

# Bildkompression

Johannes Dambacher

Otto-von-Guericke-Universität Magdeburg  
johannes.dambacher@st.ovgu.de

**Zusammenfassung.** Ohne Bildkompression wäre unser alltäglicher Umgang mit Bildern nicht möglich. Sie erlaubt es mittels Handys und Digitalkameras Bilder zu machen und diese über das Internet zu verbreiten. Die Bilder selbst finden vielfältige Anwendung in Wissenschaft, Industrie, Medizin, Kunst und nicht zuletzt im Privaten. Dieser Text wird kurz auf verschiedene Möglichkeiten der verlustlosen und verlustbehafteten Kompression eingehen, verwendete Standards und Formate beschreiben, sowie deren Vor- und Nachteile darstellen.

## 1 Einleitung

Ohne Bildkompression wäre der digitale Bildaustausch über das Internet, ebenso wie der Bau günstiger Digitalkameras und Handys mit Kamerafunktion teurer und ein vielfaches an Speicher und Datenübertragung notwendig. Dies liegt an der Menge Bits, die nötig ist, um Bildinformation zu codieren. Ziel der Bildkompression ist es, möglichst wenige Bits für die Codierung eines Bildes zu verwenden, ohne dabei relevante Information zu verlieren. Dies ist zum einen möglich, da Bilder redundante Information enthalten, zum anderen, da die menschliche Wahrnehmungsfähigkeit begrenzt ist [12]. Es ist zwischen verlustloser und verlustbehafteter Kompression zu unterscheiden [13]. Bei ersterer besteht nach der Decodierung kein Unterschied zum Original, bei letzterem enthält das codierte Bild weniger Informationen als das Original. Dies stellt bei Bildern und Fotos, die für die menschliche Betrachtung gedacht sind kein Problem dar, solange sich kein Unterschied feststellen lässt. Sollen die Bilder digital weiter bearbeitet werden, kann der Informationsverlust die Ergebnisse verschlechtern. Als Beispiel für den Nutzen der Bildkompression betrachten wir ein einfarbiges Bild ohne Rauschen. Wird dieses pixelweise in einer Matrix  $M \times N$  abgespeichert, enthält es kaum Information, da sich keine Pixel grauwertweise voneinander unterscheiden. Dennoch bedarf es für diese Art der Speicherung als Farbbild  $M \cdot N \cdot$  drei Byte, obwohl es ausreicht, nur das Seitenverhältnis und die Farbe zu speichern, um das Original Bild herstellen zu können.

Enthält das Bild zusätzlich Linien, genügt es diese über Start-, Endpunkt, Farbe und Breite zu definieren, anstatt pixelweise die gesamte Matrix. Auf ähnliche Weise lassen sich Kurven und geometrische Figuren ebenso wie Farbverläufe codieren. Diese Art der Speicherung findet bei sogenannten Vektorgraphiken Verwendung [7]. Im Gegensatz zur Speicherung in Matrixform, bedarf es bei der Speicherung als Vektorgrafik in diesem

Fall weniger Speicherplatz und die Bilder sind skalierbar, jedoch ist die Decodierung aufwändiger. Verwendet werden Vektorgrafiken aus diesem Grund fast ausschließlich für digital erzeugte Bilder, wie beispielsweise Diagramme, Logos oder Zeichensätze.

Folgend wird auf die Kompression und Codierung von Fotos und Bildern eingegangen. Bei diesen ist eine Speicherung als Vektorgrafik ungünstig, da weder gleichmäßige Farbflächen noch Farbverläufe auftreten und selbst einfache geometrische Objekte wie Linien durch Rauschen und Artefakte gestört sind. Diese Störung bewirkt, dass entweder mehrere kurze Linien erzeugt werden, oder der Linienverlauf durch zusätzliche Stützpunkte beschrieben wird. Dies erhöht zum einen den Speicherbedarf, zum anderen dauert die Decodierung länger. Hinzu kommt die Notwendigkeit geometrische Formen erst in Fotos zu finden, bevor dieses anhand der Formen codiert werden kann. Die Suche nach geometrischen Formen ist jedoch rechenintensiv und nicht in akzeptabler Zeit auf Geräten wie Handys und Kameras zu realisieren.

