. Zeile: Parameter (Zahl oder Schlüsselwort).

### 4. CGM (Computer Graphics Metafile)

entstanden in Anlehnung an das Grafische Kernsystem GKS.

Aktuelle Version: Standard nach ISO 8632-1992.

ASCII-Dateien (aber auch Binärcodierung möglich).

Rahmenstruktur (Begrenzungselemente):

```
begin metafile
```

```
(Bilddatei-Beschreibungselemente)
```

```
  begin picture
```

```
(Bild-Beschreibungselemente)
```

```
  begin picture body
```

```
(Kontroll-Elemente)
```

```
(Darstellungselemente)
```

```
(Darstellungsattribute)
```

```
  end picture
```

```
end metafile
```



Bilddatei-Beschreibungselemente: Identifikator, Festlegung des Datentyps (real oder integer) der virtual device coordinates (VDC), Angaben zur Präzision, Liste verwendeter Fonts...

Bildbeschreibungselemente: Skalierungs-Modus und -Faktor, Angaben zur Strichstärke und Marken-Größe, Angabe zur Gesamt-Ausdehnung (VDC extend)...

Kontroll-Elemente: weitere Präzisionsangaben, Definition eines Clipping-Rechtecks

Darstellungselemente: eigentliche grafische Informationen.

5 Gruppen von Elementen:

Linienelemente (z.B. polyline, elliptical arc...)

Markenelemente

Textelemente

Füllgebietselemente

Zellmatrix- (Raster-) Elemente

Wegen großer Farb- und Rastertabellen, die explizit mit ins CGM-File mit aufgenommen werden, können schon bei kleinen Zeichnungen sehr große Dateiumfänge entstehen.

## 5. IGES (Initial Graphics Exchange Standard)

Format für den Austausch zwischen CAD/CAE-Systemen; enthält Produktdaten und Herstellungsinformationen, 2D- und 3D-Zeichnungen, Drahtgitter- und Oberflächenmodelle.

zeilenorientiert, Gruppierungs- und Identifikationsnummern am *Ende* jeder Zeile. ASCII-Dateien (auch Binärformat möglich); Datei strukturiert in verschiedene Sektionen ähnlich wie bei CGM. Einbeziehung von nicht-geometrischen Elementen: "annotation"-Elemente (Maßhilfslinien etc.), "structure"-Elemente zur Spezifikation der Darstellung.

6. *STEP* (Standard for the Exchange of Product Model Data): neues CAD/CAE-Format, das IGES ablösen soll.

## 7. *WMF* (Windows Metafile Format)

Grafikformat von MS-Windows, Binärdateiformat; enthält Folge von Funktionsaufrufen für das Graphics Device Interface (GDI) von Windows.

Aufbau: Header + beliebige Anzahl von Records, die jeweils eine GDI-Funktion mit den dazugehörigen Parametern enthalten. Ansprechbare Zeichenobjekte: Pens, brushes, bitmaps.

Aufbau des Headers einer WMF-Datei:

1.–2. Byte	Typ
3.–4. Byte	Länge des Headers in words (Byte-Paaren)
5.–6. Byte	Versionsnummer
7.–10. Byte	Länge des Metafiles in words
11.–12. Byte	Zahl der Zeichenobjekte, die in dem Metafile angelegt werden
13.–16. Byte	Länge des längsten Records im Metafile
17.–18. Byte	bleiben leer

Aufbau eines Records:

1.–4. Byte	Länge des Records in words
5.–6. Byte	Nummer der aufgerufenen Funktion
ab 7. Byte	Record-Parameter, für jeden ein word

(nach Pohl & Eriksdottar 1991).

## 8. *Lotus-PIC*

binäres Grafikformat von Lotus 1-2-3 und Symphony. Nur für relativ einfache Linien-, Balken- und Kreisdiagramme konzipiert, daher einfacher Aufbau und leichte Editiermöglichkeiten trotz Binärformat.

## 9. VRML (Virtual Reality Modeling Language)

Sprache zur Beschreibung von (evtl. animierten) 3D-Objekten und -Szenen. Eröffnet Interaktionsmöglichkeiten u. Einbindung von Videos, Bilddaten, Klängen (Multimedialität).

ASCII-Dateien; plattformunabhängig und internetfähig (VRML-Browser als Plugins für Web-Browser).

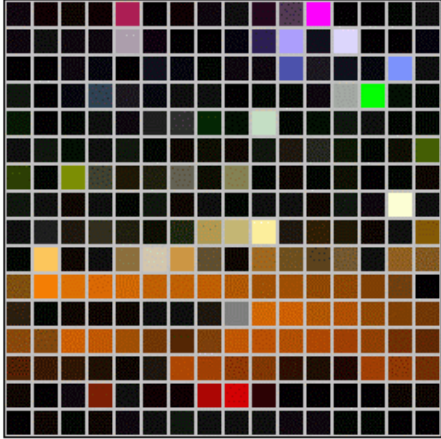
Dateistruktur: *Szenengraph*, bestehend aus Header, Kommentaren, Prototypen, *Knoten* und Routen (zur Ereignisbehandlung).

