

Institut für Softwaretechnik und Interaktive Systeme  
Interaktive und Multimediale Systeme  
Technische Universität Wien

Projektpraktikum (mit Bakkalaureatsarbeit)  
SS 2004

# Vergleich von Komprimierungsverfahren

Thema Nr. 25

Wien, 15.06.2004

Marie-Theres Tschurlovits 0125975

[mth.web@gmx.net](mailto:mth.web@gmx.net)

**Inhalt**

<b>1.</b>	<b>Zusammenfassung</b> .....	<b>3</b>
1.1	Links auf andere Lerneinheiten .....	3
<b>2.</b>	<b>Level of Detail 2</b> .....	<b>4</b>
2.1	Allgemeines zur Kompression .....	4
2.2	GIF – Graphics Interchange Format.....	4
2.2.1	Aufbau .....	6
2.2.2	Kompression.....	6
2.2.3	Pro und Contra .....	8
2.3	BMP – Bitmap Picture .....	8
2.3.1	Aufbau .....	8
2.3.2	Kompression.....	9
2.3.3	Pro und contra .....	9
2.4	PNG – Portable Networks Graphic.....	9
2.4.1	Aufbau .....	9
2.4.2	Kompression.....	11
2.4.3	Interlacing .....	11
2.4.4	Pro und contra .....	11
2.5	TIFF – Tagged Image File Format .....	11
2.5.1	Aufbau .....	12
2.5.2	Kompression.....	12
2.5.3	Pro und contra .....	12
2.6	JPEG – Joint Photographic Experts Group .....	13
2.7	JPEG2000 .....	13
2.8	Übersicht .....	14
<b>3.</b>	<b>Level of Detail 3</b> .....	<b>15</b>
3.1	Vergleich der Auflösung von Farb- bzw. Helligkeitswerten .....	15
3.1.1	Vorschlag interaktive Anwendung .....	15
3.1.2	Weiterführende Literatur und Links .....	15
<b>4.</b>	<b>Literatur</b> .....	<b>16</b>
4.1	Eigene Literatur .....	16

## 1. Zusammenfassung

In dieser LU wird versucht einen kurzen Überblick über einen kleinen Teil der großen Fülle von unterschiedlichen Bildformaten zu geben. Jedes dieser Formate hat seine Stärken und seine Schwächen, kaum eines ist universell einsetzbar, da große Vorteile in einem Bereich oft Nachteile in einem anderen bedingen. Oft ist es schwierig für die jeweilige Anwendung das richtige Format auszuwählen. Es wird versucht einen kurzen Einblick in die Eigenschaften, den Aufbau, die verwendeten Kompressionsverfahren und die unterschiedlichen Einsatzgebiete zu geben.

### 1.1 Links auf andere Lerneinheiten

LU18 – JPEG

LU22 – JPEG2000

LU26 – JPEG vs. JPEG2000

## 2. Level of Detail 2

*Definition:* Unter *Kompression* oder *Komprimierung* versteht man einen Vorgang, Daten (Audio, Video, Text, ...) auf eine Weise zu kodieren, sodass der Datenumfang (der Speicherplatz) reduziert wird. Dabei gibt es die Möglichkeit einer *verlustlosen* Kompression, die verkleinerten Speicherbedarf bei gleicher Informationsmenge liefert. *Verlustbehaftete* Kompression führt zwar auch zu einer Verringerung des Speicherbedarfs, ist aber mit einem Informationsverlust verbunden.

Die Gründe für Kompression sind einerseits, durch Verkleinerung von Files mehr Information auf einem Speichermedium ablegen zu können, andererseits die Datenmenge, die über einen Übertragungskanal geschickt werden soll so klein wie möglich zu halten. Allgemein ausgedrückt ist Kompression für eine effiziente Nutzung von Systemressourcen (Rechenleistung, Speicherplatz, Breitbandausnutzung) unerlässlich.

Anforderungen, die an Verfahren zur Datenkomprimierung gestellt werden sind u.a., eine hohe Qualität der komprimierten und anschließend dekomprimierten Daten zu gewährleisten, trotzdem hohe Kompressionsraten zu erzielen und die Komplexität der verwendeten Verfahren gering zu halten um eine effektive Realisierung zu ermöglichen.

Die Vielzahl der heutzutage eingesetzten Bildformate lässt sich dadurch erklären, dass diese Formate meist für jeweils verschiedene Einsatzgebiete entwickelt und in dieser Hinsicht optimiert wurden. [E14, S. 14 ff]

### 2.1 Allgemeines zur Kompression

Die Wahl eines geeigneten Kompressionsverfahrens ist nicht immer einfach, da immer Aspekte wie Qualität und Speicherbedarf gegeneinander abgewogen werden müssen. Oft gibt es nicht das „perfekte“ Komprimierungsverfahren, da meist ein Kompromiss zwischen passabler Qualität und annehmbarer zu übertragender Datenmenge geschlossen werden muss.

#### Verlustfreie Kompression

Die *verlustfreie Kompression* (*lossless compression*) wird in Bereichen eingesetzt, wo man sich keinen Informationsverlust erlauben darf (z.B. medizinische Daten). Bildformate wie BMP, PNG und TIFF unterstützen verlustfreie Komprimierung.

Zu den Verfahren verlustfreier Komprimierung zählen u.a. LZW (Lempel-Ziv-Welch), RLE (Run-Length-Encoding, Lauflängenkodierung) und CCITT (Fax-Standard).

#### Verlustbehaftete Kompression

Kompression, bei der ein gewisser Informationsverlust entsteht (*lossy compression*) wird vor allem dort eingesetzt wo Bildinformation übertragen werden muss (z.B. digitales Fernsehen, Internetanwendungen, ...).