Dennoch ist auch in Fotos Information redundant vorhanden. Betrachtet man ein Bild mit blauem Himmel, tragen die einzelnen Pixel innerhalb des Himmels wenig Information. Die Notwendigkeit, Fotos zu komprimieren, entstand erst mit der Verbreitung leistungsstarker Computer, Farbmonitoren und digitalen Bildsensoren. Da sowohl die Codierung als Vektorgrafik, als auch die Verwendung bekannter Codierungsverfahren, wie die Entropie- oder Lauflängencodierung, ungeeignet sind, mussten neue Verfahren entwickelt werden. Bei der Entropiecodierung liegt es an der großen Anzahl Symbolen und deren statistische Verteilung in Farbbildern, bei der Lauflängencodierung werden die Runs zu kurz, im Extremfall nur ein Pixel lang.

Im Zuge dessen entstanden verschiedene Formate zur Codierung und dem Austausch digitalen Fotos. Verwendet wurden neben frequenzbasierten Verfahren wie der Diskreten Cosinus oder Fouriertransformation, auch Fraktale Kompressionsarten und andere Methoden. 1992 wurde von der Joint Photographic Experts Group (JPEG)-Komitee nach sechs Jahren Arbeit erstmals ein internationaler Standard zur Komprimierung digitaler Bilder vorgestellt [1]. Obwohl darin keine Spezifikation zur Speicherung der codierten Daten gemacht wird, ist der Standard inzwischen meist verbreiteter Standard unter den verlustbehafteten Kompressionsverfahren. Ein neuerer Standard des JPEG-Komitees zur verlustbehafteten Kompression stellt JPEG2000 dar [1]. Ein weiteres weit verbreitetes Format stellt das Portable Network Graphics (PNG) Format dar [2]. Es ist das meist verbreitete verlustfrei komprimierende im Internet verwendete Format.

In Teil 2 werden grundlegende Kompressionsarten, welche auch in der Bildkompression Verwendung finden, beschrieben, während sich Teil 3 mit den speziell in der Bildkompression verwendeten Verfahren beschäftigt. In Teil 4 wird abschließend anhand gängiger Bildformate das Zusammenspiel der vorgestellten Kompressionsarten erläutert und Teil 5 gibt einen kurzen Ausblick über das Thema.

## 2 Entropie-, Lauflängen- und Prädiktionscodierung

Als grundlegende Verfahren zur Bilddatenkompression kommen angepasste Verfahren zu Einsatz, welche auch in anderen Gebieten zur Datenkompression eingesetzt werden. Folgend werden deshalb Entropie-, Lauflängen- und Prädiktionscodierung vorgestellt. Diese finden - meist kombiniert und in unterschiedlicher Weise - bei allen verbreiteten Formaten Verwendung. Unter Entropiecodierung versteht man die Verwendung unterschiedlich langer Codewörter für die zu codierenden Symbole in Abhängigkeit der Symbolhäufigkeit. Den am häufigsten vorkommenden Grauwert wird das möglichst kürzeste Codewort - ein Bit - vergeben, dem seltensten auftretenden Grauwert das längste benötigte [13]. Damit ist diese Art der Codierung dem Morsecode ähnlich, da auch dort oft vorkommende Buchstaben kurze Codewörter haben. Jedoch muss die Codierung im Gegensatz zum Morsecode eindeutig sein, da sonst zusätzliche Zeichen zur Trennung der Codewörter benötigt werden. Durch die Notwendigkeit der Verwendung ganzer Bits und Zahlen begrenzter Länge ist die mathematisch erreichbare Codelänge mit Computern nicht erreichbar, lässt sich jedoch annähern. Eine beweisbar optimale Lösung im Binärraum lässt sich mittels Huffmancodierung erzeugen [10]. Mittels arithmetischer Codierung kann eine bessere Kompressionsrate erreicht werden, da diese im Zahlen- statt Bitraum arbeitet. Nachteil der arithmetischen Codierung ist ihre vergleichsweise aufwendige Implementierung und Langsamkeit gegenüber der Huffmancodierung. Da die Huffmancodierung praktisch kaum schlechtere Ergebnisse erzielt als die arithmetische Codierung, wird zumeist auf die Huffmancodierung zurückgegriffen [13].

