



Verlustlose Bildkompression

Lossless Image Compression

Kai Uwe Barthel, FHTW Berlin

Zusammenfassung Für die verlustbehaftete Kompression von Bildern existieren leistungsfähige Algorithmen. Je nach Bildtyp und visuellen Qualitätsanforderungen lässt sich die Bilddatenmenge um den Faktor 5 bis 100 reduzieren. Die verlustlose Kompression von Bilddaten hingegen gestaltet sich sehr viel schwieriger. Standardtools wie *Compress* oder *Zip* versagen hierbei vollständig, selbst neue sehr aufwändige Verfahren erzielen typischerweise nur Kompressionsfaktoren zwischen zwei und drei. Dieser Beitrag erläutert zunächst die Funktionsweise und Ergebnisse von Standardverfahren zur verlustlosen Bildkompression, im Anschluss werden die Prinzipien

und Ergebnisse neuerer Verfahren zur verlustlosen Bildkompression vorgestellt und diskutiert. ▶▶▶ **Summary** For lossy compression of images there are powerful algorithms. Depending on the image type and the visual needs compression factors between 5 and 100 can be achieved. Lossless compression of images however is much harder. Standard compression tools like *Compress* or *Zip* do fail on typical images. However even sophisticated algorithms only can achieve moderate compression factors in the range from 2–3. This article does explain the principle of standard techniques for lossless image compression. Finally ideas of newer algorithms are discussed.

KEYWORDS 1.4.2 [Image Processing] Lossless Image Compression, Context Modelling

1 Einleitung

Digitale Bilder werden heutzutage in nahezu allen technischen Bereichen eingesetzt. Neben dem Internet, in dem es kaum noch Webseiten ohne digitale Bilder gibt, erlebt insbesondere die digitale Fotografie einen rasanten Aufschwung. Auch in anderen Bereichen wie der Archivierung von (Bild-)Dokumenten oder der technischen Qualitätskontrolle setzen sich digitale Bilder immer weiter durch. Werden an digitale Bilder die gleichen Qualitätsansprüche wie an die klassische Fotografie gestellt, so entstehen sehr große Datenmengen im Bereich von mehreren Megabytes pro Bild. Für eine Speicherung oder Übertragung dieser Bilder ist eine effiziente Kompression unabdingbar.

Die Anforderungen an die visuelle Qualität von digitalen Bildern sind höchst unterschiedlich. Im Internet steht meist eine effiziente Übertragung der Bilddaten im Vordergrund, die sich nur durch eine

starke verlustbehaftete Kompression erzielen lässt, wodurch die Bildqualität typischerweise eher gering ist. Andere Einsatzfelder verlangen hingegen eine verlustlose Kompression der Bilder. Hierzu gehören juristische oder medizinische Anwendungen, die eine Veränderung der Bilddaten verbieten. Weiterhin sind Bilder im kulturellen/künstlerischen Produktionsbereich oder bei der Archivierung von digitalen Fotos von historischen Dokumenten oder Objekten zu nennen, da zum Zeitpunkt der Kompression oft noch nicht feststeht, welche weiteren Bearbeitungsschritte mit diesen Bildern durchgeführt werden, bzw. welche Anforderungen zukünftige Betrachter an diese Bilder haben werden.

Dieser Artikel geht kurz auf die Grundlagen digitaler Bilder ein, um dann zunächst das allgemeine Prinzip der Kompression von Bildern zu erläutern. Es wird gezeigt, dass die verlustlose Bildkompression ähnliche Methoden wie verlustbehaftete

Bildkompressionsverfahren einsetzt, wobei jedoch einige spezielle Randbedingungen zu beachten sind. Im Anschluss werden klassische Verfahren zur verlustlosen Bildkompression beschrieben, um dann die Prinzipien und Ergebnisse neuerer verlustloser Bildkompressionsalgorithmen vorzustellen.

2 Grundlagen digitaler Bilder

Werden digitale Bilder mit einem Scanner von Papiervorlagen erzeugt oder direkt mit einer digitalen Kamera aufgenommen, so wird die Genauigkeit der Bilderfassung über drei Parameter gesteuert:

Die *Auflösung* legt fest, wie viele Bildpunkte (Pixel) pro Längeneinheit erzeugt werden ($ppi = \text{pixel per inch}$ oder oft auch $dpi = \text{dots per inch}$). Typische Auflösungen von digitalen Bildern liegen zwischen 50–150 dpi für Bildschirmbilder bis hin zu 300–2400 dpi für hochwertige Ausdrücke. Abhängig von den Abmessungen der Bilder entstehen

hierbei Werte von wenigen hundert bis hin zu einigen tausend Pixeln pro Seitenlänge eines Bildes.

Die Zahl der (Farb-)kanäle entscheidet, wie das Lichtspektrum zerlegt und aufgenommen wird. Für *Graustufenbilder* ist ein einzelner Kanal ausreichend. Bedingt durch die Tatsache, dass das menschliche Auge drei Sinneszellen für die Farbwahrnehmung besitzt, können *Farbbilder* mittels dreier Farbkanaäle (Rot, Grün und Blau (RGB)) beschrieben werden. Im Druckbereich werden meist vier bis sechs Kanäle verwendet. Digitale Bilder, die auch nicht sichtbare Daten aus dem Infrarot- bzw. Ultraviolettbereich beinhalten, verwenden bis zu 256 Kanäle und werden als *Hyperpektralbilder* bezeichnet.

Die *Farbtiefe* legt schließlich fest, mit wie vielen Bits pro Pixel die digitalisierten Daten gespeichert werden. Die Einheit hierfür ist *bpp* (bits per pixel). Übliche Werte liegen zwischen 1 bis 16 Bit pro Kanal.

Die am häufigsten verwendeten Bildtypen sind: *Bitonale Bilder* bzw. *Binärbilder* – wie beispielsweise Faxseiten – sie verwenden einen Kanal mit nur einem Bit pro Pixel. *Graustufenbilder* besitzen ebenfalls nur einen Kanal, wobei die Intensitäten mit 8 bis 16 Bits repräsentiert werden. Als *Echtfarbbilder* (engl: *true color images*) werden Bilder bezeichnet, die mindestens 8 Bits pro Kanal, d.h. eine Farbtiefe von mindestens 24 bpp besitzen.

3 Bildkompressionsarten

Bei der Kompression von digitalen Bildern charakterisieren zwei Größen die Qualität des Kompressionsverfahrens:

Der *Kompressionsfaktor* ist das Verhältnis aus der Originalgröße der Bilddatei und der Größe der komprimierten Datei. Alternativ wird der resultierende Bits-pro-Pixel-Wert angegeben. Weiterhin ist die Qualität der dekomprimierten Bilddatei zu nennen. Objektiv wird diese Qualität meist über den Fehler zwischen dem decodierten Bild und dem Originalbild definiert (z. B.

