Kapitel 10

Grafik-File-Formate

Für den Austausch von Bildinformation sind standardisierte Grafik-File-Formate sinnvoll und erforderlich. Aus der großen Menge von verschiedenen Grafik-File-Formaten werden hier einige wenige vorgestellt.

10.1 PBM, PGM, PNM und PPM

Die *Portable Bitmap Utilities* (PBM, auch pbmplus oder netpbm) sind eine Sammlung freierhältlicher Programme, die von Jef Poskanzer gepflegt werden. Es handelt sich eigentlich um drei Programmpakete, die sich mit unterschiedlichen Bildarten und File-Formaten befassen:

- Die Portable Bitmap Utilies (PBM) manipulieren monochrome Bilder.
- Die Portable Greymap Utilies (PGM) manipulieren Grauwert-Bilder.
- Die Portable Pixmap Utilies (PPM) manipulieren Farbbilder.

Die Portable Anymap Utilies (PNM) arbeiten auf allen von den drei Programmpaketen erzeugten Bilddateien. Für PNM gibt es kein eigenes Dateiformat.
Die PBM-, PGM- und PPM-File-Formate sind so einfach wie möglich gehalten. Sie starten jeweils mit einem Header, dem die Bildinformation unmittelbar folgt. Der Header ist immer in ASCII geschrieben, wobei die einzelnen Einträge durch White Spaces getrennt werden. Die Bildinformation kann entweder im ASCII- oder Binärformat sein.
Es gibt jeweils zwei Header-Versionen, von denen eine für ASCII- und eine für binäre Bildinformationen benutzt wird.

PBM-Header

Ein PBM-Header besteht aus folgenden Einträgen, jeweils durch White Spaces getrennt:

Magic Value	`P1` für ASCII-, `P4` für binäre Bildinformation
Image Width	Breite des Bildes in Pixeln (ASCII-Dezimalwert)
Image Height	Höhe des Bildes in Pixeln (ASCII-Dezimalwert)

PGM-Header

Ein PGM-Header besteht aus folgenden Einträgen, jeweils durch White Spaces getrennt:

Magic Value	`P2` für ASCII-, `P5` für binäre Bildinformation
Image Width	Breite des Bildes in Pixeln (ASCII-Dezimalwert)
Image Height	Höhe des Bildes in Pixeln (ASCII-Dezimalwert)
Max Grey	Maximaler Grauwert (ASCII-Dezimalwert)

PPM-Header

Ein PPM-Header besteht aus folgenden Einträgen, jeweils durch White Spaces getrennt:

Magic Value	`P3` für ASCII-, `P6` für binäre Bildinformation
Image Width	Breite des Bildes in Pixeln (ASCII-Dezimalwert)
Image Height	Höhe des Bildes in Pixeln (ASCII-Dezimalwert)
Max Color	Maximaler Farbwert (ASCII-Dezimalwert)

Bildinformation in ASCII

Nach dem Header folgt die Beschreibung der Breite × Höhe vielen Pixel, beginnend in der linken oberen Bildecke zeilenweise von links nach rechts.
Bei PPM besteht jedes Pixel aus drei ASCII-Dezimalwerten zwischen 0 und dem angegebenen maximalen Farbwert, die die Rot-, Grün- und Blauwerte des Pixels darstellen.
Bei PBM und PGM gibt es nur einen ASCII-Dezimalwert pro Pixel. Bei PBM ist der maximale Wert implizit 1; die Werte müssen nicht durch White Spaces getrennt werden.

Beispiel:

Das folgende Beispiel stellt ein 4 × 4-Pixmap im ASCII-Format dar:

```
P3
# feep.ppm
4 4
15
 0  0  0    0  0  0    0  0  0   15  0 15
 0  0  0    0 15  7    0  0  0    0  0  0
 0  0  0    0  0  0    0 15  7    0  0  0
15  0 15    0  0  0    0  0  0    0  0  0
```

Wie zu sehen, können auch Kommentare in ein PNM-File eingefügt werden. Zeichen ab einem `#` bis zum Zeilenende werden ignoriert.