Entropiecodierung stellt eine Möglichkeit dar, Daten aller Art unter möglichst geringem Speicheraufwand zu codieren und kann zusätzlich zu anderen Komprimierungen verwendet werden. Einziger Nachteil ist die Notwendigkeit einer Codierungstabelle, welche zu jedem Bild generiert und gespeichert werden muss, um bei der Decodierung die Symbole zu dereferenzieren zu können.

Im Gegensatz zur Entropiecodierung wird die Lauflängencodierung verwendet, um häufig auftretende Symbolfolgen als Ganzes statt als einzelne Symbole zu codieren [14]. Gerade in Bildern treten Flächen auf, in welchen sich die Grauwerte über eine längere Folge nicht oder nur gering voneinander unterscheiden. Eine solche Folge lässt sich speichern, indem der Grauwert und die Länge der Folge vermerkt werden. Ein solches Tupel wird gängigerweise als Run bezeichnet. Bei geeigneter Implementierung reduziert dies bereits ab Runs der Länge drei die zur Codierung benötigte Anzahl Bits.

Auf diese Weise kann ein einfarbiges Bild bereits mit Angabe der Seitengröße und der Farbe des einzig benötigten Runs gespeichert werden, statt die gesamte Bildmatrix zu speichern. Dieser Ansatz funktioniert jedoch nur bei digital erstellten Bildern gut, welche weder Rauschen noch Farbverläufe enthalten. Auf Bildern, die Rauschen enthalten - wie es bei Fotos der Fall ist - werden die Runs im schlechtesten Fall nur je einen Pixel lang, so dass eine Lauflängencodierung den Speicherbedarf sogar erhöht statt zu verringern. Dies kann verhindert werden, indem die Codierung nicht auf

Grauwert-, sondern auf Biteebene durchgeführt wird, da bei einer Grauwertänderung nicht immer alle Bits gleichzeitig kippen. Bei Verwendung des Grey- statt Binärcodes wird dieser Effekt weiter verbessert, da durch den Greycode sicher gestellt ist, dass eine Grauwertänderung nur zum Kippen eines einzigen Bits führt. In diesem Fall bleiben immer sieben von acht Runs erhalten. Eine derartige Verwendung der Lauflängencodierung führt bei Bildern mit einfarbigen Flächen bereits zu guten Ergebnissen, Farbverläufe stellen jedoch weiterhin ein Problem dar, da die Runs trotz Verwendung des Greycodes und Codierung auf Biteebene immer wieder abbrechen. Dies kann umgangen werden, indem die Lauflängencodierung nicht auf den Grauwerten selbst, sondern der Differenz zu einem benachbarten - meist dem rechten - Grauwert geschieht. Dazu wird am Anfang jeder Bildzeile der Grauwert selbst gespeichert, danach nur noch die Differenz zum Vorgänger. Bei gleichmäßig linearen Farbverläufen sind diese Differenzen gleich und ermöglichen längere Runs. Diese Art der Kodierung stellt bereits fast eine Art der Prädiktionscodierung dar. Bei der Prädiktionscodierung wird anhand des bereits dekodierten Bildes, noch während der Dekodierung, eine Vorhersage über den Grauwert des nächsten Pixels getroffen [13]. Bei der Codierung muss nur der Fehler zwischen erwartetem und tatsächlichem Grauwert gespeichert werden. Die meisten Prädiktionscodierer verwenden zur Vorhersage nicht nur die aktuelle, sondern auch die darüber liegende Bildzeilen. Dies ermöglicht auch an Kanten eine Vorhersage der zu erwartenden Grauwerte. Bei guter Abschätzung lassen sich die Fehler mittels Lauflängen- und Entropiecodierung besser komprimieren als die original Grauwerte. Die hier vorgestellten einfachen Codierungsmethoden werden meist kombiniert, um eine besser Kompression zu erreichen.