über den *MSE mean squared error*, den mittleren quadratischen Fehler). Da objektiv gemessene Qualitäten oft nur wenig mit dem visuellen Qualitätsempfinden korreliert sind, werden häufig auch subjektive Qualitätsmessungen durchgeführt (z. B. *MOS mean opinion score*). Beim Vergleich unterschiedlicher Kompressionsverfahren ist Sorge zu tragen, dass die Kompressionsrate bei gleicher Qualität verglichen wird.

Grundsätzlich wird die *verlustbehaftete* (engl. *lossy*) Kompression und die *verlustlose* (engl. *lossless*) Kompression unterschieden. Verlustlose Kompression bedeutet, dass nach der Dekompression alle Pixel in jedem Bit identisch zu ihren Originalwerten sind. Somit entfällt bei der verlustlosen Bildkompression die Bestimmung einer Bildqualität, da die Rekonstruktion fehlerfrei ist. Als Qualitätsmerkmal einer verlustlosen Kompression ist daher zunächst nur der erzielbare Kompressionsfaktor zu nennen; zwei sekundäre Beurteilungskriterien sind der Speicherbedarf und die Komplexität bzw. die Ausführungsdauer des Algorithmus.

Neben der „echten“ verlustlosen Kompression ist häufig von *visuell verlustlosen* Verfahren zu hören. Dieser Begriff ist durch die Industrie geprägt, die zu suggerieren versucht, dass ihr jeweiliges verlustbehaftetes Verfahren so gut ist, dass Fehler – sprich Codierungsartefakte – nicht sichtbar sind. Grundsätzlich kön-

nen verlustbehaftete Kompressionsverfahren Bilder so komprimieren, dass Fehler visuell nicht zu erkennen sind, allerdings ist es sehr stark von der Darstellung der Bilder (z. B. Druck oder Bildschirm) und den Erfahrungen des Betrachters abhängig, ab wann Fehler sichtbar werden. Des Weiteren ist es möglich, dass Fehler zunächst unsichtbar sind, jedoch durch spätere Bildmanipulationen wie Größenänderungen oder z. B. Kontrastverstärkungen wieder sichtbar werden.

Near-lossless-Codierungsverfahren komprimieren Bilder derart, dass eine bestimmte Grenze für den maximalen Fehler nicht überschritten wird. Eine Fehlerschwelle ε von 2 bedeutet beispielsweise, dass alle Pixelwerte des Bildes nach der Dekompression um maximal – 2 bis 2 Intensitätsstufen von ihrem Originalwert abweichen.

Ein weiteres wichtiges Unterscheidungsmerkmal zwischen Codierungsverfahren ist die *Progressivität*. Hiermit wird die Möglichkeit beschrieben, bereits aus einem Teil der komprimierten Daten eine Ansicht zu erzeugen, die von geringerer Auflösung oder Qualität sein kann.

Grundsätzlich ist das Maß der Komprimierbarkeit abhängig von der Auflösung der Bilder. Bei sehr geringer Auflösung sind benachbarte Pixelwerte wenig korreliert, was eine Kompression erschwert. Steigt die Auflösung, werden sich benachbarte Pixel ähnlicher, was

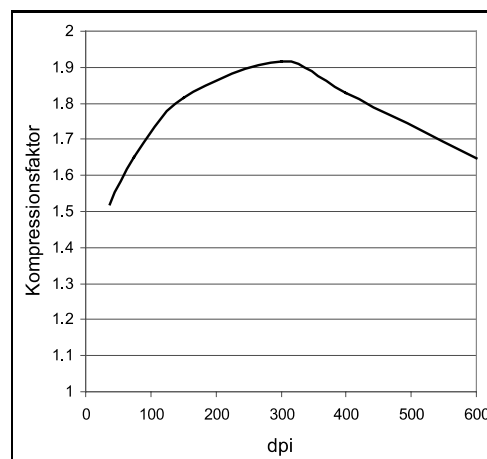


Bild 1 Abhängigkeit der Komprimierbarkeit von der Auflösung. Typischer Verlauf für ein gescanntes Foto.

die Komprimierbarkeit erhöht. Ist die Auflösung jedoch zu hoch, so dass Filmrauschen oder Druckartefakte im Bild sichtbar werden, so nimmt die Korrelation benachbarter Pixel wieder ab. Bild 1 zeigt für ein gescanntes Foto den typischen Verlauf der verlustlosen Komprimierbarkeit in Abhängigkeit von der Auflösung.

4 Prinzip der Bildkompression

Werden bei der Digitalisierung eines Bildes die Pixelintensitäten auf N unterschiedliche Werte quantisiert, so lassen sich diese mit $\lceil \log_2(N) \rceil$ Bits pro Pixel beschreiben. Somit können z. B. mit 8 Bits 256 Intensitäten oder Farben unterschieden werden.

Shannon entwickelte 1948 das Quellencodierungstheorem [1], mit dem sich die minimale Rate zur Übertragung von N statistisch unabhängigen Symbolen bestimmen lässt. Ein digitales Bild mit N unterschiedlichen Intensitäten i mit der jeweiligen Wahrscheinlichkeit p_i wird nach dem Quellencodierungstheorem dann optimal codiert (d. h. maximal komprimiert), wenn jeder Intensität i , ein Code der Länge $-\log_2(p_i)$ zugeordnet wird. Wird eine solche Codierung verwendet, so ergibt sich als durchschnittliche Bitzahl pro Symbol (bzw. Pixel):

$$H = - \sum_{i=1}^N p_i \log_2(p_i).$$

Diese mittlere Bitzahl wird auch als *Entropie* H dieser (Bild-)Quelle bezeichnet. Unabhängig davon, wie ein solcher Code zu erzeugen ist, gibt die Entropie die theoretische untere Schranke an, mit der dieses Bild codiert werden kann, wenn jeder Pixel individuell codiert wird. Wird die Entropie z. B. für die Pixelwerte realer 8-Bit-Graustufenbilder bestimmt, so ist dieser Wert eher ernüchternd, da sich typischerweise Werte zwischen 6 und 8 Bits ergeben, was Kompressionsfaktoren zwischen 1,33 und 1 entspricht. Zum Glück bedeutet dies nicht das

Ende aller Kompressionsbemühungen, denn dieser Wert trägt nicht der Tatsache Rechnung, dass Intensitätswerte von Pixeln typischerweise hoch korreliert sind und Bilder Redundanzen beinhalten, die durch eine Pixel-individuelle Codierung nicht ausgenutzt werden. D. h. Pixelwerte „normaler“ Bilder sind nicht statistisch unabhängig, daher lassen sich höhere Kompressionsfaktoren erzielen.