Die Knoten können Informationen enthalten über: Koordinaten, Materialdaten, Dimensionen von grafischen Primitiven (Quader, Kegel, Zylinder...), Farbe, Texturen, Mengen von Punkten, Lichtquellen, Viewpoint, Sensoren, Transformationen ganzer Gruppen anderer Knoten usw.; Scriptknoten als Schnittstelle zu externen Programmiersprachen.

- näheres später in der Übung -

## B. Rasterbasierte Dateiformate

Prinzipieller Aufbau einer Bilddatei in einem Rasterformat (kleinere Abwandlungen je nach gewähltem Format):

Bezeichnung	Repräsentanten	Kommentar / Beispiele
Header	Erkennungssequenz, Versionsnummer, Breite & Höhe & Farbtiefe, Quantisierungstabelle, Huffman Tabelle, Sonstiges (z.B. Kommentare vom Ersteller)	<b>GIF89a</b> .....3.3.f.....,.. .....(.....H .....: `.....L.....[...;  Der Code eines kompletten GIF-Bildes, der Version 89a.
Control Block	Zeitliche Verzögerung, Transparenzfarbe	
Palette		In diesem Block wird jedem vorkommenden RGB-Wert im Bild eine Nummer von 1 bis n zugeordnet. Mit diesen Schlüsseln wird dann der Bilddatenblock des Bildes gefüllt (siehe unten). In diesem Beispiel hat der Eintrag 1 den RGB-Wert (0,0,0). Dieses Verfahren wird nur bei 8-Bit (oder weniger Bit) Bildern angewendet, da bei 24-Bit Bildern, bei denen theoretisch das ganze RGB-Spektrum vorkommen könnte, keine Datenreduktion durch Palettenbildung stattfinden würde.
Bilddaten (Bitmap)	Entweder sind hier Verweise in die Palette, oder der RGB Code eines Pixels, oder die AC und DC Werte bei JPEG Bildern eingetragen.	Mit Palette: 5,7,4,10,14,40,...  Ohne Palette: (124,243,222), (222,123,3)

(aus Rechner 1999)

### 1. BMP (Bitmap)

binär codiertes Rasterdaten-Format mit Lauflängencodierung (RLE), weit verbreitet in der Windows-Welt. "Hausformat" von Windows und OS/2.

Nachteil: fast nur auf IBM-kompatiblen Rechnern in Gebrauch.  
Aufbau einer BMP-Datei: Header, Info-Teil, Daten.

## Aufbau von Header und Info:

<i>Header</i>	
2 Byte	"BM"
4 Byte	Datei-Länge in Byte
2 x 2 Byte	reserviert, müssen 0 sein
4 Byte	Offset Datenbereich
<i>Info</i>	
4 Byte	Länge in Byte
2 x 4 Byte	Breite / Höhe in Pixel
2 Byte	Farbtiefe
2 Byte	Bit/Pixel (1, 4, 8 oder 24, mit WinNT auch 32)
4 Byte	Kompressionstyp (ohne Kompr. = 0; 1 und 2: zwei verschiedene RLE-Typen)
4 Byte	Bildgröße in Byte
2 x 4 Byte	Auflösung horizontal / vertikal in Pixel/m
2 x 4 Byte	Zahl benutzte / wichtige Farben

Ab Offset 0x36 folgen Daten, und zwar als "RGB-Quad"-Struktur  $n \times 4$  Byte (jeweils 1 Byte Rot, Grün, Blau und "reserviert") oder in zwei alternativen Formaten.

(nach Henning 2000)

## 2. GIF (Graphics Interchange Format)

wurde 1987 von UNISYS und CompuServe definiert; große Verbreitung dank seiner Aufnahme in die HTML-Spezifikation. Binäres Rastergrafikformat, kann Bilder der max. Größe von 16000 x 16000 Pixel in max. 256 Farben darstellen. Farben in einer oder mehreren (lokalen) Farbtabelle spezifiziert.

In der Version GIF89a können auch Animationen niedergelegt werden (ein Bild wird nach einer einstellbaren Zeit durch ein anderes ersetzt, welches in der gleichen Datei steht; dafür "Graphics Control Extension Block" GCEExt vor jedem Rasterdaten-Block).

Kompression mit LZW (verlustfrei), je nach Bildtyp und -beschaffenheit Kompressionsraten zwischen 20 % und 95 %.