### **3 Transformationscodierung: Dekorellation mittels Farb- und Frequenzraum**

Die Transformationscodierung versucht mittels Transformation der Bildmatrix, unwichtige Information besser von wichtiger Information abzugrenzen, um anschließend die unwichtige Information zu löschen. Welche Information als unwichtig gilt, kann sich aus dem Kontext ergeben, oder daran liegen, dass das Fehlen der Information bei der Rücktransformation die Werte des Ergebnis fast nicht beeinflusst.

Eine anschauliche Art der Transformationscodierung stellt die Transformation aus dem RGB-Farbraum (Rot, Grün, Blau; in Kameras und Bildschirmen verwendet) in den YCbCr-Farbraum (Helligkeit, Blauanteil, Rotanteil; in Fernsehtechnik und JPEG-Standard verwendet) dar. Während im RGB-Raum Farb- und Helligkeitsinformation in allen Kanälen gemischt vorkommen, sind diese im YCbCr-Raum voneinander getrennt [13]. Die Transformation lässt sich mittels einer Matrixmultiplikation durchführen und ist umkehrbar, wodurch sie sich ohne Einschränkung einsetzen lässt.

Betrachtet man nach der Transformation lediglich den Y-Kanal, erhält man ein schwarzweiß Bild, wie es im Schwarz-Weiß-Fernsehen Verwendung findet. Der Y-Kanal enthält dabei die gesamte Helligkeitsinformati-

on des RGB-Bildes und ist für die Darstellung eines Schwarzweißbildes ausreichend. Dadurch lässt sich das YCbCr-Signals sowohl für Schwarz-Weiß-Fernsehen als auch Farbfernseher verwenden. Bei Farbbildschirmen muss lediglich der Cr- und Cb-Kanal hinzu genommen werden.

Daraus ergibt sich die Möglichkeit der Datenreduktion. Denn gibt man sich mit einem Schwarzweißbild zufrieden, genügt die Speicherung des Y-Kanals, wodurch die Datenmenge auf ein Drittel reduziert wird. Solch drastische Reduktion ist nur selten akzeptabel, weshalb oft die Schwäche der menschlichen Wahrnehmung, Farbe weniger gut als Helligkeit wahrzunehmen, ausgenutzt wird [12]. Anstatt für jedes Pixel die Farbinformation zu speichern, wird in den Cb- und Cr-Kanälen nur jedes zweite, vierte, oder achte Pixel gespeichert. Dies wird als Unterabtastung der Farbinformation bezeichnet. Bei der Decodierung werden die fehlenden Farbwerte aus den vorhandenen interpoliert. Das rekonstruierte Bild lässt sich von Menschen trotz erheblicher Reduzierung der Farbinformation nicht vom Original unterscheiden.

Andere in der Bildkompression verwendete Transformationen betreffen nicht den Farbraum, sondern verwenden andere Basisvektoren für die Bildmatrix. Dies kann zu einer Transformation aus dem Orts- in den Frequenzraum führen.

Da in der Bildkompression allem voran die Diskrete Kosinus Transformation (*DCT*) sowie die Diskrete Wavelet Transformation (*DWT*) Anwendung finden, wird im weiteren nur auf diese eingegangen. Es sei jedoch angemerkt, dass sich die *DCT* aus der Fouriertransformation ableiten lässt [3] und Kompression grundsätzlich auch mit anderen Transformationen möglich ist. *DCT* und *DWT* erfahren jedoch aufgrund besserer Eignung am meisten Verwendung. Bei der *DCT* wird das Bild aus dem Ortsraum in den Frequenzraum transformiert. Dabei werden aus den Ortsbasen der Bildmatrix Frequenzbasen in Cosinusform. Ein Wert in der Matrix bildet nicht länger einen Grauwert eines Pixels ab, sondern den *DCT*-Koeffizienten und damit genau eine Welle, welche über das gesamte Bild wirkt. Die *DCT*-Koeffizienten beschreiben die Amplitude der zugehörigen Welle, während die Richtung und Frequenz aus der Position in der Matrix hervorgehen [14]. Auch der Wertebereich umfasst nun nicht länger nur Ganzzahlen zwischen 0 und 255, sondern alle reellwertigen Zahlen.

