Dieses Kapitel stammt aus dem Buch

›Dateiformate – Eine Referenz‹

von Günter Born.

www.borncity.de

ISBN 3-934358-83-7
119,90 DM

Informationen zum Buch
mit Bestellmöglichkeit

www.galileocomputing.de

Galileo Computing

35 Das Portable Network Graphics-Format (PNG)

Dies ist ebenfalls ein Dateiformat zum Speichern von Bitmap-Grafiken. Es wurde definiert, um das von CompuServe definierte GIF zu ersetzen. Der Grund liegt darin, daß der von der GIF-Spezifikation benutzte LZW-Komprimierungsalgorithmus von der Firma Unisys patentiert wurde. Firmen, die den LZW-Algorithmus in Bibliotheken benutzen, müssen Lizenzgebühren an den Patenthalter abführen.

Das PNG-Format unterstützt Bilder mit bis zu 16 Bit pro Bildpunkt für Graustufen und mit bis zu 48 Bit für Farbbilder. Zusätzlich können die Bilddateien 16 Bit für einen Alphakanal aufweisen. Bei Graustufenbildern reichen die Werte für einen Bildpunkt von 0 (steht für einen schwarzen Punkt) bis zum maximalen Intensitätswert. Dieser maximale Wert wird durch 2**Bit per Pixel-1 spezifiziert (z. B. 28 ergibt den Wert 256, was 8 Bit pro Pixel entspricht – folglich werden Werte zwischen 0 und 255 für die Bildpunkte verwendet). Ein Bildpunkt in einem farbigen Bild kann entweder über eine Farbpalette oder als RGB-Wert (mit den Farbanteilen rot, grün und blau) definiert sein. Bei RGB-Bildern können die Farbwerte kalibriert oder unkalibriert vorliegen. (Die Kalibrierung der Farbwerte erfolgt über den *cHRM*-CHUNK.) Beachten Sie auch, daß die Farbwerte nicht unbedingt linear sein müssen. Über den *gAMA*-CHUNK läßt sich die Charakteristik des Viewers und der Source-Device festlegen.

Das Bild wird als rechteckige Pixelmap gescannt und gespeichert, wobei der Nullpunkt in der linken oberen Ecke liegt. Die Zeilen werden dann von links nach rechts ausgegeben. Hierbei lassen sich die Bildzeilen kontinuierlich oder interlaced speichern. Alle 16-Bit-Werte (Integer) werden im *Big-endian*-Format gespeichert, d. h., das oberste Byte (MSB) wird auf der niederwertigen Adresse gespeichert. Auf Intel-Plattformen müssen die beiden Bytes eines Worts getauscht werden, um einen Integerwert zu erhalten.

Der Aufbau der PNG-Datei

Eine PNG-Datei besitzt einen blockweisen Aufbau, der aus sogenannten CHUNKs besteht (siehe auch IFF oder WAV). Die Zahl der CHUNKs ist dabei abhängig vom gespeicherten Bild.

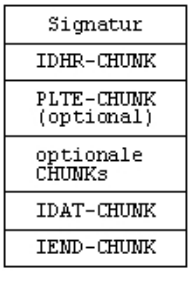

Abbildung 35.1 Struktur einer PNG-Datei

Die Datei beginnt mit einer Signatur (8 Byte), an die sich mindestens drei CHUNKs (IDHR, IDAT und IEND) anschließen. Darüber hinaus gibt es optionale CHUNKs (z. B. PLTE für Palettendaten), die in der Datei gespeichert werden können.

Die Signatur

Die ersten 8 Bytes einer PNG-Datei enthalten die Signatur 89H, 50H, 4EH, 47H, 0DH, 0AH, 1AH, 0AH. In dieser Signatur ist die Zeichenfolge PNG enthalten, und gleichzeitig bedeutet diese Signatur, daß die Datei ein einzelnes PNG-Bild enthält, das zwischen den CHUNKs IHDR und IEND eingebettet ist.