Binäre Bildinformation

Deutlich kleinere Dateien, die schneller gelesen und geschrieben werden können, werden mit Hilfe der binären Bilddarstellung erreicht.
Die Pixelwerte werden dabei wie folgt abgelegt:

Format	Magic Value	binäre Darstellung
PBM	`P4`	acht Pixel pro Byte
PGM	`P5`	ein Pixel pro Byte
PPM	`P6`	drei Bytes pro Pixel

Beim binären Format ist nach dem Header nur noch ein einziger White Space (typischerweise Newline) erlaubt. Innerhalb der binären Bilddaten sind keine White Spaces erlaubt. Die Bitorder innerhalb der Bytes ist "most significant bit first".
Beachtenswert ist, daß für das Binärformat nur maximale Werte bis 255 benutzt werden können.

Konvertierungsroutinen

Für die Umwandlung vom bzw. in das PNM-Format steht eine ganze Reihe von Routinen zur Verfügung, z.B.:
`anytopnm`, `asciitopgm`, `bmptoppm`, `giftopnm`, `pbmtopgm`, `pgmtopbm`, `pgmtoppm`, `ppmtopgm`, `pstopnm`, `rgb3toppm`, `tifftopnm`, `xbmtopbm`, `xwdtopnm`, `pbmtoascii`, `pbmtolps`, `pbmtopgm`, `pbmtoxbm`, `pgmtopbm`, `pgmtoppm`, `pnmtops`, `pnmtotiff`, `pnmtoxwd`, `ppmtobmp`, `ppmtogif`, `ppmtopgm`, `ppmtorgb3`.

10.2 GIF

Das *Graphics Interchange Format* (GIF) wurde von CompuServe eingeführt und wird benutzt, um ggf. mehrere Bilder in einer einzigen Datei zu speichern und zwischen verschiedenen Rechnersystemen auszutauschen. Bezogen auf die Anzahl der existierenden Dateien ist GIF eines der weitverbreitetsten Formate zum Speichern von Bilddaten.
Im Gegensatz zu vielen anderen Dateiformaten basiert GIF auf einem Strom der Daten. Das Format besteht aus einer Reihe von Datenpaketen, Blöcke genannt, kombiniert mit zusätzlicher Protokollinformation.
Das GIF-Format ist in der Lage, Bilddaten mit einer Farbtiefe von 1 bis 8 Bits zu speichern. Die Bilder werden immer im RGB-Modell unter Benutzung einer Farbtabelle gespeichert.
Die Bildinformation in GIF-Dateien ist immer LZW-komprimiert. Diese von Lempel und Ziv 1977 eingeführte sowie 1984 von Welch verbesserte Datenkompression wird häufig und nicht nur für Bilddaten verwendet. Der Algorithmus reduziert Folgen von gleichen Bytewerten in ein einzelnes Codewort und erreicht auf diese Weise für typische 8-Bit-Pixel-Daten eine Reduktion der Datenmenge um 40%.
GIF gibt es in zwei Versionen: GIF87a und GIF89a.

GIF87a Format

Header
Logical Screen Descriptor
Global Color Table
Local Image Descriptor for Image 1
Local Color Table for Image 1
Image Data for Image 1
Local Image Descriptor for Image 2
Local Color Table for Image 2
Image Data for Image 2
...
Trailer

GIF89a Format

Header
Logical Screen Descriptor
Global Color Table
Comment Extension
Application Extension
Graphics Control Extension
Local Image Descriptor for Image 1
Local Color Table for Image 1
Image Data for Image 1
...
Comment Extension
Plain Text Extension
Trailer