Struktur von GIF-Dateien: Header, Infos, global color map, Extension block, data, Terminations-Marker (näheres bei Henning 2000, S. 105 ff.).

Nachteile:

- nur 256 Farben darstellbar
- benutzt patentgeschützte Version des LZW-Verfahrens, daher Probleme bei Verwendung in freier Grafiksoftware: xv hat Unterstützung von GIF eingestellt wegen juristischer Probleme mit UNISYS. "Nachfolge-Format": PNG (noch nicht voll etabliert).

Empfehlung: Verwenden Sie dieses Format für die Darstellung von Logos und Zeichnungen mit weniger als 256 Farben im Internet.

### 3. *TIFF* (Tagged Image File Format)

binäres Bilddatenformat, entw. 1980 von der Aldus Corp. in Zusammenarbeit mit HP und Microsoft. Erlaubt verschiedene Farbtiefen und Komprimierungsalgorithmen. Komplizierter, aber zugleich universeller als andere Raster-Formate. Verwendung beim Fax.

Aufbau: geschachtelte Struktur.

Header, Infos, verkettete Liste von Image File Directories (IFDs) mit Zeigern auf die eigentlichen Daten. Zeiger sind enthalten in 12-Byte-"tag"-Strukturen, die auch Informationen über Größe u. Auflösung der Bilder enthalten.

Farbtiefe: 1–32 Bit RGB oder 1–48 Bit CMYK (Tiefe über 24 bzw. 32 für Tonwertkorrekturen).

Komprimierung: RLE, LZW, CCIT, Packed Bits (oder keine).

Empfehlung: mit LZW komprimieren, da so die Kompatibilität am höchsten. Raten zwischen 20 und 60 % (verlustfrei).

Vorteile:

- wird von fast allen Rechnerplattformen unterstützt
- verlustfreie Komprimierung
- kann in vielen Bildbearbeitungsprogrammen nachbearbeitet werden.

Nachteile:

- nicht alle Programme öffnen die komprimierten Versionen korrekt
- zu viele Subtypen
- relativ große Dateien i. Vgl. zu GIF und JPG
- wird nicht auf allen Plattformen im vollen Umfang unterstützt

Empfehlung: bei Verwendung zum Datenaustausch vorher klären, ob das Programm auf der Empfängerseite die Dateien fehlerfrei öffnen kann; ggf. die unkomprimierte Version wählen.

#### 4. *IFF* (Interchange File Format)

eingeführt von Electronic Arts 1985. Aufbau: Einteilung in einzelne *Chunks*, welche die Daten beinhalten.

Vorteil: Einbeziehung von Multimedia-Daten möglich (Video, Audio, MIDI).

Verwandtes Format:

*RIFF* (Resource Interchange File Format) von Microsoft.

#### 5. *PBM* (Public Bitmap)

ASCII-Format, entw. 1991 von Jeff Poskanzer. Keine Datenkompression, kein Alphakanal, sehr große Dateien. Auf allen Systemen kostenlos erhältlich.

3 Varianten:

- PBM: Bitmap für s/w-Bilder
- PGM: Graymap für Graustufenbilder
- PPM: Pixmap (full color) für True Color-Bilder (24 Bit Farbtiefe).

Für jedes der 3 Grundformate gibt es auch die Möglichkeit der binären Codierung.

Datenstruktur für PBM-Dateien:

<i>Zeile</i>	<i>Inhalt</i>
1	Text "P1"   "P2"   "P3"   "P4"   "P5"   "P6"
2	Breite, Höhe in Pixel
3	Farbtiefe in Bit (außer bei s/w-Bildern)
ab 4	Daten ASCII bei P1,...P3; binär bei P4,..., P6. ASCII darf beliebig viele Leerzeichen beinhalten.

6. *XBM* (X Bitmap) und *XPM* (X Pixmap)  
Grafikdatenformate aus dem unter Unix verbreiteten Fenstersystem X11. Verwendung vor allem für Cursor- und Icon-Bitmaps.  
Aufbau: C-Quellcode (Konstantendefinitionen und Deklarationen).

7. *PNG* (Portable Network Graphics)  
propagiertes "Nachfolge-Format" von GIF, LZW-ähnliche Kompression, max. 48 Bit Farbtiefe. Entw. von Boutell & Lane 1997, lizenzfrei.

8. *JFIF* (JPEG-File Interchange Format),

besser bekannt unter den Kürzeln JPEG (Joint Photographers Expert Group) oder JPG, die aber eigentlich die Organisation abkürzen. Entw. 1991–1993, standardisiert von der CCITT (ISO-10918).



Universelles Austauschformat zwischen fast allen Plattformen und Anwendungen, Verbreitung im Web.

Einsatz: Weitergabe von großen Bilddateien, insbes. Fotos. Hohe Kompressionsraten, jedoch oft mit Informationsverlust erkaufte.