Die Struktur eines CHUNK

Die sich an den Header anschließenden CHUNKs besitzen die in folgender Tabelle gezeigte Struktur:

Offset	Bytes	Bedeutung
00H	4	Länge des Datenbereichs in Byte
04H	4	CHUNK-Typ
08H	n	Datenbereich des CHUNK
..H	4	CRC-Wert des Datenbereichs

Tabelle 35.1 Struktur eines CHUNKs

Das erste DWORD gibt die Länge des Datenbereichs des CHUNKs an. Daran schließen sich vier Bytes mit einer Signatur für den CHUNK an. Diese Signatur besteht aus den ASCII-Codes der Buchstaben des CHUNK-Namens (z. B. cHRM, IHDR, gAMA etc.). Die vier Bytes mit dem Namen des CHUNK enthalten zusätzlich einige Hinweise auf die Eigenschaften des CHUNK. Hierbei wird das Bit 5 benutzt, um im Namen zwischen Groß- und Kleinbuchstaben umzuschalten (der PNG-Reader liest aber nur Bytes). Je nach der Vergabe der Groß- und Kleinbuchstaben im Namen werden die Eigenschaften des CHUNK festgelegt. Enthält das erste Byte des CHUNK-Namens einen Großbuchstaben, handelt es sich um einen kritischen CHUNK (der von einem PNG-Programm unterstützt werden muß). Ein Kleinbuchstabe im Anfang des CHUNK-Namens weist auf einen untergeordneten (*ancillary*) CHUNK mit optionalen Informationen hin. Das zweite Zeichen kann ebenfalls als Groß- oder Kleinbuchstabe ausgeführt werden. Bei einem Großbuchstaben (dann ist Bit 5 = 0 gesetzt) handelt es sich um einen öffentlichen CHUNK, der ein Teil der PNG-Spezifikation ist. Eine Anwendung kann auch private (nicht registrierte) CHUNKs erzeugen. Dann ist der zweite Buchstabe im CHUNK-Namen mit einem Kleinbuchstaben zu kodieren. Der dritte Buchstabe muß z. Z. immer groß geschrieben werden, da Bit 5 in diesem Byte reserviert ist. Im letzten Buchstaben des Namens wird die *Safe to copy*-Option festgelegt. Ist das Bit 5 = 1 (Kleinbuchstabe), kann der CHUNK ko-

piert werden. Dies ist für PNG-Editoren hilfreich. Wird das Bit auf 0 gesetzt, muß der CHUNK im Editor implementiert sein, um die PNG-Daten korrekt für Änderungen zu interpretieren.

Die Daten des CHUNK folgen ab Offset 08H. Die letzten 4 Bytes eines CHUNK enthalten den CRC-32-Wert für den Datenbereich und das Feld mit dem CHUNK-Typ. (Das PNG-Format benutzt dabei das in der ISO 3309-Norm beschriebene Polygon $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ zur Berechnung des CRC-Werts.)

Kritische CHUNKs

Die PNG-Definition unterscheidet zwischen kritischen CHUNKs und untergeordneten (*ancillary*) CHUNKs. Die kritischen CHUNKs müssen von jedem Programm beherrscht werden, das eine PNG-Datei liest oder schreibt. (Diese kritischen CHUNKs beginnen mit einem Großbuchstaben im CHUNK-Namen. Untergeordnete (ancillary) CHUNKs beginnen dagegen mit einem kleinen Buchstaben im Namen wie z. B. cHRM.) Nachfolgend finden Sie die Beschreibung der kritischen CHUNKs.

Header-CHUNK (IHDR)

Der Header-CHUNK enthält Informationen über die Daten, die in der PNG-Datei gespeichert werden. Der CHUNK muß sofort nach den 8 Bytes der Signatur auftreten. Als Signatur wird der Name IHDR benutzt. Der Datenbereich des CHUNK besitzt den in Tabelle 35.2 gezeigten Aufbau.