Gelingt es, die Pixelwerte des Bildes vollständig zu dekorrelieren, so beschreibt die Entropie dieses nun redundanzbefreiten Bildes eine mögliche untere Grenze, die angibt, welcher Kompressionsfaktor maximal erreichbar ist. Hierbei gilt es zwei Probleme zu lösen: Als erstes muss versucht werden, dem Bild jegliche inhärente Redundanz zu entziehen. Dies ist schwierig, da die genaue Art der Abhängigkeit der Pixelwerte untereinander, die sich auch lokal ändern kann, unbekannt ist. Das zweite Problem besteht darin, anschließend einen Code – angepasst an die Wahrscheinlichkeiten der verbleibenden Symbole – zu konstruieren, der eine resultierende durchschnittliche Bitrate erzeugt, die der zuvor bestimmten Entropie möglichst nahe kommt.

Für die Redundanzbeseitigung gibt es im Wesentlichen vier unterschiedliche Prinzipien, die einzeln, aber auch miteinander kombiniert angewendet werden können:

- Als erstes sind *Wörterbuchverfahren* zu nennen, die versuchen, Teile des Bildes durch identische, bereits übertragene Teile zu beschreiben. Ist diese Beschreibung effizienter als eine direkte Übertragung der Pixelwerte, so wird eine Kompression erreicht.
- Eine zweite Methode besteht darin, eine *Prädiktion* (Vorhersage) für den aktuell zu codierenden Pixelwert zu treffen. Hierbei braucht nur der Prädiktionsfehler codiert werden. Ist diese Vorhersage gut, so wird die Entropie des Prädiktionsfehlers geringer als die der Originalwerte sein.
- Die Anwendung einer dekorrelierenden *Transformation* auf Pixelblöcke konzentriert die Energie eines Pixelblocks auf wenige Transformationskoeffizienten, wodurch die Entropie verringert wird.
- Die letzte Methode besteht darin, durch eine *Kontextbildung* die bedingten Wahrscheinlichkeiten der Pixelwerte zu modellieren. Gelingt es, die Wahrscheinlichkeit des Werts zu erhöhen, so wird die Entropie verringert.

Prinzipiell lassen sich die meisten Bildcodierungsverfahren nach dem Blockschaltbild in Bild 2 darstellen.

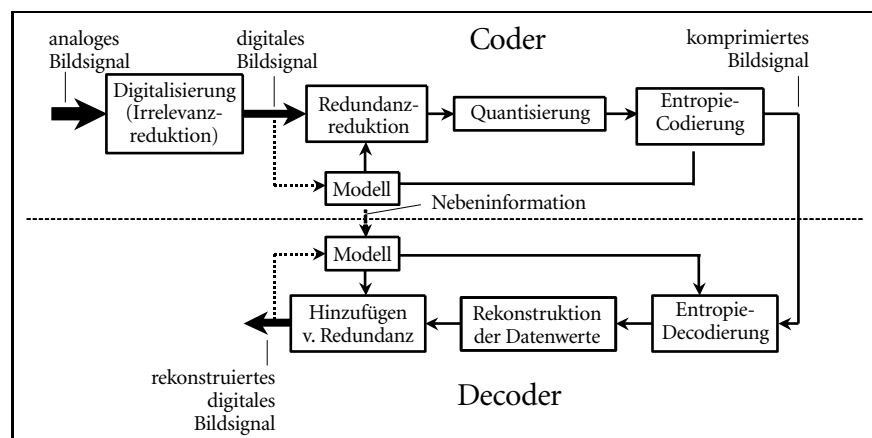


Bild 2 Blockschaltbild eines Bildkompressionsverfahrens.

Nachdem bei der Digitalisierung und eventuellen Vorverarbeitungsschritten nur die relevanten Daten des digitalen Bildes erhalten geblieben sind, wird dem digitalen Bild die innewohnende Redundanz entzogen. Dies geschieht – wie bereits dargestellt – meist durch eine Prädiktion der Pixelwerte und/oder eine Transformation der Bilddaten. Typischerweise stellt der Schritt der Redundanzbefreiung eine lineare Operation dar, die durch eine entsprechende inverse Operation beim Decoder wieder aufgehoben werden kann. Im Fall der verlustlosen Bildkompression muss die Berechnung der Redundanzreduktion unbedingt mit einer ganzzahligen Integer-Arithmetik realisiert werden, da die Verwendung einer *floating-point*- bzw. Gleitkomma-Arithmetik bei der Rekonstruktion Rundungsfehler erzeugen kann, die eine verlustlose Rekonstruktion unmöglich machen.

Bei verlustbehafteten Kompressionsverfahren erfolgt als nächster Schritt eine *Quantisierung* der transformierten Bilddaten bzw. des Prädiktionsfehlers. Dies ist der Punkt, an dem die Bilddaten gezielt so verändert werden, dass sich eine geringere Entropie für die verbleibenden Symbole ergibt und diese sich daher einfacher codieren lassen. Dadurch, dass die Daten durch die Quantisierung verfälscht werden, ist eine fehlerfreie bzw. verlustlose Rekonstruktion nicht mehr möglich. Daher darf bei der verlustlosen Bildkompression keine Quantisierung durchgeführt werden.

Als letzter Kompressionsschritt wird eine *Entropiecodierung* durchgeführt. Hierbei wird versucht, einen Code zu bestimmen, dessen durchschnittliche Codelänge möglichst nahe an die Entropie kommt. Eine besondere Effizienzsteigerung lässt sich erzielen, wenn es gelingt, die Symbole möglichst gut über ihre bedingten Wahrscheinlichkeiten zu modellieren. Diese Modellierung erfolgt über *Kontexte*, die im folgenden Abschnitt ausführlicher beschrieben werden.

Soll jedem Symbol ein individuelles Codewort so zugewiesen werden, dass die mittlere Codelänge minimal wird, so lässt sich ein solcher Code z. B. mit der *Huffman-Codierung* erzeugen. Hierbei sind allerdings keine Codes möglich, die weniger als 1 Bit pro Symbol verwenden. Dies ist insbesondere dann kritisch, wenn es Symbole mit Wahrscheinlichkeiten größer 0,5 gibt. Hier schafft die *Arithmetische Codierung* [2] Abhilfe, die jeweils mehrere Symbole gemeinsam codiert und es somit schafft, der Entropie beliebig nahe zu kommen.