### JPEG-Kompressionsverfahren

(werden für jeden der 3 Farbkanäle separat durchgeführt):

#### 1. Verlustfreie Variante

Differenzverfahren: die einzelnen Pixel werden aus Nachbarpixeln vorhergesagt nach 1 von 7 möglichen Algorithmen.

Auswahl des Algorithmus, welcher die beste Vorhersage liefert. Ausgabe-Datenstrom: Nummer des gewählten Algorithmus und Differenz zwischen Vorhersage- u. tatsächl. Datenwert.

⇒ Differenzwerte häufig 0 ⇒ effiziente Kompression mittels Huffman-Codierung oder einer arithm. Codierung.

Einsparungen bis ca. 50 %.

#### 2. Verlustbehaftete Variante

Verfahren in mehreren Schritten.

Grundidee: spezielle Fourier-Transformation (diskrete Kosinus-Transformation), d.h. Übersetzung der Bildinformation in den Frequenzraum. Anschließende Quantisierung (Rundung der Fourier-Koeffizienten) ⇒ Unterdrückung der hochfrequenten (feineren) Details des Bildes. Visuelle Wahrnehmung (bes. bei Internet-Seiten) "merkt" diese Vergrößerung nicht.

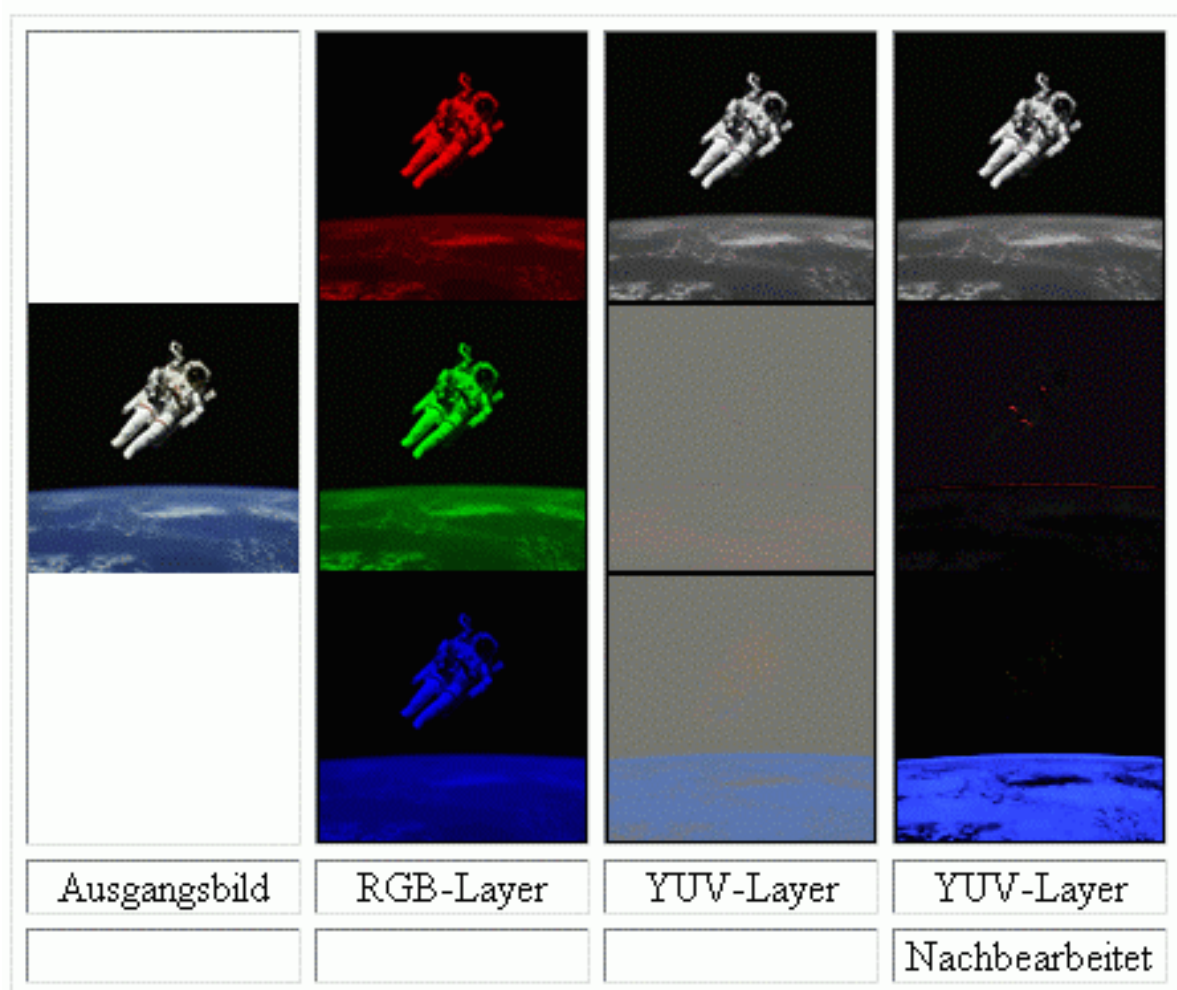
Kompressionsgrad (bzw. Qualitätsgrad) kann vom Benutzer vorgegeben werden.