Offset	Bytes	Bedeutung
00H	4	Breite des Bildes in Pixel
04H	4	Höhe des Bildes in Pixel
08H	1	Bit pro Pixel (oder pro Sample)
09H	1	Color Typ
0AH	1	Compression-Typ
0BH	1	Filter Typ
0CH	1	Interlace-Flag

Tabelle 35.2 Der Datenbereich des IHDR-CHUNK

Die ersten beiden DWORD-Einträge im Datenbereich des CHUNK geben die Abmessungen des Bildes in Pixel wieder. Danach folgt ab Offset 08H ein Byte, das die Zahl der Bits pro Sample festlegt. Bei Graustufenbildern sind die Werte 1, 2, 4, 8 und 16 zulässig. Bei Farbbildern, die auf einer Farbpalette basieren, sind 1, 2, 4 und 8 Bit pro Sample (und damit pro Bildpunkt) zugelassen. Bei True Color-Bildern können Sie dagegen 8 oder 16 Bit pro Sample (jedes Sample entspricht dann einem Farb- oder Alphakanal) hinterlegen.

Der *Color Typ* in Offset 09H gibt an, wie die Bildpunkte zu interpretieren sind. Hierbei wird eine bitweise Kodierung benutzt:

Bit	Bemerkung
0	1: eine Palette wird benutzt
1	1: Farbbild
2	1: Alphakanal wird benutzt

Tabelle 35.3 Kodierung des Farbtyps

Hierbei lassen sich mehrere Bits kombinieren. Ein Wert von 0 bedeutet zum Beispiel, daß ein Graustufenbild vorliegt. Der Wert 2 steht für ein TrueColor-Bild (ohne Palette), und der Wert 3 weist auf ein Farbbild mit einer Palette hin. (Es sind die Werte 0, 2, 3, 4 und 6 gültig.)

Die Bilddaten können komprimiert vorliegen. Dies wird durch ein Byte ab Offset 0AH im Datenbereich angezeigt. Zur Zeit ist nur der Wert 0 vorgesehen, der für die Deflate/Inflate-Komprimierung (mit 32 kByte Sliding Window) steht.

Die Filtermethode gibt an, wie die Daten vor der Komprimierung zu filtern sind. Zur Zeit wird nur der Wert 0 unterstützt (adaptive Filterung mit den fünf Basisfiltertypen).

Das letzte Byte im Datenbereich gibt an, ob die Bildzeilen interlaced gespeichert wurden. Der Wert 0 steht für eine kontinuierliche Speicherung der Bildzeilen (das Bild wird Zeile für Zeile gespeichert). Der Wert 1 steht für das Adam7-Interlaced-Verfahren.

Der Palette-CHUNK (PLTE)

Dieser CHUNK besitzt die Signatur PLTE für den CHUNK-Typ und ist in allen PNG-Dateien enthalten, die Farbbilder mit Palettendaten verwenden. Der PNG-Datenstrom darf dabei nur einen PLTE-CHUNK enthalten. Der CHUNK kann dabei zwischen 3 und 768 Bytes Daten enthalten.

Offset	Bytes	Bemerkungen
00H	3	Wert für die Palette 1 Byte rote Komponente 1 Byte grüne Komponente 1 Byte blaue Komponente

Tabelle 35.4 Aufbau des Palette-CHUNK

Jeder Eintrag der Palette umfaßt drei Bytes mit den Farbanteilen Rot, Grün und Blau. Im Palette-CHUNK lassen sich dabei zwischen 1 und 256 Farben definieren. Ein PLTE-CHUNK muß vor dem IDAT-CHUNK mit den Daten stehen.

Der Image-Daten-CHUNK (IDAT)

Dieser CHUNK besitzt die Signatur IDAT und enthält die komprimierten Daten des Bildes. Bei umfangreicheren Bildern können diese in mehreren aufeinanderfolgenden IDAT-CHUNKs hintereinander abgelegt werden.

Der Trailer-CHUNK (IEND)

Die PNG-Datei muß mit einem IEND-CHUNK abgeschlossen werden. Dieser CHUNK enthält keinen Datenbereich.