Verlustbehaftete Kompression hat den Nachteil, dass aufgrund des Informationsverlusts das ursprüngliche Bild nicht mehr hundertprozentig wiederhergestellt werden kann. Außerdem ist darauf zu achten, dass bei zu starker Komprimierung sichtbare Fehler entstehen, so genannte *Artefakte*. Verlustbehaftete Codierung wird u.a. von JPEG und PNG unterstützt. [E02, S. 15]

### 2.2 GIF – Graphics Interchange Format

*GIF* (*Graphics Interchange Format*) wurde von CompuServe als plattform- und geräteunabhängiges Dateiformat entwickelt. Die Kompression ist verlustfrei und relativ effizient. Außerdem gibt es die Möglichkeit mehrere Bilder in eine Datei zu verpacken. Die 1989 fertiggestellte Erweiterung des Formats unterstützt Transparenz und so genannte *Extensions*. Aufgrund all dieser Umstände ist GIF neben JPEG (und event. PNG) eines der wichtigsten Formate für Anwendungen im Internetbereich.

Das Format arbeitet mit bis zu 256 indizierten Farben, die aus der Menge der 24-Bit Farben ausgewählt werden können. In einer *CLUT* (*Color Lookup Table*) werden die Farben einer indizierten Bilddatei abgelegt. Die RGB-Farbwerte werden nur an einer Stelle, der CLUT, global gespeichert. Da nur noch auf diese Farben referenziert werden muss und nicht für jeden Pixel die Farbwerte separat gespeichert werden, kann so eine Verkleinerung der

## Grundlagen der perzeptuellen Bildkodierung

Filegröße erreicht werden. Für Anwendungen, bei denen nicht allzu viele verschiedene Farben benötigt werden (Logos, Illustrationen, Cartoons, ...) ist dies ein sehr effizientes Verfahren.

GIF unterstützt weiters *Interlacing* zum stückweisen Anzeigen des Bildes (z.B. auf einer Webseite) und Transparenz. Dabei wird eine Farbe aus der Farbpalette als transparent definiert (z.B. alle Pixel mit dem weißem Farbwert #FFFFFF werden transparent gesetzt).

Es gibt drei verschiedene Sonderformen von GIF: animiertes, transparentes und interlaced GIF.

### Animiertes GIF

Das GIF-Format erlaubt die Speicherung mehrerer Bilder in einer Datei. Das bedeutet, dass damit einfache Animationen, ähnlich einem „Daumenkino“, erzeugt werden können.

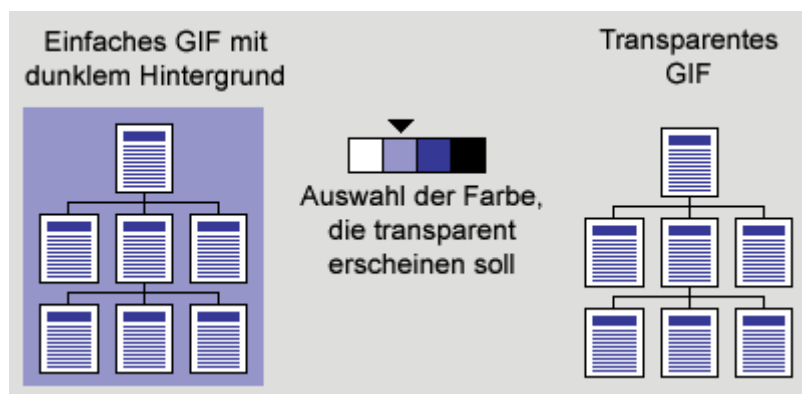
*Beispiel:* Nachdem das GIF vom Browser geladen wurde, werden die einzelnen Bilder der Reihe nach geöffnet, was den Anschein einer Bewegung erzeugt.

### GIF mit Interlacing

Im Interlaced Modus wird das Bild in vier Durchgängen aufgebaut, wobei mit jedem Durchgang mehr Information sichtbar wird. Ungefähr nach Übertragung der halben Datenmenge lässt sich bereits der Bildinhalt erkennen.

### Transparentes GIF

Dateiformate, die Transparenz unterstützen sind in Abbildungen interessant, bei denen kein rechteckige Form dargestellt werden soll. Man wählt eine Farbe aus der Farbpalette, die transparent erscheinen soll. Bei Formaten ohne Transparenz wird stattdessen der restliche Bereich rund um einen runden Gegenstand mit der Farbe Weiß aufgefüllt. [E02, S. 40 ff, E04]



**Abbildung 1: Entstehung eines transparenten GIFs**  
[<http://www.webstyleguide.com/graphics/graphics/7.18.gif>]

*Erklärung:* Abbildung 2 zeigt, wie ein GIF mit transparentem Hintergrund entsteht. Beim Speichern des Bildes wird vom Benutzer festgelegt, welche Farbe aus der globalen Farbtabelle des GIFs transparent erscheinen soll (in diesem Bildbeispiel die blau-violette Farbe #9999CC). Die so ausgewählten Pixel erscheinen nun transparent und die Hintergrundfarbe hellgrau #CCCCCC wird auch hinter der Abbildung sichtbar.

Es gibt zwei Modi des Bildschirmaufbaus: sequentiell und interlaced.

### Sequenzieller Modus

Bei dieser Einstellung wird das Bild zeilenweise von links oben nach rechts unten abgearbeitet.

### Interlaced Modus

Im Interlaced Modus erfolgt der Bildaufbau in vier Durchgängen. Im ersten Durchgang wird beginnend bei Zeile 0 jede achte Zeile angezeigt (0, 8, 16, ...). Der zweite Durchlauf liefert ausgehend von Zeile 4 wieder jede achte Zeile (4, 12, 20, ...). Im dritten Durchlauf werden die verbleibenden geraden Zeilen beginnend bei Zeile 2 ausgegeben (2, 6, 10, ...). Der vierte und letzte Durchgang liefert mit der Anzeige aller ungeraden Zeilen (1, 3, 5, ...) das vollständige Bild. Schon ab einer Anzeige von ca. 50% der Bildinformation kann der Inhalt erahnt werden. [E02, S. 41 ff, E05].

