

Proseminar Multimediatdatenbanken und Retrieval

Thema: Bilddatencodierung I, “Joint Photographic Experts Group”

Viktor Wart

In dieser Ausarbeitung behandle ich heutige Standards in der Bilddatencodierung und ihre wesentlichen Verfahren, Algorithmen und Ideen. Den Schwerpunkt bildet der JPEG-Standard (Joint Photographic Experts Group). Das JPEG-Format gehört zu den momentan am weitesten verbreiteten Bilddatenformaten. Er ist Standard auf den meisten Digitalkameras, Handys und vielen anderen Endgeräten und wird von den meisten im Bilddatenbereich angesiedelten Software-Produkten unterstützt.

1 Codierungstheorie

Dieses Kapitel erklärt die grundlegende Terminologie der Codierungstheorie und erklärt die Ansatzpunkte der Datenkompression.

1.1 Grundlagen

Eine Codierung vorliegender Daten bedeutet immer eine Veränderung dieser. Das Ziel einer Codierung ist eine möglichst große Verringerung des Datenaufkommens, man spricht von einer *Kompression*. Da eine Datenkompression stattfindet stellt sich die Frage ob das resultierende Ergebnis genauso aussagekräftig ist wie das bearbeitete Original, daher unterscheidet man bei Kompressionsverfahren zwischen *verlustfreien* und *verlustbehafteten* Verfahren. Aus verlustfrei codierten Daten lassen sich die Ausgangsdaten vollständig rekonstruieren, das genaue Gegenteil ist bei verlustbehafteten Verfahren der Fall.

Wie lassen sich Daten verlustfrei oder verlustbehaftet codieren? Um hier ansetzen zu können, unterscheidet man in der Codierungstheorie zwischen zwei unterschiedlichen Arten von Datenaufkommen: der *Redundanz* und der *Irrelevanz*.

1.2 Redundanz und Irrelevanz

Um den Begriff der Redundanz erklären zu können erfordert es zuvor der Erläuterung der Begriffe *Informationsgehalt* und *Entropie*.

1.2.1 Informationsgehalt

Ein Zeichen, Symbol oder Ereignis in einem Datensatz bezeichnen wir mit s_i und dessen Auftretenswahrscheinlichkeit mit p_i . Den Informationsgehalt eines Symbols definiert man dann folgendermaßen:

$$I(s_i) = \log_2\left(\frac{1}{p_i}\right) [1 \text{ bit}] \quad (1.2.1)$$

Die Einheit des Informationsgehalts I wird in 1 bit angegeben. Diese Definition des Informationsgehalts deckt sich nicht unbedingt mit dem umgangssprachlichen Verständnis von Informationsgehalt, denn sie besagt, dass der Informationsgehalt eines Ereignisses umso kleiner ist, umso häufiger es auftritt. Das heißt in der Codierungstheorie betrachtet man im Rahmen des Begriffs des Informationsgehalts nur die Auftretenswahrscheinlichkeit des Ereignisses, nicht aber die eigentliche Aussage des Ereignisses, wie man es umgangssprachlich definieren würde. Umso „überraschender“ das Auftreten eines Ereignisses, umso größer der codierungstheoretische Informationsgehalt.

1.2.2 Entropie

Der Begriff der Entropie stammt ursprünglich aus der Physik und zwar hauptsächlich aus der Thermodynamik. Ein klar organisiertes System mit einem minimalen Anteil an Zufälligkeit hat eine niedrige Entropie. Dementsprechend lässt sich Entropie als das Maß für „Unordnung“ verstehen. Sie berechnet sich aus der gewichteten Summe des jeweiligen Informationsgehalts.

$$H = \sum_{i=1}^K p_i \cdot I(s_i) [\text{bit/Symbol}] \quad (1.2.2)$$

K beschreibt die Anzahl möglicher Symbole. Somit lässt sich die Einheit 1 bit pro Symbol codierungstheoretisch als durchschnittlich zur Codierung notwendige bit-Zahl pro Symbol interpretieren. Dieser Zusammenhang wird im nächsten Abschnitt aufgegriffen und deutlicher erklärt.