Schritte bei der verlustbehafteten JPEG-Kompression:

(a) Wechsel des Farbmodells von RGB nach YIQ, dabei Farb-Subsampling (s.o.). Ergebnis: 3 Teilbilder ("Scans").

Grund für Wechsel der Farben-Basis: Helligkeitsinformation (Luminanz) steckt im RGB-Modell in jeder Komponente, im YIQ- (oder im YUV-) Modell nur in der Y-Komponente. Diese wird mit voller Auflösung gespeichert, die anderen beiden Komponenten nur mit halber Auflösung.

Ferner: Regionen mit gleicher Helligkeit (Y) sind häufiger als Regionen mit gleichen RGB-Werten  $\Rightarrow$  stärkere Kompression möglich.



(nach Rechner 1999)

Jeder Scan wird jetzt separat weiterbehandelt.

(b) Transformation mit DCT (diskrete Kosinus-Transformation)

bijektive Abbildung in den Frequenzraum (verlustfrei).

Bildinformation als Überlagerung von Kosinus-Wellen, gespeichert werden die Koeffizienten der Frequenz-Anteile.

1. Schritt: Zerlegung in 8 x 8 - Pixelblöcke. (größere Blöcke zu rechenzeitintensiv.)

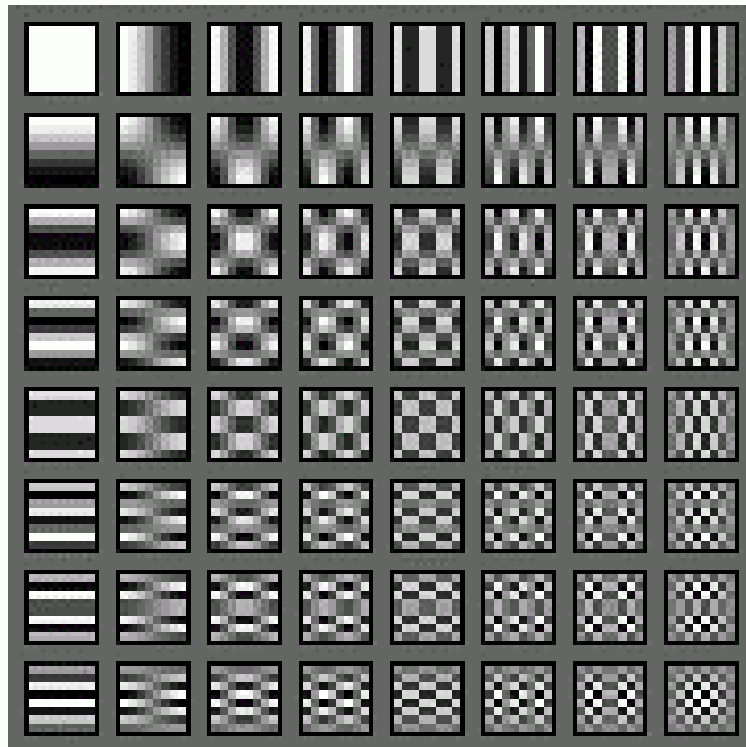
2. Schritt: Anwendung der DCT auf jeden Block.

Seien  $i$  und  $j$  die Positionen der Helligkeitswerte  $f(i, j)$  im betrachteten Block;  $u, v$  die Positionen der Werte  $F(u, v)$  in der Zielbitmap (die Frequenzen steigen mit  $u$  und  $v$ ):

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

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

Ergebnis: Überlagerung von 64 Grundfunktionen, die jeweils verschieden stark gewichtet werden. 8 x 8 - Block von Koeffizienten für diese Grundfunktionen.



Kästchen (0; 0): niederfrequentester Anteil ("Gleichstrom"; "DC coefficient"), hier wird der Mittelwert der Block-Helligkeit abgelegt.

x-Richtung: höhere Frequ. der horizontal überlagerten Kosinus-Welle, y-Richtung: entspr. für vertikale Welle. Koeffizienten der Felder ungleich (0; 0): "AC coefficients". Sind alle 0 bei homogenen Blöcken  $\Rightarrow$  hohe Kompression möglich.

(c) Quantisierung (Löschung von Information!)