## 2.2.1 Aufbau



**Abbildung 2: Aufbau des GIF89a [E02, S. 42]**

*Erklärung:* Abbildung 3 zeigt die Erweiterung des GIF-Formats aus dem Jahr 1989. Neben dem Header, der Bildschirmdefinition und der globalen Farbtabelle können zu jedem Bild noch Erweiterungen (Extensions) gespeichert werden.

Jede GIF-Datei beginnt mit einem 6 Byte großen *Header*, der die Signatur des Formats enthält. Dabei enthalten die ersten drei Byte den Text „GIF“ und die weiteren Byte die GIF-Version des Bildes (87a oder 89a).

Die *Bildschirmdefinition* beinhaltet Informationen über die Abmessungen, die Auflösung (resolution flag), die Hintergrundfarbe der Grafik und das Pixelseitenverhältnis (aspect ratio).

Auf den Header folgen die *Bildblöcke*, wobei jedes Bild durch einen eigenen Bildblock repräsentiert wird. Jedes Bild bekommt eine Position auf dem, in der Bildschirmdefinition festgelegten Schirm, zugewiesen. Eventuell hat ein Bild auch eine eigene Farbtabelle, die in diesem Fall die globale Farbtabelle für dieses Bild überschreibt.

Der *Trailer* am Ende des GIF-Files markiert das Dateiende.

Das 1989 eingeführte GIF98a-Format unterscheidet sich von dem ursprünglichen GIF durch weitere Informationsblöcke, den so genannten *Control Extensions* (Erweiterungen). Diese bieten zusätzliche Informationen über Berechnung und Darstellung der Bilder. [E02, S. 42 ff]

## 2.2.2 Kompression

GIF wird mit der LZW-Kompression verlustfrei komprimiert, was zu Kompressionsraten von 30-70% führt.

### LZW – Lempel Ziv Welch

Der verlustlose LZW Datenkomprimierungsalgorithmus ist eine Weiterentwicklung des 1977 von Abraham *Lempel* und Jacob *Ziv* veröffentlichten LZ77-Algorithmus. Die zweite Auflage des Algorithmus, LZ78 (1978), wurde von Terry *Welch* weiter optimiert was 1984 in dem als *Lempel-Ziv-Welch (LZW)* bekannten Algorithmus resultierte.

Der im Folgenden erklärte Algorithmus stützt sich auf den Umstand, dass in Texten viel redundante Information enthalten ist. Diese Sichtweise lässt sich auch auf Bildinformation umlegen, indem man bedenkt, dass mehrere gleichfarbige Pixel einer Bildzeile effizienter gespeichert werden könnten, als sich für jeden einzelnen Pixel den Farbwert zu merken.

Die Grundidee der LZW-Kompression ist also, das Auftreten einer speziellen Textphrase oder Menge von Bytes in einem Datenstück durch eine Referenz auf ein früheres Vorkommen dieses Stücks zu ersetzen. Dazu werden neue Phrasen in ein Wörterbuch eingetragen. Bei erneutem Auftreten der gleichen Phrase wird statt dieser nur noch der Indexwert aus dem Wörterbuch gespeichert.

Der zugrunde liegende Algorithmus in Pseudocode sieht folgendermaßen aus:

```

set w = NULL
loop
  read a character K           // Einlesen des nächsten Zeichens
  if wK exists in the dictionary // Ist die Zeichenkette wK bereits im Wörterbuch enthalten
    w = wK
  else                         // Ist die Zeichenkette wK noch nicht im Wörterbuch enthalten
    output the code for w      // gib w aus
    add wK to the string table // Erweitere das Wörterbuch um wK
    w = K
endloop
    
```

*Beispiel:* Am besten lässt sich die Effizienz dieses Komprimierungsverfahrens an einem Textstück mit viel redundanter Information zeigen.

Eingabe: /WED/WE/WEE/WEB

Eingabe: Zeichen	Ausgabe: Code	Neue Wörterbuchindizes und die damit verbundene Zeichenkette
/W	/	256 = /W
E	W	257 = WE
D	E	258 = ED
/	D	259 = D/
WE	256	260 = /WE
/	E	261 = E/
WEE	260	262 = /WEE
/W	261	263 = E/W
EB	257	264 = WEB
<Ende>	B	

Der Algorithmus basiert darauf, aus bereits eingelesenen Zeichenketten automatisch ein Wörterbuch zu generieren. LZW startet mit einem Ausgangswörterbuch, dessen Einträge der Stellen 0 bis 255 bereits belegt sind (z.B. ASCII-Zeichen für Buchstaben a-z, Zahlen, Sonderzeichen, ...). Neu erkannte Zeichenketten, die sich noch nicht in dem Wörterbuch befinden, werden beginnend mit Index 256 im Wörterbuch abgelegt. Endresultat der Bearbeitung ist ein Code, der die Eingabe exakt repräsentiert, in diesem Beispiel:

/WED<256>E<260><261><257>B (Die eckigen Klammern „<“ und „>“ agieren hier als Trennzeichen der Wörterbuchindizes).

Das Wörterbuch selbst muss jedoch nicht mit dem komprimierten Text oder Bild übertragen werden, da es bei der Dekomprimierung auf dieselbe Art und Weise aufgebaut werden kann, wie bei der Komprimierung. Ist der

## Grundlagen der perzeptuellen Bildkodierung

Algorithmus korrekt implementiert, entstehen bei Komprimierung und Dekomprimierung exakt dieselben Wörterbücher.