Untergeordnete (ancillary) CHUNKs

Neben den bereits erwähnten kritischen CHUNKs enthält die PNG-Spezifikation in der Version 1.0 weitere 10 untergeordnete CHUNKs. Mit diesen CHUNKs lassen sich Zusatzinformationen speichern, die zur Interpretation der PNG-Daten hilfreich sind.

Der Background Color-CHUNK (bKGD)

Dieser CHUNK besitzt die Signatur bKGD und muß vor dem IDAT-CHUNK, aber nach dem PLTE-CHUNK auftreten. Es darf nur einen solchen CHUNK in der PNG-Datei geben. Der CHUNK legt die Hintergrundfarbe für das Bild fest (der PNG-Reader kann aber diese Vorgabe ignorieren und eine eigene Hintergrundfarbe verwenden). Der Datenbereich dieses CHUNK wird durch den Bildtyp beeinflußt:

- Enthält die PNG-Datei ein Farbbild mit einer Palette, wird im bKGD-CHUNK nur ein Byte mit dem Palettenindex der Hintergrundfarbe hinterlegt.
- Bei einem Graustufenbild enthält der Datenbereich ein Wort (16 Bit) mit dem Graustufenwert für den Hintergrund.
- Enthält die PNG-Datei dagegen ein True-Color-Bild, werden im Datenbereich des bKGD-CHUNK drei 16-Bit-Werte für die Farbanteile Rot, Grün, Blau hinterlegt.

Die betreffenden Werte werden dann im Datenbereich des bKGD-CHUNK abgelegt.

Der Primary Chromaticities und White Point-CHUNK (cHRM)

Dieser CHUNK besitzt die Signatur cHRM und darf nur einmal innerhalb der PNG-Datei auftreten. Der CHUNK muß vor den PLTE- und IDAT-CHUNKs stehen und gibt die Werte *Chromaticity* und *White Point* im 1931 CIE XYZ-Farbraum an.

Offset	Bytes	Bedeutung
00H	4	X-Wert White Point
04H	4	Y-Wert White Point
08H	4	X-Wert Rot
0CH	4	Y-Wert Rot

Offset	Bytes	Bedeutung
10H	4	X-Wert Grün
14H	4	Y-Wert Grün
18H	4	X-Wert Blau
1CH	4	Y-Wert Blau

Tabelle 35.5 Aufbau des cHRM-CHUNK

Die Einträge sind alle als DWORD definiert und legen lediglich die X/Y-Koordinaten für den weißen Punkt und für den Farbpunkt im 1931 CIE-Farbraum fest. Die im CHUNK enthaltenen Werte sind durch 100.000 zu dividieren, um aus der Ganzzahl eine Dezimalzahl zu machen. Die Werte sind zur Kalibrierung von RGB-Farbbildern erforderlich.

Der Image Gamma-CHUNK (gAMA)

Der Gamma-CHUNK besitzt die Signatur gAMA und muß vor den PLTE- und IDAT-CHUNKs stehen. Der CHUNK besitzt im Datenbereich einen 32-Bit-Eintrag (DWORD), der dem mit 100.000 multiplizierten Gamma-Wert (der Kamera) entspricht.

Der Image Histogramm-CHUNK (hIST)

Der Histogramm-CHUNK besitzt die Signatur hIST und muß zwischen PLTE- und IDAT-CHUNK stehen. Der CHUNK besitzt im Datenbereich ein Feld mit 16-Bit-Werten. Jeder Eintrag beschreibt die Häufigkeit der Paletteeinträge (Usage Frequency) im Bild. Können die in der Palette enthaltenen Farben nicht dargestellt werden, läßt sich anhand dieses CHUNK festlegen, welche Farben in darstellbare Farben umzuwandeln sind (die Farben, die am seltensten im Bild vorkommen).