Die genaue Methode der Redundanzreduktion und die Beschreibung der (bedingten) Wahrscheinlichkeiten bei der Entropiecodierung beschreibt ein *Modell*. Ein *konstantes Modell* kann beispielsweise an einem Set von Beispielbildern optimiert worden sein, seine Modelldaten sind dann Coder und Decoder a priori bekannt. Im Falle eines *adaptiven Modells* wird versucht, eine bessere Anpassung an lokale Änderungen der Statistik des Bildes zu erreichen. Werden zur Bestimmung der Modellparameter ausschließlich kausale (bereits codierte) Bildwerte verwendet, so spricht man von einer *rückwärtsgesteuerten Adaption*. Werden auch noch nicht codierte Bildwerte zur Modelladaption verwendet, so ist dies eine *vorwärtsgesteuerte Adaption*, bei der bestimmte Modellparameter zusätzlich als *Nebeninformation* übertragen werden müssen und somit dem eigentlichen Ziel der Kompression entgegenwirken. Verfahren, die zunächst das gesamte Bild analysieren, um daraus Modellparameter zu extrahieren, um dann in einem zweiten Durchgang das Bild zu komprimieren, werden als *two-pass*-Verfahren bezeichnet. Durch die Analyse des gesamten Bildes entsteht der Nachteil eines höheren Speicherbedarfs und einer verzögerten Codierung.

Die jeweils eingesetzten Bildkompressionsverfahren unterscheiden sich abhängig davon, welcher Bildtyp zu komprimieren

ist. Bilder mit nur wenigen Farben/Intensitäten lassen sich häufig schon sehr gut mit speziell angepassten Entropiecodierungsverfahren komprimieren. Diese Verfahren werden im nächsten Abschnitt beschrieben.

5 Elementare verlustlose Komprimierungstechniken

Im Folgenden seien zunächst am Beispiel eines einfachen Bildes mit nur drei unterschiedlichen Intensitätswerten (entsprechend Bild 3) die unterschiedlichen Prinzipien elementarer verlustloser Codierungstechniken beschrieben. Die Reihenfolge der Darstellung orientiert sich an der erzielbaren Kompression, die jeweils in Bits pro Pixel (fett dargestellt) angegeben wird.

Binärcodierung. Läge dieses Bild als Graustufenbild vor, so wären **8 Bits pro Pixel** zur Repräsentation nötig. Da im Bild allerdings nur drei unterschiedliche Intensitäten vorkommen, kann eine Binärcodierung mit $\lceil \log_2(3) \rceil = 2$ **Bits pro Pixel** verwendet werden.

Huffman-Codierung. Zunächst werde für dieses Bild angenommen, dass die Pixelwerte statistisch unabhängig seien. Mit Wahrscheinlichkeiten von 0,5 für die weißen Pixel und jeweils 0,25 für die grauen Pixelwerte ergibt sich eine Entropie $H = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{4} \cdot 2 = 1,5$ Bits pro Pixel. Dieser Wert lässt sich erreichen, indem für die Pixelwerte Codes variabler Länge verwendet

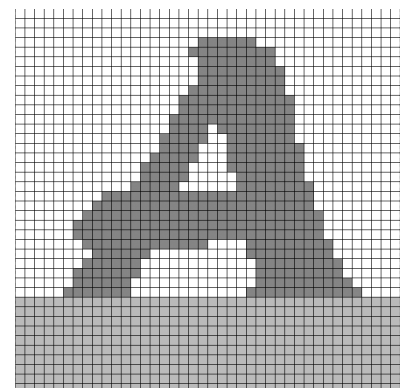


Bild 3 Beispielbild mit drei Intensitäten.

werden. Eine Möglichkeit zur Generierung von Codes variabler Länge bietet die *Huffman-Codierung*. Sind die Wahrscheinlichkeiten als Zweierpotenzen darstellbar, so lässt sich ein Huffman-Code konstruieren, der diese Entropie exakt erreicht. Ein möglicher Code wäre in diesem Fall z. B. eine 1 für Weiß und die Codes 01 und 00 für die beiden Grautöne. Hiermit ergäbe sich eine mittlere Coderate von **1,5 Bits pro Pixel**, was der Entropie entspricht. Dieser Wert ist gleichzeitig das Minimum, das nicht unterschritten werden kann, solange Abhängigkeiten benachbarter Pixel nicht beachtet bzw. ausgenutzt werden.

Man erkennt im Beispielbild, dass die Pixelwerte korreliert sind, sehr häufig sind benachbarte Pixelwerte identisch. Hier ist es die Aufgabe eines geeigneten Modells, diese Abhängigkeiten möglichst gut zu beschreiben. Gelingt dies, so kann eine stärkere Kompression erzielt werden. Im Folgenden sollen einige typische Ansätze hierzu beschrieben und verglichen werden.

Gruppierung von Pixelwerten. Ein möglicher einfacher Ansatz besteht darin, jeweils zwei Pixel gemeinsam zu codieren. Bestimmt man die Entropie über die Wahrscheinlichkeiten der neun möglichen Pixelpaare, ergibt sich ein Wert von 1,77 Bits pro Pixelpaar, was einer Entropie von 0,885 Bits pro Pixel entspricht. Dieser Wert ließe sich durch Verwendung eines arithmetischen Coders erreichen. Mit einer Huffman-Codierung ergäbe sich ein Wert von **0,935 Bits pro Pixel**.

Im Prinzip könnte dieser Ansatz – mehrere Pixel zu (größeren) Blöcken zusammenzufassen – immer weitergeführt werden, dies ist jedoch nicht sinnvoll. Werden zu viele Pixel zusammengefasst, so ergeben sich zu viele mögliche Kombinationen für diese Blöcke, für die dem Decoder die jeweiligen Wahrscheinlichkeiten bzw. Codes mitgeteilt werden müssen. Diese Nebeninformation kann schnell so groß werden,

dass sie die Kompression zunichtemacht. Der Extremfall wäre erreicht, würden alle Pixel des Bildes zu einem Block zusammengefasst. Für diesen Block wäre die Wahrscheinlichkeit 1, was einer Entropie von 0 Bits entspräche, allerdings müsste dieser Block jetzt vollständig als Nebeninformation übertragen werden.

„*Wörterbuchverfahren*“. Eine weitere Gruppe von Kompressionsverfahren stammt von der Textkompression ab. Hierbei sind *LZW*, *Compress* und *Zip* als typische Vertreter zu nennen. Bei diesen „*Wörterbuchverfahren*“ beruht die Kompression darauf, dass sich zu codierende Symbolfolgen teilweise wiederholen. Diese Wiederholungen können effizient über Pointer (bestehend aus Offset und Länge) auf bereits codierte Symbolfolgen oder durch einen Index auf einen Eintrag einer dynamisch aufgebauten Bibliothek der bereits übertragenen Symbolfolgen beschrieben werden. Diese Verfahren, die für Texte sehr gut funktionieren, versagen sehr häufig bei Bildern, da hier fast nie längere identische Symbolfolgen auftreten. Für unser einfaches Beispielbild ergibt sich mit einer Zip-Codierung eine Rate von etwa **0,7 Bits pro Pixel**.