*Beispiel (Fortsetzung):* Bei der Rückübersetzung der komprimierten Zeichenkette, werden die Wörterbuchindizes der Reihe nach wieder in Zeichenketten umgewandelt, das bedeutet:  $/WED<256>E<260><261><257>B$  wird wieder zu  $/WED/WE/WEE/WEB$ . [E11, E12, E13]

Eingabe: Code	Ausgabe: Zeichen	Neue Wörterbuchindizes und die damit verbundene Zeichenkette
/	/	256 = /W
W	W	257 = WE
E	E	258 = ED
D	D	259 = D/
256	/W	260 = /WE
E	E	261 = E/
260	/WE	262 = /WEE
261	E/	263 = E/W
257	WE	264 = WEB
B	B	

### 2.2.3 Pro und Contra

#### Vorteile

- Plattform- und Hardwareunabhängigkeit
- Interlacing
- Transparenz
- Möglichkeit für einfache Animationen
- Mitcodierung von ASCII-Text, z.B. für Copyright-Informationen

#### Nachteile

- Maximal 256 Farben darstellbar, deshalb nicht für fotorealistische Bilder geeignet [E02, S. 9]

## 2.3 BMP – Bitmap Picture

Das *BMP-Format (Bitmap Picture)* wurde von Microsoft entwickelt und wird so auch hauptsächlich von Windows benutzt. Es ist unabhängig vom jeweiligen Ausgabegeräte definiert (device independent). Dem BMP-Format liegt das RGB-Modell zugrunde, mit geeigneten Farbtabellen ist aber auch die Kodierung von monochromen Bildern und Graustufenbildern möglich. Das bedeutet BMP kann Schwarz-Weiß oder Farbbilder mit einer Farbtiefe von 1, 4, 8 oder 24 Bit speichern. Die Bilder werden meist ohne Datenkompression Pixel für Pixel gespeichert.

### 2.3.1 Aufbau

Eine Bitmap-Datei ist in vier Abschnitte unterteilt: Bitmap Header, Information Header, Farbpalette, Daten.

Der *Bitmap-Header* beinhaltet die Dateisignatur „BM“, die Datenlänge und Abstand zwischen Datei- und Datenanfang (Offset).

Im *Information-Header* werden Bildinformationen gespeichert, wie Abmessung, Auflösung, Komprimierungsart und Anzahl der Farben.

In der *Farbpalette* werden die Farben durch ihren Rot-, Grün- und Blauanteil definiert.

Der *Datenteil* enthält die zeilenweise Rasterinformation des Bildes, beginnend an der linken unteren Ecke. Bei Bildern mit 1-, 4- oder 8-Bitinformation werden pro Pixel Verweise auf die Farbpalette gespeichert. Bilder mit



einer Farbtiefe von 24-Bit benötigen keinen Verweis auf eine Farbtabelle, da hier der Pixelwert dem Farbwert entspricht. [E02, S. 26 ff]



**Abbildung 3: Aufbau des BMP-Formats [E02, S. 26]**

*Erklärung:* Abbildung 4 zeigt die Gliederung des BMP-Formats in den Bitmap-Header, den Information-Header, die Farbpalette und die eigentlichen Bilddaten.

### 2.3.2 Kompression

Bitmaps benutzen meist gar keine Kompression. In manchen Fällen wird Run-Length-Encoding (RLE, Lauflängenkodierung) eingesetzt.

### 2.3.3 Pro und contra

#### **Vorteile**

- Bitmaps können von Programmen ohne Verwendung eines Importfilter gelesen werden

#### **Nachteile**

- Aufgrund meist fehlender Datenkompression benötigen Bitmaps viel Speicherplatz

## 2.4 PNG – Portable Networks Graphic

Das Bildformat *PNG (Portable Networks Graphic)* wurde am in den Neunziger Jahren des 20. Jahrhunderts vom *MIT (Massachusetts Institute of Technology)* für den Einsatz in Internetanwendungen und für eine effiziente Speicherung von True-Color-Bildern (16,7 Millionen Farben) entwickelt. Die Absicht war es, das ältere und einfachere GIF-Format zu ersetzen und eine Alternative für das viel komplexere TIFF-Format zu schaffen. Im Vergleich mit GIF hat PNG die Vorteile, dass es Alpha-Kanäle (und damit auch Transparenz), Gamma-Korrektur (plattformübergreifende Kontrolle der Helligkeit eines Bildes) und zweidimensionales Interlacing unterstützt. In den meisten Fällen weist das PNG-Format auch einen höheren Kompressionsfaktor auf als GIF. Die einzige nicht von GIF übernommene Eigenschaft ist die Speicherung mehrerer Bilder und damit die Möglichkeit für kleinere Animationen.

Mit all diesen Eigenschaften bietet das PNG-Format im Internet Einsatz einen guten Kompromiss zwischen den Vorteilen des GIF- und den JPEG-Formats [E03, S. 950 ff, E15]

### 2.4.1 Aufbau

Das *PNG*-Format ist aus Blöcken, so genannten *Chunks*, aufgebaut. Die Anzahl der *Chunks* hängt von dem zu speichernden Bild ab.

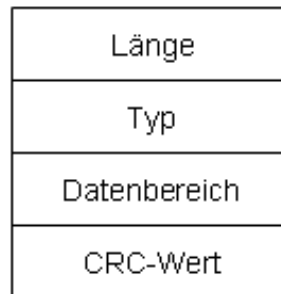
Die *Chunks* enthalten die gesamte Information einer *PNG*-Datei. Verschiedene Informationsarten werden dabei in unterschiedlichen Arten von *Chunks* gespeichert. Jeder *Chunk* ist aus vier Komponenten aufgebaut: Länge des *Chunks*, *Chunktyp*, Datenbereich des *Chunks*, *CRC*-Wert des Datenbereiches.

Die ersten vier Byte beinhalten eine Zahl vom Typ *Integer*, die die *Länge* des Datenteils angibt.