Der Physical Pixel Dimensions-CHUNK (pHYs)

Der Histogramm-CHUNK besitzt die Signatur pHYs und muß vor dem IDAT-CHUNK stehen. Der CHUNK besitzt im Datenbereich die folgenden Einträge:

Offset	Byte	Bedeutung
00H	4	Pixel per Unit in X-Richtung
04H	4	Pixel per Unit in Y-Richtung
08H	1	Flag 0 = Einheit unbekannt 1 = Einheit Meter

Tabelle 35.6 Der Physical Pixel Dimension-CHUNK

Die Einträge des CHUNK geben die Abmessungen eines Bildpunkts (Pixel) in X- und Y-Richtung an. Ist die Einheit (Unit) nicht bekannt, gibt der pHYs-CHUNK ein Seitenverhältnis (Aspect Ratio) an. Fehlt der CHUNK, sind die Pixel als rechteckige Punkte anzunehmen.

Der Significant Bits-CHUNK (sBIT)

Dieser CHUNK besitzt die Signatur sBIT und legt fest, daß nur eine bestimmte Anzahl an Bits in den Bilddaten zu berücksichtigen sind. Dies erlaubt auch Bilder zu speichern, deren Bit pro Pixel-Werte (oder Bits per Sample) nicht der PNG-Spezifikation entsprechen. In diesem Fall müssen die Daten bis zur nächsten gültigen Stufe mit Nullbits aufgefüllt werden (z. B. aus 5 Bit pro Pixel wird 8 Bit pro Pixel). Der CHUNK besitzt folgende Einträge:

Bytes	Bedeutung
1	Graustufenbilder (Color Typ 0) Zahl der signifikanten Bits bei Graustufenbildern
1	Graustufenbilder mit Alphakanal (Color-Typ 4) Zahl der signifikanten Bits in den Graustufen
1	Signifikante Bits im Alphakanal
1 1 1	TrueColor (Color Typ 2 oder 3) Signifikante Bits rote Farbe Signifikante Bits grüne Farbe Signifikante Bits blaue Farbe
1 1 1	TrueColor mit Alphakanal (Color-Typ 6) Signifikante Bits rote Farbe Signifikante Bits grüne Farbe Signifikante Bits blaue Farbe

Tabelle 35.7 Der Significant Bits-CHUNK

Der CHUNK muß vor dem PLTE- und vor dem IDAT-CHUNK auftreten. Im IDAT-CHUNK sind dann nur die signifikanten Bits für jeden Bildpunkt auszuwerten. Die Werte im Datenteil des CHUNK müssen größer 0 und kleiner als die erlaubte Zahl der Bits pro Sample sein.

Der Textual Data-CHUNK (tEXt)

Dieser CHUNK besitzt die Signatur tEXt und erlaubt es, lesbare Texte (z. B. Autor, Copyright etc.) in einer PNG-Datei mit abzulegen. Der Datenteil des CHUNK besitzt folgenden Aufbau:

Offset	Byte	Bedeutung
00H	n	String mit dem Schlüsselwort
..H	1	00H als Separator
..H	n	Text

Tabelle 35.8 Der Textural Data-CHUNK

Das erste Feld kann zwischen 1 und 79 Bytes umfassen und muß mit einem Nullbyte 00H abgeschlossen werden. In diesem Feld wird der Typ der Information festgehalten. Zur Zeit sind folgende Begriffe definiert: Title, Autor, Description, Copyright, Creation Time,

Software, Disclaimer, Warning, Source, Comment. In *Source* können Sie zum Beispiel den Namen des erzeugenden Geräts oder Programms hinterlegen. Weiterhin lassen sich neue Schlüsselwörter mit einer maximalen Länge von 79 Bytes definieren. Das Nullbyte dient als Separator für den folgenden Text. Dessen Länge ist aus der Längenangabe des CHUNK und der Länge des vorhergehenden ASCIIZ-Strings zu ermitteln. Eine PNG-Datei kann mehrere tEXt-CHUNK enthalten.