Lauf längencodierung. Insbesondere bei Bildern mit wenigen hochkorrelierten Intensitätsstufen bietet sich die *Lauf längencodierung* (engl. *run length encoding (RLE)*) als einfaches Verfahren an. Aufeinanderfolgende Pixel mit identischen Werten werden zu Läufen zusammengefasst, für die dann jeweils der Intensitätswert und die Anzahl der Pixel (die Lauflänge) übertragen werden. Ein einfacher Code könnte in unserem Beispielbild aus 2 Bits zur Beschreibung der Intensitätswerte und 5 Bits für die Lauflänge (von 1 bis 32) bestehen und würde zu einer Bitrate von **0,433 Bits pro Pixel** führen. Eine weitere Steigerung der Kompression kann erreicht werden, wenn die Symbole der Lauf längencodierung selbst noch

einmal entropiecodiert werden. Eine Lauf längencodierung macht dann keinen Sinn, wenn es zu viele unterschiedliche Intensitätswerte bzw. zu kurze Lauf längen gibt.

Kontextmodellierung. Die absoluten Wahrscheinlichkeiten bestimmter Pixelwerte beschreiben ihre globale Häufigkeit im Bild, sie sagen aber nichts über die lokalen Abhängigkeiten der Pixel untereinander aus. Soll die lokale Abhängigkeit der Pixelwerte ausgenutzt werden, so ist es sinnvoll, die Wahrscheinlichkeit für einen bestimmten Pixelwert abhängig vom Kontext seiner örtlichen Umgebung auszudrücken. Aufgrund der starken Abhängigkeit eines Pixelwerts von seiner örtlichen Umgebung ist es sinnvoller, eine Entropiecodierung unter Verwendung der *bedingten* anstelle der *absoluten Wahrscheinlichkeiten* durchzuführen, da die bedingten Wahrscheinlichkeiten eine bessere Modellierung des zu codierenden Pixelwertes ermöglichen.

Wird in unserem Beispielbild beispielsweise der Intensitätswert des vorangegangenen Pixels als Kontext verwendet, so ergeben sich drei mögliche Kontexte. Durch die Verwendung der bedingten Wahrscheinlichkeiten ergeben sich für jeden Kontext deutlich reduzierte Entropien. Beispielsweise ist die bedingte Wahrscheinlichkeit für die grauen Pixel im unteren Bereich des Beispielbildes gleich 1. In diesem Beispiel ergäbe sich bei Verwendung eines arithmetischen Coders eine theoretische Rate von etwa **0,265 Bits pro Pixel**.

6 Modellbildung bei Binärbildern

Im Folgenden soll die Idee der leistungsfähigen Kontextmodellierung etwas detaillierter beschrieben werden, wobei zunächst Binärbilder behandelt werden sollen. Diese Binärbilder könnten z. B. bitonale Faxdokumente sein oder aber auch durch die im Folgenden beschriebene Zerlegung von Graustufen oder Farbbilder in Bitebenen entstanden sein.

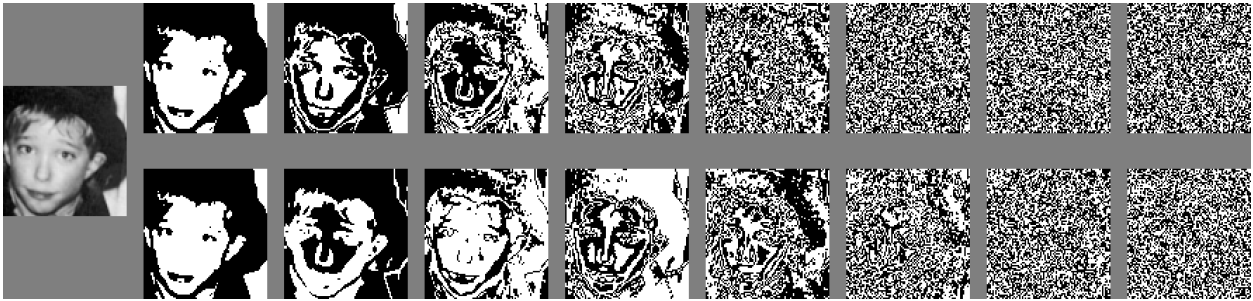


Bild 4 Zerlegung eines 8-Bit-Graustufenbildes in seine Bitebenen, oben: Binär-, unten: Graycodierung der Pixelwerte.

Zerlegt man die Pixelwerte von Graustufen- oder Farbbildern in die einzelnen Bits, so lässt sich ein Bild mit m Bits pro Pixelwert auch durch m Binärbilder repräsentieren. Diese binären Bilder werden als *Bitebenen* (engl. *bitplanes*) bezeichnet. Somit können Graustufen oder Farbbilder auch dadurch verlustlos codiert werden, dass ihre binären Bitebenenbilder jeweils verlustlos codiert werden. Abbildung 4 zeigt die Zerlegung eines 8-Bit-Graustufenbildes in seine acht Bitebenen. Bei den höherwertigen Bits sind klar die Bildstrukturen zu erkennen, wohingegen die niederwertigen Bitebenen eher zufällig wirken. Werden Bilder zur verlustlosen Codierung in Bitebenen zerlegt, so ist eine *Graycodierung* der Pixelwerte immer einer Binärcodierung vorzuziehen, da eine Intensitätsänderung um eine Stufe immer auch nur ein Bit verändert. Bei Repräsentation der Pixelwerte mit einer Binärcodierung hingegen ändern sich meist mehrere Bits. Hierdurch lassen sich mit Graycodierten Bildern bessere Kompressionsergebnisse erzielen.

In Bild 4 ist ebenfalls zu erkennen, wie schwer es ist, eine starke verlustlose Kompression zu erzielen. Wird vereinfachend angenommen, dass die vier höherwertigen Bitebenen extrem redundant und somit unendlich gut komprimierbar sind, wohingegen die vier niederwertigen Bits rein zufällig und daher unkomprimierbar sind, würde sich als Entropie ein Wert von $4 \cdot 0 + 4 \cdot 1 = 4$ Bits ergeben. Dies entspräche gerade einem Kompressionsfaktor von zwei. Dieser Faktor verschlech-

tert sich bei Bildern mit höherer Bittiefe.