*Ganzzahlige* Division der Koeffizienten  $F(u, v)$  durch vorgegebene Zahl  $q(u, v)$  (für alle gleich oder aus Tabelle, die mitübertragen wird):

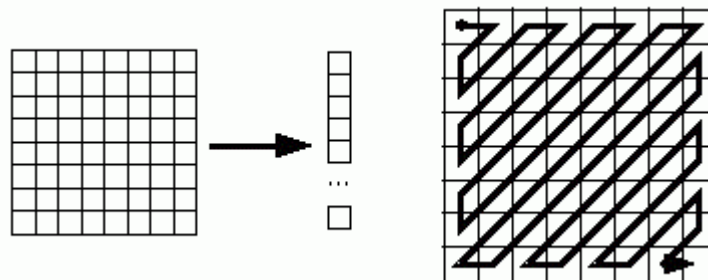
$$F'(u, v) = \text{int} (F(u, v) / q(u, v))$$

dadurch kontrollierte Beseitigung hochfrequenter Anteile.

(d) Codierung mit erneuter verlustfreier Kompression

für die DC-Koeffizienten: Differenzverfahren (s.o., verlustfreie Variante der JPEG-Kompression)

für die AC-Koeffizienten:  
Zick-Zack-Serialisierung,



Ergebnis: 1 x 64 - Vektor, hochfreq. Anteile unten;





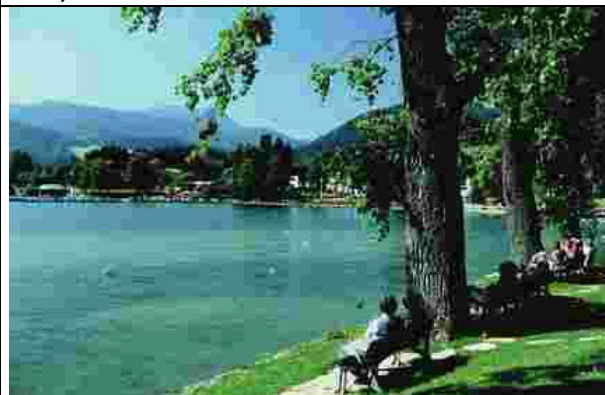
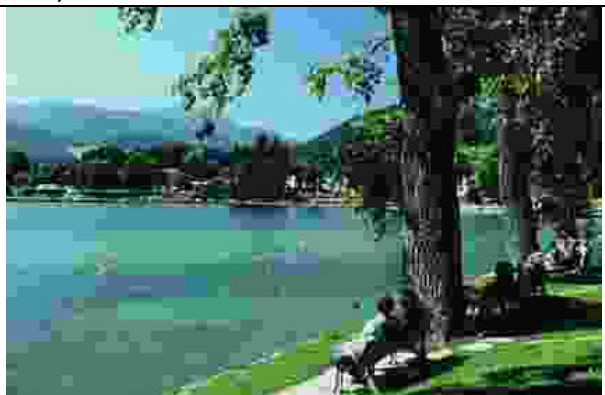
anschließende Lauflängen-Codierung, gefolgt von statistikbasierter Codierung (Huffman oder arithmetisch).

Progressiver Modus (approximative Übertragung mit sukzessiver Verfeinerung, wichtig beim Aufbau von Webseiten):