Der Image Last-Modification Time-CHUNK (tIME)

Dieser CHUNK besitzt die Signatur tIME und gibt den Zeitpunkt der letzten Änderung am Bild an. Der CHUNK besitzt folgende Einträge im Datenbereich:

Offset	Bytes	Bedeutung
00H	2	Jahr (z.B. 1995)
02H	1	Monat (1-12)
03H	1	Tag (1-31)
04H	1	Stunde (1-23)
05H	1	Minute (0-59)
06H	1	Sekunde (0-60)

Tabelle 35.9 Der Image Last-Modification Time-CHUNK

Die Zeitangaben sollten in der Universalzeit (GMT) und nicht in der lokalen Zeit erfolgen.

Der Transparency-CHUNK (tRNS)

Dieser CHUNK besitzt die Signatur tRNS und muß hinter dem PLTE-CHUNK, aber vor dem IDAT-CHUNK auftreten. Dieser CHUNK kann benutzt werden, falls kein Alphakanal definiert wurde, um die Sichtbarkeit von Bildteilen festzulegen. Der CHUNK besitzt folgende Einträge im Datenbereich:

Bytes	Bedeutung
2	Graustufenbilder (Color-Typ 0) Wert der Graustufe, die transparent darzustellen ist
2 2 2 2	TrueColor (Color-Typ 2) Rotanteil der transparenten Farbe Grünanteil der transparenten Farbe Blauanteil der transparenten Farbe
N*1	Farbbilder mit Palette (Color Typ 3) Transparenzwert (für jeden Wert der Palette)

Tabelle 35.10 Datenbereich des Transparency-CHUNK

Entspricht eine Farbe bei Graustufen- und TrueColor-Bildern dem Transparenzwert, ist der betreffende Bildpunkt transparent darzustellen. Bei Farbbildern mit einer Palette ent-

hält der Datenbereich für jeden Paletteneintrag ein Byte mit dem Transparenzwert. Hier werden die Transparenzwerte wie Alphawerte interpretiert (0 = 100 % transparent, 255 = opak).

Enthält der CHUNK weniger Werte als Paletteneinträge, ist für die fehlenden Einträge der Wert 255 anzunehmen. Soll nur die erste Farbe transparent sein, wird z. B. im CHUNK nur der Wert 0 im ersten Datenbyte abgelegt.

Der Compressed Textual Data-CHUNK (zTXt)

Dieser CHUNK besitzt die Signatur zTXt und wird benutzt, um längere Texte zu speichern. Der CHUNK besitzt den gleichen Zweck und einen ähnlichen Aufbau wie der tEXt-CHUNK. Die Texte werden allerdings mit dem Deflate-Verfahren komprimiert. Der CHUNK beginnt mit dem komprimierten Schlüsselwort, an das sich das Nullbyte anschließt. Im folgenden Byte wird der Wert 0 für die Deflate/Inflate-Komprimierung hinterlegt. Daran schließt sich der komprimierte Text an. Sobald Sie den Inhalt des CHUNK dekomprimiert haben, liegt die gleiche Struktur wie beim tEXt-CHUNK vor.

Anmerkung: Damit sind die in der PNG-Version 1.0 definierten CHUNKs vorgestellt. Es kann aber weitere CHUNKs geben. Die Vorschläge für neue CHUNKs lassen sich im Internet unter der Adresse *png-info@uunet.uu.net* abrufen. Weiterhin lassen sich private CHUNKs definieren, die nur innerhalb einer Anwendung genutzt werden können.

Der Bilddatenbereich

Die Bilddaten werden in IDAT-CHUNKs hinterlegt. Eine PNG-Datei kann dabei einen oder mehrere aufeinanderfolgende IDAT-CHUNKs aufweisen. Ein IDAT-CHUNK enthält immer komprimierte Bilddaten (die Länge des Bilddatenbereichs darf dabei durchaus 0 oder 1 Byte betragen, auch wenn dies aus Speichergründen nicht allzu sinnvoll ist). Beim Schreiben der Bilddaten führt das Programm folgende Schritte aus:

- Es wird eine Zeile des Bildes gelesen. Dann ist die geforderte Filtermethode auf diese Bilddaten anzuwenden.
- Anschließend werden die gefilterten Daten komprimiert. Diese komprimierten Daten sind in einen IDAT-CHUNK zu schreiben.