Da bei Binärbildern nur zwei Symbole zu codieren sind, lässt sich kein Kompressionsgewinn erzielen, wenn jeweils pro Symbol (binärem Pixel) ein Codewort verwendet wird, da die minimale Länge eines Codeworts bereits ein Bit ist. Daher müssen die Pixel entweder gemeinsam in Gruppen codiert werden oder es kann z.B. ein arithmetischer Coder für die Entropiecodierung verwendet werden.

Kontextbildung. Praktisch erfolgt eine Modellbildung der Abhängigkeiten der Pixel untereinander durch die Verwendung von *Kontexten*. Ein Kontext bezeichnet eine bestimmte Konstellation einer begrenzten Menge von benachbarten, bereits codierten Pixeln. Durch die Kontextbildung kann eine Verbesserung der Kompression dann erzielt werden, wenn sich der Wert des zu codierenden Pixels möglichst gut vorhersagen lässt, d.h. wenn die Wahrscheinlichkeit für das zu codierende Symbol erhöht wird. Hierbei gilt

$$H(x) \leq H(x|K),$$

d.h. durch die Verwendung von geeigneten Kontexten K lässt sich die Entropie verringern. Ziel ist, die Wahrscheinlichkeiten der Pixelwerte für unterschiedliche Kontexte möglichst gut zu differenzieren. Werden beispielsweise vier Pixel für die Kontextbildung verwendet (wie z.B. in Bild 5: obere Zeile, Mitte), so ergeben sich $2^4 = 16$ unterschiedliche Kontexte. Für jeden dieser Kontexte wird im Laufe der Codierung

die Häufigkeit der Nullen bzw. der Einsen für den gerade codierten Pixel protokolliert (gezählt), sodass hieraus – bei Coder und Decoder – eine Schätzung der Wahrscheinlichkeiten für eine 0 oder 1 des aktuellen Pixels „gelernt“ werden kann. Sind z.B. alle vier Kontextpixel gleich 0, so wird die Wahrscheinlichkeit, dass der zu codierende Pixelwert ebenfalls 0 sein wird, sehr hoch sein.

Lässt man den Einfluss älterer Protokollbeiträge im Laufe der Zeit abklingen, so wird hierdurch eine adaptive Wahrscheinlichkeitschätzung realisiert, die auf lokale schwankende Wahrscheinlichkeiten reagieren kann. Durch eine Vergrößerung der Kontextregion ergeben sich mehr mögliche Kontexte, was eine verbesserte Modellierung der Wahrscheinlichkeiten ermöglicht. Zu große Kontextregionen sind jedoch problematisch, da sehr schnell eine extrem große Anzahl von Kontexten entsteht, die größer als die Pixelanzahl des Bildes sein kann. Wenn bei der Codierung ein bestimmter Pixel mit einem speziellen Kontext codiert wird, kann es passieren, dass dieser Kontext noch nie aufgetreten ist und daher keine sinnvollen Wahrscheinlichkeiten geschätzt werden können.

Bild 5 zeigt optimale Kontextformen für unterschiedliche Kontextgrößen, die aus 80 typischen Testbildern generiert worden sind [3]. Hierbei ist der zu codierende Pixel der Punkt mit zwei Ringen. Schwarze Pixel sind bereits codiert und stellen die Kontextregion dar, die weißen Pixel sind ebenfalls bereits codiert, werden aber nicht zur Kontextbildung verwen-

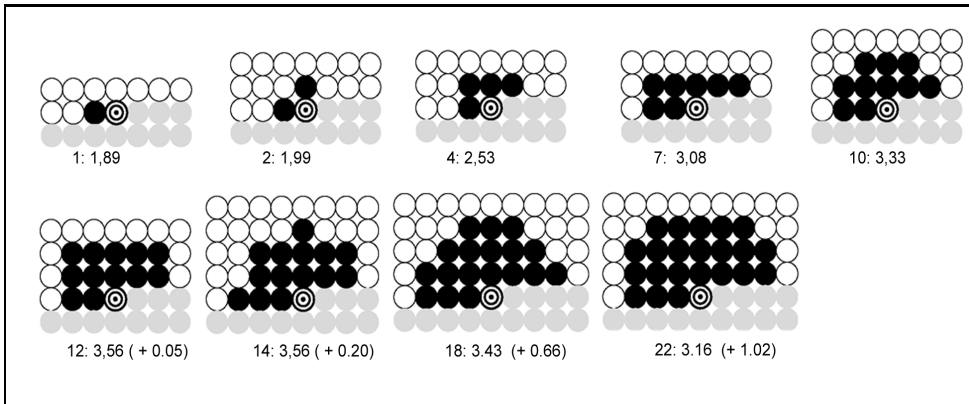


Bild 5 Optimale Kontextformen für 1 bis 22 Kontextpixel, 1. Zahl: Kontextgröße, 2. Zahl: erzielbarer Kompressionsfaktor (+ theoretischer Gewinn bei bekannten Wahrscheinlichkeiten) (nach [3]).

det. Die grauen Pixel sind noch nicht codiert worden und da sie dem Decoder nicht zur Verfügung stehen, können sie nicht mit für die Kontextbildung verwendet werden. Unter den einzelnen Abbildungen ist jeweils die Kontextgröße und der erzielbare Kompressionsfaktor angegeben. Man erkennt, dass der Kompressionsfaktor für zu große Kontextregionen wieder abnimmt. Die Zahlen in Klammern stellen die theoretische Zunahme des Kompressionsfaktors dar, der sich ergäbe, wenn alle Wahrscheinlichkeiten vor Beginn der Codierung bekannt wären. Diese Faktoren sind in der Praxis jedoch nicht erreichbar, da die zu übertragende Nebeninformation zu groß wäre.

Im JBIG1 Standard wird dieses Kontext-basierte Verfahren zur Codierung von binären Bildern eingesetzt [13]. Hierbei wird eine Kontextgröße von 10 Pixeln verwendet, für die Entropiecodierung wird eine Variante des Q-Coders, ein beschleunigter binärer arithmetischer Coder, verwendet. In praktischen Implementierungen – wie bei JBIG1 – ist eine klare Trennung zwischen der Modellierung der Bilddaten und der Entropiecodierung nicht immer exakt möglich. Häufig werden diese beiden Schritte aus Effizienzgründen gemeinsam realisiert.