Innerhalb des Frequenzraumes lassen sich Filter auf das Bild anwenden. Dabei haben beispielsweise Tiefpassfilter eine glättende, Rausch und Kanten unterdrückende Auswirkung, wodurch das Bild nach der Rücktransformation verwaschen und unscharf ist. Im Gegensatz dazu sind Hochpassfilter Kanten und Rausch erhaltend, es entfallen jedoch Flächen- und Farbverblaufinformationen. Die Komprimierung mittels *DCT* nimmt an, dass hochfrequente Anteile weniger wichtig für das Bild als niederfrequente Anteile sind, da wenige abrupte Grauwertänderungen vorhanden sind [13]. Die hochfrequenten *DCT*-Koeffizienten werden nicht benötigt und deshalb mit dem Wert Null ersetzt. Dies lässt sich auch damit begründen, dass die Koeffizienten hoher Frequenzen gewöhnlich gegen Null gehen und damit fast keinen Einfluss bei der Rücktransformation haben. Da dies dennoch eine Kanten unterdrückende Wirkung zur Folge hat, wird statt einfachem Ersetzen mit Null gewöhnlich zuerst eine Quantisierung der *DCT*-Koeffizienten mittels Quantisierungstabelle durchgeführt.

Durch die Division bleiben auch große hochfrequente Amplituden erhalten, welche bei der Rücktransformierung die Kanteninformation tragen. Während die DCT im JPEG-Standard verwendet wird, findet die diskrete Wavelettransformation DWT [4] [11] im neueren JPEG2000 Verwendung [1]. Grund sind Eigenschaften der DWT, welche sie für die Kompression besser geeignet machen. Während eine Änderung an nur einem DCT-Koeffizienten bereits Auswirkungen auf das gesamte Bild hat, hat eine Änderung an den WCT-Koeffizienten nur an einem begrenzten Ort im Bild Auswirkung. Daher stammt auch der Name Wavelet aus dem Französischen für 'kleine Welle'. Auch lassen sich mittels DWT viele Funktionen kompakter repräsentieren als mittels DCT. Wie auch bei der DCT werden zur weiteren Kompression die Koeffizienten zuerst quantisiert und anschließend codiert. Ein weiterer Vorteil der DWT gegenüber der DCT ist die - zumindest analytisch - bessere Laufzeit in  $O(n)$  statt  $O(n \cdot \log n)$  [6]. Die einfachste Form eines Wavelets ist das bereits 1910 entdeckte Haarwavelet, diese findet in der Haartransformation - eigentlich nur eine Form der Wavelettransformation - Anwendung [9]. Erst später wurden weitere Wavelets und die Wavelettransformation selbst entdeckt [8].

Eine weitere Methode der Bildkomprimierung bietet die Fraktalkompression [5]. Weder fand, noch findet diese jedoch in Bildkompressionsstandards Verwendung. Idee der Fraktalenkompression ist die Selbstähnlichkeit eines Bildes. Vereinfacht gesagt wird dazu das Bild bei der Codierung in wenige Pixel große Blöcke zerlegt. Anschließend wird versucht, die Blöcke mittels Transformation, Rotation und Skalierung eines anderen Blockes darzustellen. Grauwertunterschiede werden mittels Grundrechenarten beschrieben. Da das Finden ähnlicher Blöcke komplex ist, findet die Fraktalkompression derzeit keine breite Verwendung, obwohl es Formate dazu gibt und diese anfänglich von Browsern und Betriebssystemen unterstützt wurden. Eine interessante Anwendungsmöglichkeit bietet sich bei der Archivierung von Bildern, da verglichen mit der DWT gute Kompressionsraten erreicht werden, das Decodieren jedoch schneller geht [13].