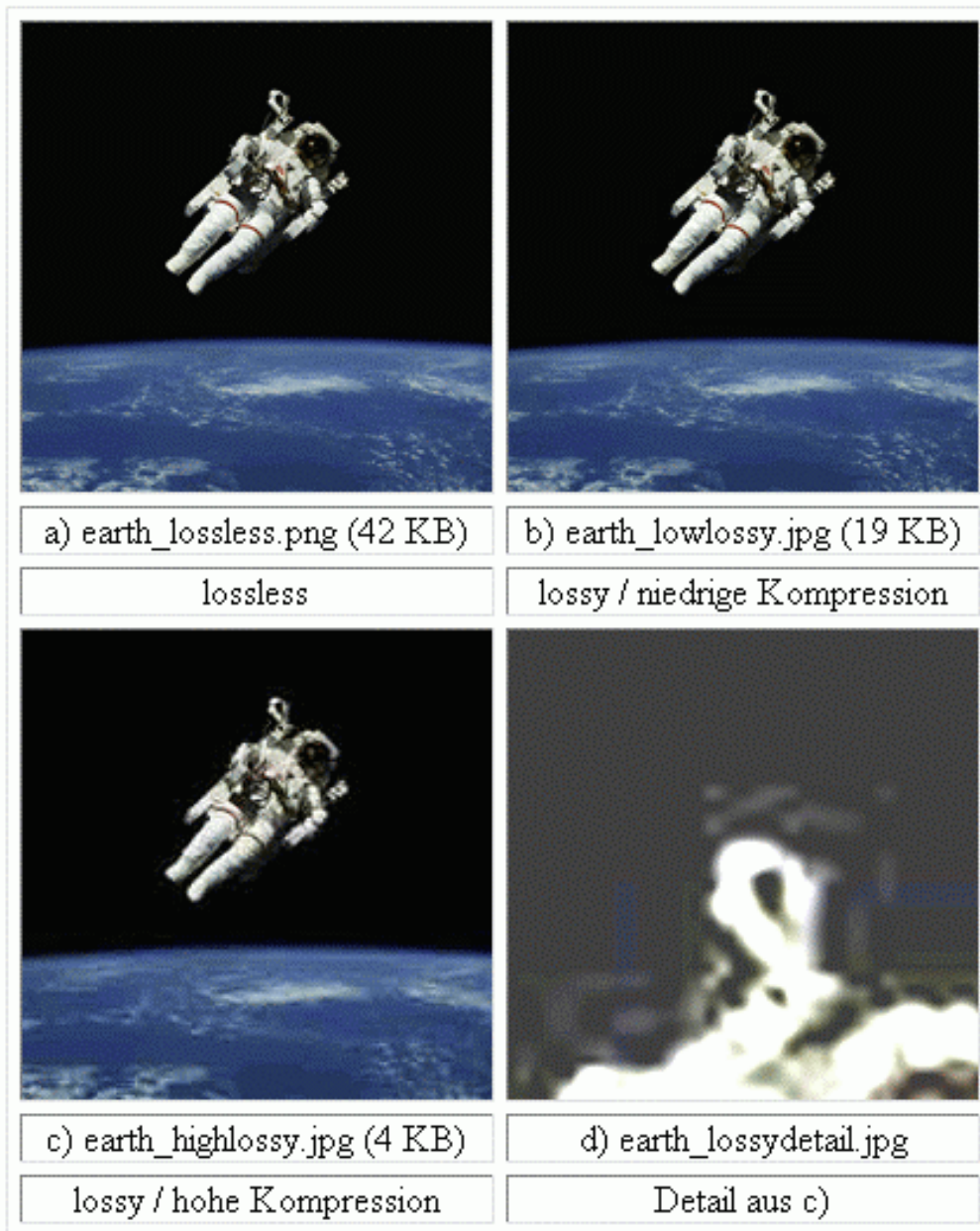
zuerst die DC-Koeffizienten ( $\Rightarrow$  8 x 8 - Blockgrafik), dann die niederfrequenten AC-Koeffizienten, dann die hochfrequenten.

## Anwendungsbeispiel:

Originalabbildung: Screenshot von SGI-Workstation (RGB-Format, Originaldatei 352 K, war aber bereits ein Bild aus dem Web).

	
JPEG, verlustfrei komprimiert: 123 K	JPEG, Qualitätsfaktor 75 %: 30,6 K
	
JPEG, Qual.faktor 50 %: 20,7 K	JPEG, Qual.faktor 25 %: 13,3 K
	
JPEG, Qual.faktor 10 %: 7,4 K	JPEG, Qual.faktor 5 %: 4,9 K

weiteres Beispiel:



Detail-Ausschnitt zeigt "Halos" im dunklen Weltraum.  
(aus Rechner 1999).

Verlustbehaftete JPEG-Kompression gut geeignet für Fotos, besonders bei Verwendung fürs Internet (wo Darstellung auf dem Bildschirm sowieso Details verkleinert) und für realistische Farbdarstellung.

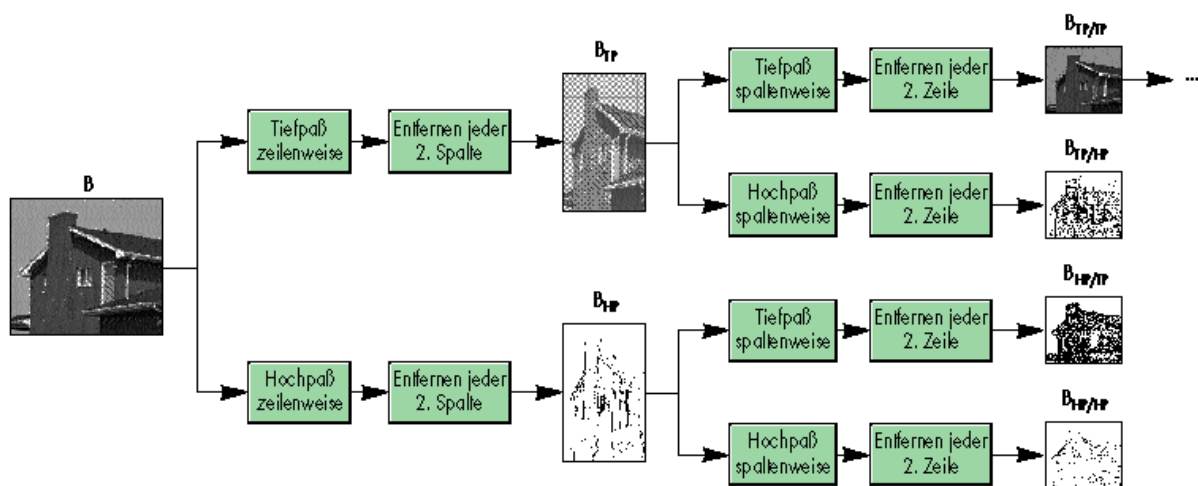
Schwächen bei Bildern mit scharfen Kontrasten und Kanten, bei künstlichen Bildern, Texten, detail-sensitiven Daten (z.B. medizinische Bilddaten).