7 Modellbildung bei Graustufen- und Farbbildern

Soll das Prinzip der Kontextbildung für Binärbilder auf Graustufen oder Farbbilder erweitert werden,

so ist dies nicht ohne weiteres möglich. Selbst bei kleinen Kontextgrößen entstünden extrem viele Kontexte, die jeweils eine Vielzahl von Wahrscheinlichkeiten für die unterschiedlichen Pixelwerte erlernen und modellieren müssten. Als erster Schritt wird daher stets versucht, den Wertebereich der zu codierenden Pixelwerte zu reduzieren. Wie bereits dargestellt, kann hierfür eine Prädiktion oder eine Transformation verwendet werden. Bild 6 vergleicht ein Originalbild und die Prädiktionsfehlerbilder bei Verwendung unterschiedlicher Prädiktoren. Abhängig vom verwendeten Prädiktor gelingt es, die Entropie deutlich zu reduzieren. Hierbei ist es wiederum möglich, die Prädiktion selbst adaptiv zu gestalten und zu versuchen, eine optimale Prädiktion aus bereits codierten Pixeln abzuschätzen. Bild 6d zeigt das Prädiktionsfehlerbild eines adaptiven Prädiktors.

Umkehrbare Farbtransformation. Bei Farbbildern wird man darüber hinaus versuchen, die Abhängigkeiten zwischen den Farbkanälen durch eine umkehrbare Farbtransformation zu reduzieren. Die folgende Gleichung zeigt ein Beispiel einer häufig verwendeten verlustlos umkehrbaren Farbtransformation. Der theoretische Dynamikbereich wird um 2 Bits erhöht, die Entropie nimmt jedoch für die meisten Bilder ab.

$$Y = \left\lfloor \frac{R + 2G + B}{4} \right\rfloor$$

$$U = R - G$$

$$V = B - G$$

$$G = Y - \left\lfloor \frac{U + V}{4} \right\rfloor$$

$$R = U + G$$

$$B = V + G$$

Problematisch ist, dass die optimale Farbtransformation bildabhängig ist. Ein interessanter Ansatz

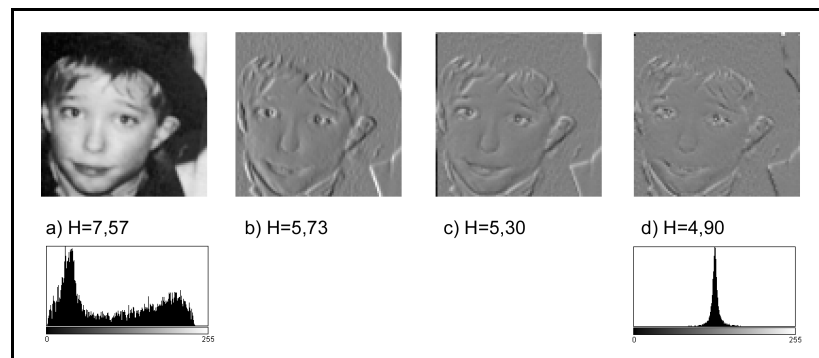


Bild 6 (a) Originalausschnitt und die Fehlerbilder (Offset 128) unterschiedlicher Prädiktoren mit Histogrammen und Angabe der Entropie (b) horizontale Prädiktion, (c) horizontale + vertikale Prädiktion und (d) adaptive Prädiktion.

zur Bestimmung einer verlustlosen, maximal dekorrelierenden Karhunen-Loève Farb-Transformation (KLT) wurde in [4] entwickelt.

Kontextquantisierung. Dekorrelierende Techniken können verwendet werden, um den Wertebereich der zu codierenden Pixelwerte zu reduzieren, allerdings reicht dies noch nicht aus, um sinnvolle Kontexte zu bilden. Daher besteht der nächste Schritt zumeist darin, Kontextvariablen zu quantisieren, d.h. bestimmte Wertebereiche für die Kontextbildung zusammenzufassen, um somit auf eine handhabbare Zahl von Kontexten zu kommen. Ein sehr gutes Beispiel für den Einsatz der Kontextbildung und der Kontextquantisierung ist die sehr leistungsfähige *CALIC-Kompression* [5]. CALIC steht für „*context-based adaptive lossless image compression*“ und wurde 1995 von der ISO bei der Suche nach einem neuen Standard für die verlustlose Bildkompression als bestes Verfahren eingestuft.

CALIC. CALIC verwendet zwei unterschiedliche Kontexttypen. Nach einer adaptiven Prädiktion wird eine Modellierung des Prädiktionsfehlers durchgeführt. Hierzu wird die Prädiktionsfehlerenergie aus den Gradienten der Nachbarpixel und dem benachbarten Prädiktionsfehler abgeschätzt. Dieser Schätzwert wird in vier Bereiche quantisiert. Zusätzlich wird erfasst, ob die umgebenden Pixel größer oder kleiner als der Prädiktionswert sind. Diese beiden Informationen bilden zusammen 576 mögliche Kontexte zur Modellierung der Prädiktionsfehler bei unterschiedlichen Bildtexturen. Aller-

dings wird nicht die Verteilung der Prädiktionsfehler modelliert, sondern „nur“ der Erwartungswert des Prädiktionsfehlers bestimmt. Dieser Wert kann als weitere Verbesserung der Prädiktion verwendet werden, da gelernt wird, in wie weit ein Prädiktor bei bestimmten Kontexten versagt. Bei der anschließenden Codierung des Prädiktionsfehlers werden acht Kontexte verwendet. Die Klassifizierung erfolgt über die zu erwartende Prädiktionsfehlerenergie. CALIC verwendet noch einige weitere Konzepte zur Steigerung der Kompression, deren Darstellung jedoch den Rahmen dieses Beitrags sprengen würde. Der 1997 verabschiedete Standard zur verlustlosen Bildkompression JPEG-LS [6], der auch als LOCO bekannt ist, ging im Wesentlichen aus einer vereinfachten Version von CALIC hervor.

JPEG2000. Der *neue Bildkompressionsstandard JPEG2000* [7] erlaubt ebenfalls eine verlustlose Kompression von Bilddaten. Hierbei wird zunächst eine umkehrbare Integer-Wavelet-Transformation durchgeführt. Die Waveletzerlegung wird in Blöcke unterteilt, die ihrerseits in ihre Bitebenen zerlegt werden, die dann mit einem arithmetischen MQ-Coder entropiecodiert werden. Für die Kontextbildung eines Bits eines Koeffizienten werden die acht umgebenden Positionen verwendet. Insgesamt werden neun Kontexte unterschieden. Dadurch, dass jeder Block unabhängig von den anderen codiert wird, besteht eine gute Möglichkeit zur Adaptation an lokale Schwankungen der Bildstatistik. Durch die Zerlegung in Bitebenen erlaubt JPEG2000 eine progres-

sive Übertragung der Daten, sodass aus einem Bruchteil einer verlustlos codierten JPEG2000-Datei z. B. eine Vorschau generiert werden kann.