## 4 Die Standards und Formate JPEG, JPEG2000 und PNG im Vergleich

Es folgt eine Beschreibung der verwendeten Kompressionsarten und eine Gegenüberstellung der Methoden. Der JPEG-Standard definiert verschiedene Möglichkeiten der Bildkompression, verwendet wird jedoch mit wenigen Unterschieden nur die im Folgenden beschriebene Variante. Da die einzelnen Schritte nicht fest vorgeschrieben sind und keine Vorgaben zur Implementierung gemacht werden, können die Codierungen verschiedener Codierer voneinander abweichen. Zu erst erfolgt eine Umrechnung in den YCbCr-Farbraum. Dies geschieht meist aus dem RGB-Farbraum, es sind jedoch für alle herkömmlichen Farbräume Umrechnungen möglich. Theoretisch ist die Umrechnung verlustfrei, durch Rundung ist dies mittels Computer jedoch nicht möglich. Die Verwendung des Alphakanals zur Darstellung von Transparenzen wird erst im JPEG2000-Standard unterstützt.

Die Umrechnung erfolgt da im YCbCr-Raum Farbe und Helligkeit voneinander dekorreliert sind und so eine Unterabtastung der Farbwerte möglich wird. Vor der Unterabtastung erfolgt eine Tiefpassfilterungen um an Kanten geeigneter Werte zu erhalten. Der Grad der Unterabtastung lässt sich unterschiedlich wählen, ist aber immer verlustbehaftet. Anschließend erfolgt eine Unterteilung in  $8 \times 8$ -Blöcke, welche getrennt voneinander mittels DCT in den Frequenzraum transformiert werden. Die Einteilung in Blöcke erfolgt, da sich die Frequenzen von Ausschnitt zu Ausschnitt unterscheiden. Auch die DCT ist ebenso wie die Farbraumtransformation nur theoretisch verlustfrei.

Verglichen mit der folgenden Quantisierung der DCT-Koeffizienten ist dies zu vernachlässigen. Die Quantisierung erfolgt mittels Quantisierungstabelle, welche die einzelnen DCT-Koeffizienten unterschiedlich wichtet. Grundsätzlich ist die Wahl der Wichtungen nicht vorgeschrieben. Es gibt experimentell erarbeitete Tabellen, welche Eigenschaften der menschlichen Wahrnehmung mit einbeziehen. Nach der Quantisierung werden die Werte gerundet wodurch hohe Frequenzen zu Null werden. Die Koeffizienten werden frequenzaufsteigend gespeichert, was einer Zickzackform auf dem Block entspricht. Da die hohen Frequenzen Null sind, brauchen diese nur bis zur letzten von Null verschiedenen Zahl gespeichert werden. Abschließend wird eine Entropiecodierung - meist mittels Huffmancode - angewendet. Dabei kann auch eine Prädiktionscodierung zum Einsatz kommen, welche die niedrigste Frequenz des nächsten Blocks anhand vorhergegangener vorauszusagen versucht. Die Dekodierung erfolgt auf umgekehrte Weise.

Grundsätzliche unterschied zum JPEG2000-Standard ist die Verwendung der DWT statt der DCT. Ein weiterer Unterschied ist die verlustlose Kompression. Obwohl diese zwar auch mit JPEG möglich ist, wird kaum darauf zurück gegriffen, während dies bei JPEG2000 der Fall ist. Wie auch bei JPEG werden bei JPEG2000 die einzelnen Schritte nicht fest vorgeschrieben und keine Vorschriften bezüglich der Implementation vorgelegt. Auch in diesem Fall können die Ergebnisse verschiedener Codierer voneinander abweichen.