10.3 JPEG

Die **J**oint **P**hotographic **E**xperts **G**roup bildete sich aus Mitgliedern der Standardisierungsgremien CCITT (Consultative Committee for International Telephone and Telegraph) und ISO (International Standardization Organization). JPEG wurde schließlich zum Namen für den Standard selbst.
Zunächst wird das RGB-Bild in den YUV-Raum transformiert. Da das Auge für Helligkeitssprünge sensitiver ist als für Farbdifferenzen, kann man nun die Y-Pixel-Matrix in der vollen Auflösung belassen und in den U- und V-Matrizen jeweils 4 Pixel mitteln (4:2:2 Subsampling). Für je 4 Originalpixel mit insgesamt 12 Bytes werden nun $4 + 1 + 1 = 6$ Bytes benötigt (pro Bildpunkt also $6 \cdot 8/4 = 12$ Bit). Die Reduktion beträgt 50 %.
Werden je 16 Originalpixel bzgl. der Farbwerte gemittelt, so werden für 16 Originalpixel statt 48 Bytes nur noch $16 + 1 + 1 = 18$ Bytes benötigt (pro Bildpunkt also $18 \cdot 8/16 = 9$ Bit). Die Bilddatei schrumpft auf 37.5 %.
Beim JPEG-Verfahren werden die drei Matrizen in Blöcke mit 8×8 Abtastwerten aufgeteilt. Anschließend durchlaufen die Blöcke folgende Schritte:

1. Diskrete Cosinus Transformation,
2. Rundung der Frequenzkoeffizienten,
3. Lauflängencodierung der quantisierten Werte,
4. Huffmancodierung der Lauflängenbeschreibung.

Um aus dem komprimierten Bild das Original zu rekonstruieren, werden die Schritte in umgekehrter Reihenfolge und inverser Funktionalität durchlaufen.
Durch die Wahl der Rundungstabelle läßt sich der Tradeoff zwischen Qualität und Kompression beliebig steuern. Ein typisches Farbbild läßt sich ohne für das Auge sichtbare Artefakte auf 10 % seiner Originalgröße reduzieren. Eine Reduktion auf 5 % verursacht oft nur leichte, kaum wahrnehmbare Verzerrungen.

10.4 Erzeugung einer Farbtabelle

Zunächst wird für jede Farbe die Häufigkeit ihres Auftretens ermittelt. Hierdurch erhält man ein Histogramm.
Sei $d(a,b)$ der Abstand zweier Farbtupel a, b,
z.B. $d(a,b) = \sqrt{(a_{Rot} - b_{Rot})^2 + (a_{Grün} - b_{Grün})^2 + (a_{Blau} - b_{Blau})^2}$.

Optimaler Algorithmus

Gegeben sei eine Menge F von beobachteten Farben. Gesucht ist eine Menge M von k Repräsentanten, so daß der Ausdruck

$$\Delta := \max_{x \in F} \quad \min_{p \in M} d(p,x)$$

minimiert wird, d.h., Δ ist der maximale Abstand, den eine Farbe zu ihrem nächsten Repräsentanten hat.

Da die exakte Bestimmung von M eine exponentielle Laufzeit verursacht, begnügt man sich mit einer Näherungslösung.

Popularity-Algorithmus (1978)

Wähle die k häufigsten Farben. Nachteil: Selten vorkommende Farben werden schlecht repräsentiert.

Diversity-Algorithmus (*xv*, John Bradley, 1989)

Initialisiere M mit der häufigsten Farbe.

```
for i := 2 to k do
   erweitere M um die Farbe des Histogramms,
   die zu allen Farben aus M den groessten Abstand hat
end
```

Median-Cut (Heckbert, MIT 1980)

```
Initialisiere RGB-Wuerfel mit Haeufigkeiten der beobachteten Farbtupel
Initialisiere Wurzel des Schnittbaums mit Gesamtzahl der Pixel

While noch_nicht_genuegend_Blaetter do
   Waehle Blatt mit der groessten Pixelzahl
   Bestimme umschliessende Box
   Bestimme Achse mit groesstem Wertebereich
   Durchlaufe Box laengs dieser Achse
   Teile am Median in zwei Haelften
   Trage Haelften als Soehne ein
end
Fuer jedes Blatt waehle den Mittelwert aller in ihm liegenden Farben.
```

Der Median-Cut-Algorithmus von Heckbert wird verwandt im Programm `ppmquant`, das Teil des PBM-Pakets ist.