Aktuelle Entwicklungen. Trotz jahrelanger Forschungsaktivitäten im Bereich der verlustlosen Bildkompression sind die Fortschritte – zumindest im Vergleich zu den verlustbehafteten Verfahren – eher moderat. Die aktuellen Forschungen zur verlustlosen Bildkompression lassen sich in zwei Gruppen einteilen: Die eine Gruppe versucht ohne Rücksicht auf Rechenzeit und Speicherbedarf zu ergründen, wo die theoretische Grenze der Komprimierbarkeit von Bildern liegt. Hierzu zählen z. B. das TWM-Verfahren von Meyer [8] oder MRP von Matsuda [9], deren Ergebnisse im folgenden Abschnitt mit dargestellt werden. Diese Verfahren benötigen für die Kompression aber bis zu mehreren Stunden. Die zweite Gruppe versucht, möglichst gute Kompressionsergebnisse bei Wahrung einer akzeptablen Ausführungszeit zu erzielen. Eine beschleunigte Weiterentwicklung eines Verfahrens, das versucht, für jeden Pixel einen optimalen Prädiktor zu verwenden [10], stellt das von Meyer vorgeschlagene Glicbawls-Verfahren dar [11].

8 Vergleich von verlustlosen Kompressionsverfahren

Die folgenden Tabelle vergleicht das ZIP-Verfahren als universelles Kompressionsverfahren, die Bildformate GIF und PNG mit speziellen Bildkompressionsverfahren. Zu PNG ist anzumerken, dass PNG unterschiedliche Prädiktionsfilter unterstützt, von denen jeweils dasje-

Tabelle 1 Vergleich unterschiedlicher Verfahren zur verlustlosen Bildkompression (Raten in bpp).

| | ZIP | GIF | PNGcrush | Jpeg-LS | MRP | BMF | CALIC | GLIC | TMW | JP2 |
|-----------------|-----------|-----------|-----------|---------|----------------|--------|-------|--------|--------------|------|
| Near-Lossless | nein | nein | nein | ja | nein | ja | nein | ja | nein | nein |
| Progressivität | nein | ja | ja | nein | nein | nein | nein | nein | nein | ja |
| Geschwindigkeit | sehr hoch | sehr hoch | sehr hoch | hoch | extrem langsam | mittel | hoch | mittel | sehr langsam | hoch |
| Goldhill 8 Bit | 6,56 | 7,69 | 4,68 | 4,48 | 4,23 | 4,27 | 4,39 | 4,28 | 4,27 | 4,66 |
| Airplane 8 Bit | 5,74 | 6,55 | 4,22 | 3,81 | 3,61 | 3,63 | 3,74 | 3,66 | 3,60 | 4,04 |
| Peppers 8 Bit | 7,08 | 8,15 | 4,89 | 4,51 | 4,23 | 4,25 | 4,42 | 4,25 | 4,25 | 4,65 |

nige Ergebnis mit der besten Kompression genommen wurde. Wird die Standardeinstellung von PNG verwendet, so sind die Ergebnisse meist deutlich schlechter. JPEG-LS ist zwar als Standard verabschiedet worden, hat bis jetzt allerdings nur sehr geringe Verbreitung gefunden. CALIC, TMW und Glicbawls (GLIC) sind als Forschungsreferenz mit angegeben. Besonders gute Kompressionsergebnisse bei akzeptabler Codierungszeit erzielt ein Verfahren namens BMF [12], zu dem allerdings keine Literatur verfügbar ist.

Der neue Bildkompressionsstandard JPEG2000 (JP2) besticht durch eine Vielzahl von sehr interessanten Merkmalen, von denen die verlustlose Kompression nur eines ist. Weiterhin ist die breite Unterstützung unterschiedlichster Bildtypen und die Möglichkeit zur interaktiven Steuerung der Darstellung z. B. bei Webapplikationen zu nennen. Die Ergebnisse der verlustlosen Kompression von JPEG2000 sind gut, auch wenn andere Verfahren diese noch leicht übertreffen können.

Literatur

- [1] C. E. Shannon: A mathematical theory of communication. – In: Bell Systems Technical Journal, 1948 27, pp. 379–423, 623–656.
- [2] I. H. Witten, R. M. Neal, J. G. Cleary: Arithmetic coding for data compression. – In: Communications of the ACM (1987), vol. 30, no. 6, pp. 520–540.
- [3] A. Moffat, T. C. Bell, I. H. Witten: Lossless compression for text and images, International Journal of High Speed Electronics and Systems 8 (1), pp. 179–231, 1997.
- [4] S. Van Assche, W. Philips, I. Lemahieu: Lossless compression of pre-press images using linear color decorrelation. Proceedings of the Data compression conference, 1998, pp. 578.
- [5] X. Wu, N. Memon: Context-based, Adaptive, Lossless Image Codec. – In IEEE Trans. on Communications, vol. 45, no. 4, 1997.
- [6] M. J. Weinberger, G. Seroussi: From LOCO-I to the JPEG-LS Standard Computer Systems Laboratory HP Laboratories Palo Alto HPL-1999-3, 1999.
- [7] ISO/IEC 15444-1, Information Technology: / JPEG 2000 Image Coding System/Part 1: Core Coding System, 2000.
- [8] B. Meyer, P. E. Tischer: TMW – a new method for lossless image compression. In Picture Coding Symposium (PCS'97), VDE-Verlag, pp. 533–538, Berlin, Germany, 1997.
- [9] I. Matsuda, H. Mori, S. Itoh: „Lossless Coding of Still Images Using Minimum-Rate Predictors“, Proceedings of 2000 IEEE International Conference on Image Processing (ICIP 2000), Vol. I, pp. 132–135, Sep. 2000.
- [10] X. Wu, K. U. Barthel, W. Zhang: Piecewise 2D Autoregression for Predictive Image Coding. ICIP 1998: pp. 901–904.
- [11] B. Meyer, P. Tischer: Glicbawls – Grey Level Image Compression By Adaptive Weighted Least Squares.
- [12] D. Shkarin: TGA,BMP,RAS,GIF,RAW to BMF file convertor, v.1.1 (C) 1998–1999.
- [13] ITU-T Recommendation T.82|ISO/IEC 11544: 1993, Coded representation of Picture and Audio Information – Progressive Bi-Level Image Compression.



Prof. Dr.-Ing. Kai Uwe Barthel lehrt an der FHTW Berlin am Studiengang Internationale Medieninformatik. Hauptarbeitsgebiete: Medientechnik, Bild- und Videoverarbeitung, insbesondere Kompressionstechniken sowie Multimediaapplikationen.
Adresse: Internationale Medieninformatik, FHTW Berlin, FB4, Treskowallee 8, D-10313 Berlin, Tel.: +49/30/5019-2416 Fax: +49/30/5019-2400.
E-Mail: barthel@fhtw-berlin.de, Web: <http://www.f4.fhtw-berlin.de/~barthel>