Zuerst erfolgt ebenfalls eine Farbraumtransformation zur Dekorellation. Im verlustbehafteten Falle in den YCbCr-Farbraum, im verlustlosen Fall korrespondiert die Transformation mit der Wavelettransformation. Anstatt einer Unterabtastung im YCbCr-Farbraum wird ein Weglassen der hohen Frequenzen empfohlen. Eine Einteilung in  $8 \times 8$ -Blöcke erfolgt ebenfalls nicht, statt dessen ist bei großen Bildern eine Unterteilung in sogenannte Tiles vorgesehen. Diese erfolgt über Angabe der Koordinaten, zusammen mit der Möglichkeit, Tiles einzeln zu Decodieren ergibt sich die Möglichkeit Region of Interests zu definieren. Die DWT erfolgt auf den Tiles separat und mit unterschiedlichen Wavelets für verlustlose und verlustbehaftete Komprimierung. Eine Quantisierung der Koeffizienten wird nur im Falle einer verlustbehafteten Komprimierung empfohlen. Anschließend erfolgt die Entropiecodierung der Koeffizienten unter Verwendung eines binären arithmetischen Codierers, während Prädiktionscodierung nur rudimentär verwendet wird. Die Dekodierung erfolgt wie bei JPEG auf umgekehrte Weise.

Das Portable Networks Graphics Format *PNG* sieht im Gegensatz zu

vorigen Varianten ausschließlich eine verlustfreie Kompression vor. PNG ist für die Verwendung mit Alphakanal geeignet. Auch in diesem Fall sind einige Codierungsschritte optional. Die Codierung erfolgt ausschließlich im RGB-Raum. Es erfolgt dennoch eine Dekorellation mittels Differenzen zwischen Nachbarpixeln. Es stehen fünf unterschiedliche Varianten zur Verfügung, zwischen welchen zeilenweise unterschieden werden kann. Zugunsten schnellerer Laufzeit wird jedoch meist auf eine Optimierung verzichtet und auf statistische Werte zurück gegriffen. Durch die Nachbarschaftsdifferenzierung entstehen im Betrag kleinere Werte welche anschließend einfacher codiert werden können. Zur abschließenden Codierung wird erst eine verbesserte Lauflängencodierung angewandt - das LZSS-Verfahren - um anschließend mittels Huffmancode eine Entropiecodierung durchzuführen. Prinzipiell sind auch andere Verfahren zur Codierung möglich, derzeit wird jedoch nur diese unterstützt.

## 5 Ausblick

### Literatur

1. Offizielle JPEG Website, <https://jpeg.org/>
2. PNG (Portable Network Graphics) Specification, Version 1.2. (1999)
3. Ahmed, N., Natarajan, T., Rao, K.: Discrete Cosine Transform. IEEE Transaction on Computers 23, 90–93 (1974)
4. Bäni, W.: Wavelets. Oldenburg Verlag, Oldenburg, 2. auflage edn. (2005)
5. Barnsley, M., Hurd, L.: Fractal Image Compression. AK Peters (1993)
6. Brani, V., Mueller, P.: Wavelets for kids (1991), <http://www2.isye.gatech.edu/brani/wp/kidsA.pdf>
7. Bungartz, H.J.: Einführung in die Computergraphik: Grundlagen, geometrische Modellierung, Algorithmen. Vieweg, Braunschweig (2002)
8. Daubechies, I.: Orthonormal bases of compactly supported wavelets. Communications Pure and Applied Mathematics 41, 909–996 (1988)
9. Haar, A.: Zur Theorie der orthogonalen Funktionensysteme. Mathematische Annalen 69, 331–371 (1910)
10. Huffman, D.: A method for the construction of minimum redundancy codes. Proceedings of the IRE 40 (1952)
11. Louis, A.K., Maaß, P., Rieder, A.: Wavelets. Teubner (1998)
12. Mallot, H.: Sehen und die Verarbeitung visueller Information. Vieweg, Wiesbaden (1998)
13. Strutz, T.: Bilddatenkompression. Vieweg+Teubner Verlag, Rostock, 4. auflage edn. (2009)
14. Tömmies, K.D.: Grundlagen der Bildverarbeitung. Pearson Studium, München (2005)