Die nächsten vier Byte geben den *Typ* des *Chunks* an, der festgelegt, wie die in diesem *Chunk* abgelegten Daten zu interpretieren sind. Für die Belegung dieser vier Byte sind nur *ASCII*-Zeichen für a-z bzw. A-Z gestattet, die zusammen das Kürzel des *Chunks* bilden.

Der *Datenteil* beinhaltet das eigentliche Bildmaterial, das entsprechend der *Typ*angabe interpretiert wird.

Den Abschluss eines Chunks bildet der *CRC-Wert* (*Cyclic Redundancy Check*). Dieser Prüfwert wird für die Erkennung von Fehlern (z.B. durch falsche Übertragung) benutzt. Da bei jeder neuen Speicherung auch ein neuer CRC-Wert berechnet wird, lässt sich durch das Vergleichen von CRC-Werten auch feststellen, ob eine PNG-Datei verändert wurde.



**Abbildung 4: Aufbau eines Chunks [E02, S. 47]**

*Erklärung:* In Abbildung 5 wird der Aufbau eines Chunks dargestellt. Ein Chunk besteht aus der Längeninformation, dem Chunk-Typ, dem Datenbereich des Chunks und dem CRC-Prüfwert.

### Critical Chunks

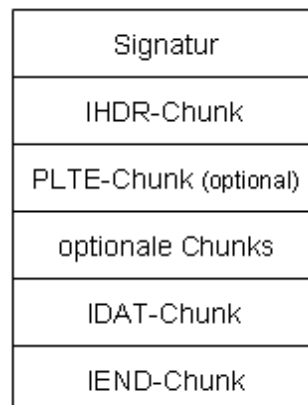
Es gibt folgende vier Standard-Blöcke, die so genannten Critical Chunks, die jedes PNG-verarbeitende Programm verstehen muss: Header Chunk, Palette Chunk, Image Data Chunk, Image Trailer Chunk.

Der *Header Chunk* (*IHDR*) enthält die Basisinformationen (Abmessung, Datentiefe, Farbmodell, Kompression, Interlacing) über die Datei.

Der optionale *Palette Chunk* (*PLTE*) enthält die Farbpalette und muss daher vor dem Chunk für die Bilddaten angeordnet sein.

Der *Image Data Chunk* (*IDAT*), der aus mehreren Blöcken bestehen kann, enthält die eigentlichen Bilddaten.

Das Ende des Datenstroms wird vom *Image Trailer Chunk* (*IEND*).



**Abbildung 5: Aufbau des PNG-Formats [E02, S. 47]**

*Erklärung:* Abbildung 6 zeigt den Aufbau einer PNG-Datei. Ein PNG-File gliedert sich in einzelne Blöcke, die so genannten Chunks. Zu Beginn des Files steht die Signatur. Danach folgen die Chunks für den Header (IHDR), die optionale Farbpalette (PLTE), weitere optionale Chunks, der Daten-Chunk (IDAT) und der Abschluss-Chunk (IEND), der das Ende des Datenstroms markiert.

### Aucillary Chunks

Die Aucillary Chunks sind ergänzende Chunks, die Zusatzinformationen beinhalten, die zur Darstellung des Bildes nicht unbedingt erforderlich sind.

### 2.4.2 Kompression

Das PNG-Format speichert Bilder immer in komprimierter Form. Als Kompressionsverfahren wird die Deflate Compression verwendet, eine Variante des LZ77-Verfahrens. Dabei werden die Differenzwerte der Pixel verlustfrei gespeichert.

Um einen höheren Kompressionsfaktor zu erzielen, kann ein Bild zur Vorbereitung vor dem Kompressionsschritt gefiltert werden. Die Filter werden zeilenweise eingesetzt, was bedeutet, dass theoretisch für jede Zeile ein anderer Filter möglich wäre. Fünf verschiedene Filtertypen werden unterschieden: none, sub, up, average und paeth.

### 2.4.3 Interlacing

Die Technik zum teilweisen Laden eines PNG-Bildes wird „Adam 7“ genannt. In den ersten sechs der insgesamt sieben Durchläufe werden die geraden Bildzeilen (0, 2, 4, ...) erzeugt. In einem Durchlauf werden jedoch noch nicht alle Bilddaten einer Zeile angezeigt, sondern nur Teile davon, entsprechend der Maske in Abbildung 7. Erst im siebten Durchlauf werden die ungeraden Bildzeilen (1, 3, 5, ...) ergänzt.

```
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
```

**Abbildung 6: Maske des PNG-Formats für die Erzeugung der geraden Bildzeilen [E02, S. 49]**

*Erklärung:* Die Maske in Abbildung 7 wird über das Bild gelegt. Die Zahl an der Stelle eines Pixel gibt den Durchgang an, in dem das Pixel angezeigt wird. Man sieht z.B. deutlich, dass die ungeraden Bildzeilen erst im siebten Anzeige-Durchlauf erzeugt werden.

Schon bei ca. 20-30% der übertragenen Daten kann man den Inhalt eines Bildes erkennen.