Dateiformat JFIF (selbes für verlustfreie und verlustbehaftete JPEG-Kompression):

Header, dann die 3 Scans hintereinander (jeweils mit Scan-Header), Blöcke: entsprechen den 8 x 8 - Makroblöcken. Näheres bei Henning 2000, S. 119.

neues Dateiformat, soll evtl. JFIF ablösen: *SPIFF*.

neues Kompressionsverfahren: *Wavelet-Zerlegung* anstelle der DCT ( $\Rightarrow$  höhere Bildqualität). Wavelets: verschiedene Wellenformen anstelle von Sinus/Kosinus. Originalbild wird in Wavelet-Koeffizienten umgewandelt. D.h.: hoch- und tiefpassgefilterte Anteile, die weiter zerlegt werden können.



(aus Pichler 1998).

# Übersicht zu neueren Grafikformaten zur Darstellung von Grafiken im WWW (aus Ansorg 2001):

## Optimale Darstellung von Grafiken im WWW

WWW-Grafikformate					
	GIF87a	GIF89a	JPEG	P-JPEG	PNG
Farbtiefe	1-8 Bit	1-8 Bit	24 Bit	24 Bit	8,16,24,48 Bit
Farbcodierung	Farbpalette	Farbpalette	Direkt RGB	Direkt RGB	Farbpalette o. Direkt RGB
max. Farbanzahl	256	256	16,8 Mio.	16,8 Mio.	16,8 Mio.
Farbspektrum	16,8 Mio.	16,8 Mio.	16,8 Mio.	16,8 Mio.	16,8 Mio.
Gamma Informationen	nein	nein	nein	nein	ja
Fehlererkennung	nein	nein	nein	nein	ja
Kompressionsmethode	LZW	LZW	DCT und Huffman-Cod.	DCT und Huffman-Cod.	LZ77-Deflate
Kompressionsrate	3:1-5:1	3:1-5:1	10:1-50:1	10:1-50:1	4:1-7:1
Verlustfreie Kompression	ja	ja	nein	nein	ja
Interlacing	ja (horiz.)	ja (horiz.)	nein	ja (2-dim.)	ja (2-dim.)
Transparentfarbe	nein	ja	nein	nein	ja
Mehrere Einzelgrafiken	ja	ja	nein	nein	nein
Erweiterbarkeit	ja	nein	nein	nein	ja
Alpha-Kanäle	nein	nein	nein	nein	ja
Bitmap Ebenen	nein	nein	nein	nein	ja
Offene Spezifikation	ja	ja	ja	ja	ja
Referenzimplementierung	nein	nein	ja	ja	ja
Lizenzpflichtig	ja	ja	nein	nein	nein

Offene Spezifikation	ja	ja	ja	ja	ja
Referenzimplementierung	nein	nein	ja	ja	ja
Lizenzpflichtig	ja	ja	nein	nein	nein
Direkte Unterstützung Netscape Navigator 3.0	ja	ja	ja	ja	nein
Direkte Unterstützung Microsoft Explorer 3.0	ja	ja	ja	ja, aber ohne optisch. Effekt	nein
Direkte Unterstützung Netscape Navigator 4.04	ja	ja	ja	ja	ja
Direkte Unterstützung Microsoft Explorer 4.0	ja	ja	ja	ja	ja



Es gibt weitere rasterbasierte Grafikformate, auf die hier nicht näher eingegangen werden soll:

- RAS (Sun Raster-Format)
- RGB (SGI Bilddatenformat)
- MAC (Mac-Paint-Format von Apple)
- PCX (ZSoft; alter PC-Standard für Paintbrush-Software)
- MSP (Format von Microsoft Paint, nicht mehr unterstützt)
- IMG (unter GEM)
- WPG (Word Perfect Graphics Format)
- ...

#### Konvertierungssoftware:

- auf PC: IrfanView (iView)
- auf Unix/Linux: xv
- auf Unix: alchemy
- ...