1.2.3 Redundanz

Der Anteil am Datenaufkommen, der für die Repräsentation der Information nicht notwendig ist heißt Redundanz. Codierungen die hier ansetzen und diesen Anteil beseitigen, um eine Datenkompression zu erzielen, machen sich den Entropiebegriff zu Nutzen. Man spricht hier auch von *Entropie-Codierung*. Entropie-Codierung ist eine verlustfreie Technik der Datenkompression. Der wesentliche Ansatzpunkt ist, dass man besonders häufig auftretenden Symbolen kürzere und selten auftretenden Symbolen längere Codewörter zuordnet, als dass man z.B. allen Symbolen

Codewörter gleicher Länge gibt. Der Informationsgehalt eines Symbols kann hierbei als notwendige Anzahl an Bits zur Codierung verstanden werden. Folgendes Beispiel erklärt die Datenkompression in der Entropie-Codierung:

	$p(s_i)$	$I(s_i)$	Code	Code _{komprimiert}
a	0,5	1	00	0
b	0,25	2	01	10
c	0,125	3	10	110
d	0,125	3	11	111

Im obigen Beispiel gibt es vier mögliche Symbole, die in einem Datensatz auftreten können. Jedes Symbol tritt erfahrungsgemäß mit einer unterschiedlicher Wahrscheinlichkeit auf. Die Spalte „Code“ zeigt die gewöhnliche Codierung, für 2 Symbole benötigt man eben 2 Bit zur Darstellung. Die letzte Spalte zeigt den Code die sich der Entropie-Eigenschaft bedient und die Auftrittswahrscheinlichkeiten berücksichtigt. Die Entropie des komprimierten Codes beträgt gemäß Formel 1.2.2 1,75 Bit/Symbol, das heißt man benötigt zur Codierung des Datensatzes 1,75 Bit pro Symbol während es ursprünglich 2 Bit pro Symbol waren. Am obigen Beispiel erkennt man den verlustfreien Charakter der Entropie-Codierung, aus dem komprimierten Code lässt sich stets das Original rekonstruieren, trotz Datenkompression liegt kein Informationsverlust vor.

Aus dem obigen Beispiel lässt sich auch erkennen, dass sich in Ausnahmefällen bei besonders untypischem Datenaufkommen eventuell keine Kompression erzielen lässt. Allerdings ist dies umso unwahrscheinlicher, umso größer der Datensatz ist. Ein auf dieser Grundlage arbeitendes Verfahren ist beispielsweise der *Huffman-Algorithmus*.

1.2.4 Irrelevanz

Der Anteil am Datenaufkommen, der beim Empfänger der Daten nicht wahrgenommen werden kann oder nicht von Interesse ist heißt Irrelevanz. Im Bereich der Bildverarbeitung spricht man auch von *psycho-visueller Redundanz*. Wird beispielsweise ein Bild im RGB-Format übertragen, das heißt jeder Bildpunkt wird durch drei Farbkomponenten zu je 8-Bit repräsentiert, so beträgt der Aufwand 24-Bit pro Bildpunkt. Wenn aber nun der Empfänger das Bild nur mit 256 Farben pro Bildpunkt darstellen kann, wären bereits 8-Bit pro Bildpunkt ausreichend gewesen. Die übrigen Bit wären in dem Fall irrelevante Informationen.

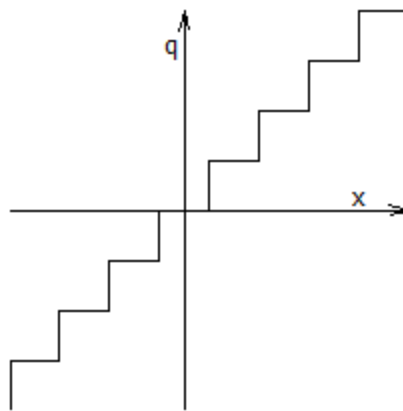
Die Definition der Irrelevanz richtet sich immer nach den Wahrnehmungseigenschaften des Empfängers. Kompressionsverfahren die sich der Irrelevanz bedienen sind verlustbehaftete Verfahren, wenn im obigen Beispiel der Empfänger eine bessere Darstellungsmöglichkeit bekommen würde, z.B. einen neuen Monitor, so hätte er mit den zuvor komprimierten Daten immer noch das gleiche Bild

und könnte auch nicht die im Original noch vorhanden gewesenen Informationen wiederherstellen.

Verfahren die die Irrelevanz geschickt ausnutzen, können eine deutlich bessere Kompression als bei verlustfreien Verfahren erzielen.

1.3 Quantisierung

Ist ein verlustbehaftetes Kompressionsverfahren, dabei wird einzelnen Werten x ein *quantisierter* Wert q zugewiesen.



Durch die Quantisierung lässt sich der Wertebereich verkleinern. Eine solche Art von Quantisierung bezeichnet man auch als *skalare Quantisierung*. Es gibt unterschiedliche Quantisierer-Typen definiert. Die oben gezeigte Quantisierung ist eine *gleichmäßige Quantisierung*, jedes Intervall hat die gleiche Breite. Im Gegensatz dazu wird bei der *ungleichmäßigen Quantisierung* eine Anpassung der Intervallbreite gemäß der Wahrscheinlichkeit der auftretenden Werte. So werden für Bereiche mit hoher Wahrscheinlichkeit kleinere Intervalle gewählt, sodass sich hier eine feinere Quantisierung erzielt wird. Dementsprechend werden die Bereiche kleinerer Wahrscheinlichkeit in größere Intervalle aufgeteilt.