### 2.4.4 Pro und contra

#### Vorteile

- Sehr gute verlustfreie Kompression
- Lizenzfrei (Gebührenfrei)
- Unterstützung von 48-Bit Vollfarben und 16-Bit Graustufen
- Plattformunabhängigkeit
- 16-Bit Alphakanal
- Gamma-Korrektur
- Interlacing

#### Nachteile

- Im Gegensatz zu GIF keine Animationsmöglichkeit
- PNG-Format wird nicht von allen Programmen zur Gänze unterstützt [E02, S. 9]

## 2.5 TIFF – Tagged Image File Format

Das pixelorientierte Format *TIFF (Tagged Image File Format)* wurde 1986 von Aldus, Hewlett Packard und Microsoft entwickelt, und hat sich zu einem Standard der digitalen Bildverarbeitung entwickelt. Der Grundgedanke war, eine programm- und plattformunabhängige Speicherung von Pixeldaten zu ermöglichen. So entstand ein universell einsetzbares Format, das auf Unix-, Macintosh- und PC-Betriebssystemen einsetzbar ist. Auch

## Grundlagen der perzeptuellen Bildkodierung

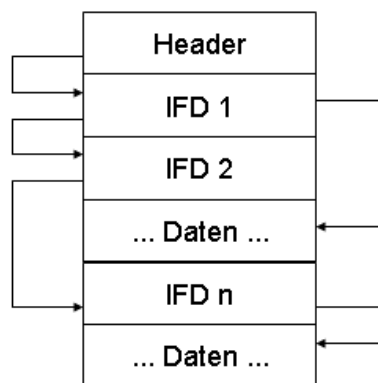
wegen der vielfältigen Möglichkeiten der Speicherung ist das TIFF-Format eines der wichtigsten Formate für Rasterdateien.

### 2.5.1 Aufbau

Der Header des TIFF-Formats hat eine fixe Größe und belegt immer die ersten acht Byte des Files. Er beinhaltet die Versionsnummer des Formats und einen Zeiger auf das erste *IFD (Image File Directory)*, welches das Inhaltsverzeichnis der TIFF-Datei darstellt.

Die Länge eines IFDs ist variabel und wird durch die Anzahl der Tag-Einträge bestimmt. Ein *Tag* („Anhänger“) ist eine 12 Byte große Datenstruktur, die Informationen über die Bilddaten (Abmessung, Auflösung, ...) speichert. Nach dem letzten Tag eines IFDs steht immer ein 4 Byte großer Zeiger auf das folgende IFD bzw. auf das Dateiende.

TIFF unterstützt außer dem RGB-Modell und Graustufen, auch 16-Bit Farbtiefe pro Grundfarbe und das CMYK-Farbmodell. Auch eine Speicherung von Alphakanälen ist möglich. Bei RGB-Bildern weist TIFF eine Datentiefe von 24 Bit auf, bei CMYK-Bildern von 32 Bit. [E02, S. 23, E03, S. 943 ff]



**Abbildung 7: Aufbau des TIFF-Formats [E02, S. 24]**

*Erklärung:* Abbildung 8 zeigt den schematischen Aufbau des TIFF-Dateiformats. Die ersten acht Byte einer TIFF-Datei sind für den Header reserviert. Danach werden die IFDs (Image File Directory) als verkettete Liste gespeichert.

### 2.5.2 Kompression

TIFF bietet sowohl eine verlustlose als auch eine verlustbehaftete Speicherung, mit einer großen Auswahl an Kompressionsverfahren. Die möglichen verlustfreien Kompressionsverfahren sind die PackBit-Codierung, die Fax- und die LZW-Komprimierung. Die Möglichkeit zur verlustbehafteten Speicherung bildet die JPEG-Komprimierung. [E02, S. 23 ff]

### 2.5.3 Pro und contra

#### Vorteile

- Informationen über die benutzte Grafiksoftware, den Scannertyp, ...
- Viele Farbmodi
- Plattformübergreifend
- Speicherung mehrerer Bilder in einer Datei
- Viele verschiedene Kompressionsmethoden

#### Nachteile

- Hoher Speicherplatzbedarf, dadurch nicht für die Verwendung im Internetbereich geeignet [E02, S. 23 ff]

## 2.6 JPEG – Joint Photographic Experts Group

*JPEG (Joint Photographic Experts Group)* ist eigentlich ein Standard für verlustbehaftete Komprimierung, der 1993 als ISO 10918 festgeschrieben wurde. Spricht man von „JPEG“, wird jedoch oft das Bildformat *JPEG JFIF (JPEG File Interchange Format)* gemeint, das von der *IJG (Independent JPEG Group)* entwickelt wurde. [E06]

JPEG ist ein verlustbehaftetes Verfahren, das auf die Komprimierung von Bildern mit fließenden Farbübergängen optimiert wurde. Für Bilder mit harten Kontrasten, Liniengrafiken oder Texten, die große Farbflächen mit starken Farbwechseln aufweisen ist dieses Bildformat weniger geeignet. [E07]

Für nähere Informationen siehe LU18 (JPEG).

## 2.7 JPEG2000

Das Format *JPEG2000*, eine Weiterentwicklung des JPEG-Formats wurde im Januar 2001 von der *International Organization of Standardization (ISO)* zum Standard erklärt. An der Entwicklung des Formats waren die ISO und die University of British Columbia (UBC) beteiligt.

JPEG2000 stellt eine Überarbeitung des bewährten JPEG-Formats dar und soll als Ergänzung dienen. Das neue Format soll bei höheren Kompressionsraten qualitativ hochwertigere Bilder liefern, da bei der Komprimierung im Gegensatz zu JPEG keine „pixeligen“ Artefakte entstehen. Anwendung findet JPEG2000 in Bilddatenbanken, der Medizin, usw.

Für nähere Informationen siehe LU22 (JPEG2000).