Beim Lesen können die Bilddaten in aufeinanderfolgenden Zeilen oder im Interlaced-Modus gewählt werden (dies wird im Header festgehalten). Zum Dekomprimieren ist die obige Reihenfolge umzukehren.

Die dekomprimierten Daten des Bildes liegen dabei zeilenweise Punkt für Punkt vor. Der Nullpunkt des Bildes befindet sich in der oberen linken Ecke. Bei der Komprimierung werden die Daten einer Zeile gepackt. Enthält ein Pixel weniger als 8 Bit, werden die fehlenden oberen Bits mit Nullbits ergänzt. Damit wird sichergestellt, daß eine Zeile immer auf einer Bytegrenze endet. Eine Zeile beginnt ebenfalls an einer Bytegrenze.

Filterung

Die PNG-Spezifikation erlaubt, verschiedene Filtermethoden auf die Bilddaten anzuwenden. Ziel der Filterung ist es, die Komprimierbarkeit der Daten zu verbessern. Durch die Filterung gehen keine Daten verloren, und der Umfang der Bilddaten nimmt auch nicht ab. In der PNG-Spezifikation sind verschiedene Filtertypen erlaubt (z. B. keine Filterung oder einer der angegebenen Filter). Für jede Bildzeile läßt sich ein eigener Filtertyp angeben. Diese Information wird im ersten Byte der komprimierten Bildzeile gespeichert. In der PNG-Spezifikation sind folgende Filtermethoden vereinbart:

Code	Name
0	kein Filter (None)
1	Sub
2	Up
3	Average
4	Path

Tabelle 35.11 Filtermethoden

Beim Filtertyp 0 werden die Bildzeilen unverändert komprimiert. Beim Filtertyp 1 wird die Differenz zwischen jedem Byte und dem korrespondierenden Byte des vorhergehenden Bildpunkts übertragen. Beim Filtertyp 2 wird die Differenz zwischen jedem Byte und dem darüberliegenden Bildpunkt (Pixel) übertragen. Dies ähnelt dem Filtertyp 1, bei dem der links danebenliegende Bildpunkt zur Berechnung der Differenz verwendet wird. Der Filtertyp 3 verwendet den Mittelwert zweier benachbarter Bildpunkte (links und oberhalb), um den Wert eines Bildpunkts zu berechnen. Beim Filtertyp 4 wird eine lineare Funktion durch drei benachbarte Bildpunkte (links, oberhalb, links oben) gelegt, um den Wert des Bildpunkts zu ermitteln.

Anmerkungen: Detailliertere Informationen über die Filtermethoden sind in der PNG-Spezifikation enthalten.

Interlacing

Die Bildzeilen können innerhalb einer PNG-Datei im Interlaced-Modus gespeichert sein. Dies ermöglicht es dem Decoder, ein Bild in Rohform aufzubauen und dann schrittweise zu verfeinern. Dies ist bei einer seriellen Übertragung aus Online-Medien hilfreich. Damit erhält der Benutzer bereits beim Laden eine Grobübersicht des Bildes.

Beim Interlaced-Modus 0 (dies entspricht einem ausgeschalteten Interlaced-Modus) werden die Bildpunkte sequentiell von links nach rechts gespeichert. Alle Bildzeilen folgen in sequentieller Reihenfolge.