## 2.8 Übersicht

*Erklärung:* Die folgende Tabelle soll ohne Anspruch auf Vollständigkeit einen kurzen Überblick über die vorgestellten Bildformate hinsichtlich der verwendeten Farbmodelle, der Anzahl der Kanäle, der Bittiefe, den Kompressionsmethoden, dem Entwickler und den unterstützten Plattformen geben. [E02, S. 7 ff, E03, S. 947 ff]

	JPEG	JPEG2000	GIF
<b>Anwendung</b>	+ : Bilder mit vielen verschiedenen Farben (True Color): Fotos (Internet) – : Bilder mit großen gleichartigen Flächen: Logos, ... hoher Kompressionsfaktor führt zu Artefakten	Siehe JPEG, jedoch kein Problem mehr mit pixeligen Artefakten	+ : Bilder mit wenigen Farben: Internet (Logos, kleine Animationen, ...) – : Bilder mit vielen verschiedenen Farben, Fotos
<b>Farbmodelle</b>	RGB, CMYK	RGB, Graustufen, CMYK	Indizierte Farben
<b>Kanäle</b>	3 oder 4 Farbkanäle	256 Kanäle	1 Kanal
<b>Bittiefe</b>	8 Bit/Kanal	1-16 Bit/Kanal	1-8 Bit/Pixel (S/W-256 Farben)
<b>Kompression</b>	DCT	Wavelet	LZW
	verlustbehaftet	verlustbehaftet	verlustfrei
	Relativ guter Kompromiss zwischen Kompression und Bildqualität	Im Vergleich zu JPEG, bessere Kompression bei gleichbleibender Bildqualität	Gute Komprimierung bei vielen einfarbigen Flächen, nicht geeignet für True Color
<b>Entwickler</b>	JPEG, ISO	JPEG, ISO	CompuServe
<b>Plattformen</b>	Mac, Windows, Unix	Mac, Windows, Unix	Mac, Windows, Unix

	BMP	PNG	TIFF
<b>Anwendung</b>	+ : einfaches Bildformat, v.a. unter Windows verbreitet, geräteunabhängig – : keine oder nur geringe Kompression	+ : Internet, Verbindung der Vorteile von JPEG (True Color) und GIF (Transparenz, verlustfrei) – : noch geringere Verbreitung, da in älteren Versionen von Netscape und Internet Explorer nicht unterstützt	+ : bei entsprechender Auflösung Format für High-End-Druckausgabe, ohne Komprimierung ein Standardaustauschformat – : schlechtere Unterstützung von TIFF mit LZW-Komprimierung,
<b>Farbmodelle</b>	RGB, Graustufen, S/W, indizierte Farben	RGB, Graustufen, indizierte Farben	RGB, Graustufen, S/W, CMYK, Lab, indizierte Farben (Volltonfarben)
<b>Kanäle</b>	1 oder 3 Farbkanäle	3 Farbkanäle, 1 Alpha-Kanal	1, 3 oder 4 Farbkanäle, 20 Alpha-Kanäle
<b>Bittiefe</b>	1-8 Bit/Kanal, 32 Bit/Pixel	1-16 Bit/Pixel	1-16 Bit/Kanal
<b>Kompression</b>	keine, RLE	LZ77 Variante	ohne, RLE, LZW, CCITT, JPEG
	verlustfrei	verlustfrei	verlustfrei, verlustbehaftet
	Universell einsetzbar, leider meist keine Kompression	Relativ gute verlustfreie Komprimierung, sowohl für Bilder mit viel als auch wenig Farben	Geringe Komprimierung, kaum für Einsatz im Internetbereich geeignet
<b>Entwickler</b>	Microsoft	MIT	Adobe
<b>Plattformen</b>	Mac, MS-Dos, Windows	Mac, Windows, Unix	Mac, Windows, Unix

### **3. Level of Detail 3**

#### **3.1 Vergleich der Auflösung von Farb- bzw. Helligkeitswerten**

3.1.1 Vorschlag interaktive Anwendung

3.1.2 Weiterführende Literatur und Links

Applet, zur Veranschaulichung der LZW-Kompression:

<http://www.cs.sfu.ca/cs/CC/365/li/squeeze/>

## 4. Literatur

### 4.1 Eigene Literatur

E01	<b>Photoshop 7.0 Kompendium</b>	
Untertitel	Für Windows und Macintosh	
URL		
Datum (URL)		
<b>Autor/ Autorin</b>		
Name(n)	<b>Heico Neumeyer</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von	<b>Markt+Technik Verlag</b>	
Veröffentlichungsdatum	2002	
Veröffentlicht in	<b>München</b>	
URL		
Datum (URL)		

E02	<b>Bildformate</b>	
Untertitel	Gegenüberstellung verschiedener Bildformate hinsichtlich ihrer Funktionsweise und Einsatzmöglichkeiten	
URL	<b><a href="http://www.scheib.info/downloads/Projekt_Bildformate.pdf">http://www.scheib.info/downloads/Projekt_Bildformate.pdf</a></b>	
Datum (URL)		
<b>Autor/ Autorin</b>		
Name(n)	<b>Paula Pfiester, Nina Tretter, Tina Koppenhöfer, Matthias Ecker</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von	<b>Fachbereich Informatik</b>	
Veröffentlichungsdatum	Februar 2002	
Veröffentlicht in		
URL		
Datum (URL)		



E03	<b>Kompendium der Mediengestaltung</b>	
Untertitel	Für Digital- und Printmedien	
URL		
Datum (URL)		
<b>Autor/ Autorin</b>		
Name(n)	<b>Joachim Böhringer, Peter Bühler, Patrick Schlaich, Hanns-Jürgen Ziegler</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von	<b>Springer-Verlag</b>	
Veröffentlichungsdatum	2003	
Veröffentlicht in	<b>Berlin Heidelberg</b>	
URL	<b><a href="http://www.springer.de">http://www.springer.de</a></b>	
Datum (URL)		

E04	<b>GRAPHICS: GIF files</b>	
Untertitel		
URL	<b><a href="http://www.webstyleguide.com/graphics/gifs.html">http://www.webstyleguide.com/graphics/gifs.html</a></b>	
Datum (URL)	<b>13.06.2004</b>	
<b>Autor/ Autorin</b>		
Name(n)	<b>Patrick J. Lynch, Sarah Horton</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von		
Veröffentlichungsdatum	2002	
Veröffentlicht in		
URL		
Datum (URL)		