Die PNG-Spezifikation 1.0 kennt noch den Interlaced-Modus *Adam7* (dieser wurde nach dem Autor Adam M. Costello benannt). Im Gegensatz zum in der GIF-Spezifikation ver-

wendeten zeilenweisen Interlaced-Modus (hier werden einzelne Zeilen in einer bestimmten Reihenfolge übertragen) verwendet die PNG-Spezifikation ein etwas komplexeres Verfahren, um Bildteile zu übertragen. Dieses Verfahren verwendet 7 Durchläufe (Passes), wobei in den ersten sechs Durchläufen die Pixel der geraden Bildzeilen 0, 2, 4, 6 etc. erzeugt werden. Im siebten Durchlauf werden dann die ungeraden Bildzeilen 1, 3, 5, 7 etc. ergänzt. Bei den ersten sechs Durchläufen werden aber nicht die Bilddaten einer kompletten Zeile bearbeitet, sondern der PNG-Algorithmus wendet die nachfolgend gezeigte Maske (beginnend in der oberen linken Ecke) auf die Bilddaten an.

```
1 6 4 6 2 6 4 6
7 7 7 7 7 7 7 7
5 6 5 6 5 6 5 6
7 7 7 7 7 7 7 7
3 6 4 6 3 6 4 6
7 7 7 7 7 7 7 7
5 6 5 6 5 6 5 6
7 7 7 7 7 7 7 7
```

Die durch diese Maske ausgewählten Bildpunkte werden anschließend in den einzelnen Durchläufen gespeichert. Dies erlaubt dem Benutzer, bereits bei 20 bis 30 Prozent der übertragenen Daten den Bildinhalt zu erkennen (bei GIF sind ca. 50 % der Daten erforderlich). Allerdings hat das Interlacing den Nachteil, daß sich die Komprimierbarkeit der Bilddaten verschlechtert. (Bei jedem Durchlauf wird immer nur der achte Bildpunkt jeder achten Bildzeile übertragen. Bei Pass 2 werden z. B. die Pixel 4, 12, 20 etc. der Bildzeilen 0, 8, 16, etc. übertragen.) Durch das Interlacing werden quasi Teilbilder mit 8 x 8 Pixel erzeugt, die anschließend gefiltert und komprimiert werden.

Deflate/Inflate-Komprimierung

Bei PNG-Dateien wird zur Zeit nur der Komprimierungstyp 0 unterstützt. Es handelt sich um eine Deflate/Inflate-Komprimierung mit einem 32 kByte *Sliding Window*. Die Deflate-Komprimierung wurde vom LZ77-Komprimierungsverfahren abgeleitet. Dieses Verfahren ist lizenzfrei, und portierbare Bibliotheken mit C-Programmen sind frei verfügbar. Die PNG-Spezifikation benutzt das zlib-Format zur Speicherung der komprimierten Daten:

Byte	Bemerkung
1	Komprimierungsmethode/Flags-Code
1	Zusatz-Flags/Check Bits
n	Komprimierter Datenblock
4	Prüfwert

Tabelle 35.12 Aufbau eines komprimierten Datensatzes

Für den PNG-Komprimiertyp 0 muß der Wert 8 im ersten Byte (Komprimierungsmethode/Flags-Code) eingetragen werden. Der Prüfwert am Ende des Datenstroms wird mit der CRC-32-Methode über die unkomprimierten Daten berechnet (es wird das gleiche Polynom wie bei den Prüfsummen der CHUNKs verwendet).

Anmerkung: Dokumentation und C-Programmbeispiele hinsichtlich der Komprimierung/Dekomprimierung finden Sie im Info-Zip-Archiv (Internet-Adresse *ftp://ftp.uu.net/pub/archiving/zip/*). Die PNG-Spezifikation ist in vielen Online-Foren enthalten. Im Internet finden Sie die Spezifikation unter *http://sunsite.unc.edu/boutell/png.html*. Das offizielle PNG-Archiv befindet sich unter *ftp://ftp.uu.net/graphics/png/*, und im Unterverzeichnis *ftp://ftp.uu.net/graphics/png/src/* ist eine Referenzimplementierung eines PNG-Readers und -Writers in C zu finden. Weitere Informationen finden sich z. B. unter *http://quest.jpl.nasa.gov/PNG/* und unter *http://www.group-42.com/*.