E05	<b>LRZ: Grafikformate und deren Konvertierung</b>	
Untertitel	Übersicht über die wichtigsten Grafikformate, deren Definition, Eigenschaften und Verwendungszweck	
URL	<a href="http://www.lrz-muenchen.de/services/software/grafik/grafikformate">http://www.lrz-muenchen.de/services/software/grafik/grafikformate</a>	
Datum (URL)	<b>13.06.2004</b>	
<b>Autor/ Autorin</b>		
Name(n)	<b>Karl Weidner, Andreas Kelz</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von		
Veröffentlichungsdatum	24.7.1997	
Veröffentlicht in		
URL		
Datum (URL)		

E06	<b>JPEG JFIF</b>	
Untertitel		
URL	<a href="http://www.w3.org/Graphics/JPEG">http://www.w3.org/Graphics/JPEG</a>	
Datum (URL)	<b>13.06.2004</b>	
<b>Autor/ Autorin</b>		
Name(n)	<b>Chris Lilley</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von		
Veröffentlichungsdatum	13.2.1996	
Veröffentlicht in		
URL		
Datum (URL)		

E07	<b>Kapitel 2 : JPEG-Bilddatenkompression</b>	
Untertitel		
URL	<a href="http://i31www.ira.uka.de/docs/semin94/02_JPEG">http://i31www.ira.uka.de/docs/semin94/02_JPEG</a>	
Datum (URL)	<b>13.06.2004</b>	
<b>Autor/ Autorin</b>		
Name(n)	<b>Dietmar Baumstark</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von		
Veröffentlichungsdatum	02.05.1995	
Veröffentlicht in		
URL		
Datum (URL)		

E08	<b>JPEG File Layout and Format</b>	
Untertitel		
URL	<a href="http://www.funducode.com/freec/Fileformats/format3/format3b.htm">http://www.funducode.com/freec/Fileformats/format3/format3b.htm</a>	
Datum (URL)	13.06.2004	
<b>Autor/ Autorin</b>		
Name(n)		
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von	<b>Kanetkar's Institute of Computing and Information Technology</b>	
Veröffentlichungsdatum	05.07.2002	
Veröffentlicht in		
URL		
Datum (URL)	<b>13.06.2004</b>	

E09	<b>JPEG-Komprimierung</b>	
Untertitel		
URL	<a href="http://141.22.17.5/~lehnart/Bildverarbeitung/tis_jpeg/html/tis_010.html">http://141.22.17.5/~lehnart/Bildverarbeitung/tis_jpeg/html/tis_010.html</a>	
Datum (URL)	<b>13.06.2004</b>	
<b>Autor/ Autorin</b>		
Name(n)	<b>Thimo Nühs</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von	<b>Fachhochschule Hamburg – Fachbereich Elektrotechnik und Informatik</b>	
Veröffentlichungsdatum	1999	
Veröffentlicht in		
URL		
Datum (URL)		

E10	<b>digit online! Know-how / Imaging Technologie: JPEG2000</b>	
Untertitel		
URL	<a href="http://www.digit.de/know_how/qtvr/know_how_jpg2000.html">http://www.digit.de/know_how/qtvr/know_how_jpg2000.html</a>	
Datum (URL)	<b>13.06.2004</b>	
<b>Autor/ Autorin</b>		
Name(n)		
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von		
Veröffentlichungsdatum		
Veröffentlicht in		
URL		
Datum (URL)		

E11	<b>LZW compression algorithm</b>	
Untertitel		
URL	<a href="http://en.wikipedia.org/wiki/LZW">http://en.wikipedia.org/wiki/LZW</a>	
Datum (URL)	<b>14.06.2004</b>	
<b>Autor/ Autorin</b>		
Name(n)		
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von		
Veröffentlichungsdatum		
Veröffentlicht in		
URL		
Datum (URL)		

E12	<b>The Squeeze Page</b>	
Untertitel		
URL	<a href="http://www.cs.sfu.ca/cs/CC/365/li/squeeze/">http://www.cs.sfu.ca/cs/CC/365/li/squeeze/</a>	
Datum (URL)	<b>14.06.2004</b>	
<b>Autor/ Autorin</b>		
Name(n)		
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von		
Veröffentlichungsdatum		
Veröffentlicht in		
URL		
Datum (URL)		

E13	<b>[70] Introduction to data compression (long)</b>	
Untertitel		
URL	<a href="http://www.faqs.org/faqs/compression-faq/part2/section-1.html">http://www.faqs.org/faqs/compression-faq/part2/section-1.html</a>	
Datum (URL)		
<b>Autor/ Autorin</b>		
Name(n)	<b>Peter Gutmann</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von		
Veröffentlichungsdatum	07.06.2004	
Veröffentlicht in		
URL		
Datum (URL)		

E14	<b>Multimedia im Internet</b>	
Untertitel		
URL	<a href="http://www.tfh-berlin.de/~godberse/semesterarbeiten/ia2/MMkompression.pdf">www.tfh-berlin.de/~godberse/semesterarbeiten/ia2/MMkompression.pdf</a>	
Datum (URL)	14.06.2004	
<b>Autor/ Autorin</b>		
Name(n)	<b>Marc Mazarin, Christoph Steckelberg</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von		
Veröffentlichungsdatum		
Veröffentlicht in		
URL		
Datum (URL)		

E15	<b>Intro to PNG Features</b>	
Untertitel		
URL	<a href="http://www.libpng.org/pub/png/pngintro.html">http://www.libpng.org/pub/png/pngintro.html</a>	
Datum (URL)	15.06.2004	
<b>Autor/ Autorin</b>		
Name(n)	<b>Greg Roelofs</b>	
URL		
Datum (URL)		
<b>Veröffentlichung</b>		
Veröffentlicht von		
Veröffentlichungsdatum	01.06.2004	
Veröffentlicht in		
URL		
Datum (URL)		