Ferner spricht man auch von einer *wahrnehmungsoptimierten Quantisierung*, wenn die Intervallgröße zusätzlich Erfahrungen und Tests der menschlichen Wahrnehmung berücksichtigt. In der Bilddatenkompression kann man sich beispielsweise vorstellen, dass das menschliche Auge bestimmte Farbnuancen nicht unterscheiden kann und man so entsprechenden Farbbereich einen festen quantisierten Wert zuordnet.

Neben der skalaren Quantisierung gibt es noch die *Vektorquantisierung*, auf die ich allerdings nicht näher eingehen werde, da sie heute von den transformationsbasierten Kompressionsverfahren, welche weit aus leistungsfähiger sind, verdrängt wurde und in der Bilddatenkompression keine Bedeutung mehr hat.

1.4. Lauflängen-Codierung (RLC)

Bei der Lauflängen-Codierung werden Läufe gleicher Werte ausgenutzt. Alle Werte werden mit ihrem Wert und dem nachfolgendem, ununterbrochenem Vorkommen codiert. Tritt der Wert 57 in einer Datei 32 mal in Folge auf, so wird bloß 57 32 festgehalten. Vor allem nach einer Quantisierung erzielt man somit gute Komprimierungen.

2 JPEG Standard

Dieses Kapitel befasst sich mit dem JPEG Standard zur Einzelbildkompression und seiner Entstehung. Dabei bedient sich der Standard der in den vorangegangenen Kapiteln beschriebenen grundlegenden Methoden der Datenkompression.

2.1 Historie

Anfang der achtziger Jahre bildete die ISO (International Organization for Standardization) die *Photographic Experts Group*, mit dem wesentlichen Ziel ein stufenweise vorgehendes Bilddatenkompressionsverfahren für die Anwendung auf ISDN-Kanälen zu entwickeln. Zu der Zeit war auf dem gleichen Gebiet auch die CCITT (International Telegraph and Telephone Consultative Committee) tätig und um die Entstehung unterschiedlicher, konkurrierender Standards zu vermeiden, schlossen sich beide Gruppen 1986 zur heutigen *Joint Photographic Experts Group* zusammen. Aus dieser Gruppe ging 1992 ein Standard hervor, der eine Familie von Verfahren definierte. Dazu gehörten sowohl verlustbehaftete Codierung auf Basis eines Transformationscoders als auch eine verlustfreie Kompression mit Prädiktionstechnik.

2.2 Codierung

Bei der JPEG-Codierung wird das Originalbild zunächst gemäß *Diskreter Cosinus-Transformation* komprimiert und anschließend Quantisiert. Das Resultat dieser Verfahren wird anschließend mittels der in 1.2.2 und 1.2.3 beschriebenen Entropie-Codierung komprimiert. Die Decodierung erfolgt in genau umgekehrter Reihenfolge.

2.2.1 Diskrete Cosinus-Transformation basierte Kompression

Die Diskrete Cosinus-Transformation ist ein verlustbehaftetes Kompressionsverfahren und bildet sozusagen das Herzstück der Komprimierung mit JPEG. Daher fasse ich zunächst die Idee und den Vorgang umgangssprachlich zusammen und gehe dann auf die Mathematik ein.

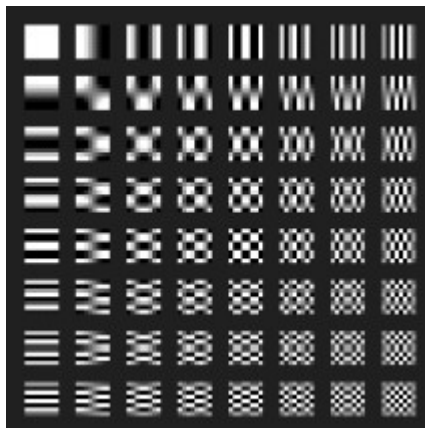
2.2.1.1 DCT umgangssprachlich

Ziel einer Transformation im Allgemeinen ist es die wichtigen Informationen auf wenige und bekannte Elemente zu konzentrieren. In darauf folgenden Schritten können dann beispielsweise durch Quantisierung die unwichtigen Elemente entfernt werden. In unserem Fall ist das Ziel hochfrequente Bildanteile raus zufiltern, was das bedeutet wird im weiteren Verlauf deutlicher.

Die Idee der DCT ist es Daten als eine Matrix zu betrachten und diese als eine Linearkombination aus standardisierten Cosinus-Funktionen darzustellen.

Wie lässt sich ein Bild als eine Kombination aus Cosinus-Funktionen darstellen? Hierfür teilt man das Bild in Blöcke von 8*8 Pixeln auf. Ein solcher Block stellt nur einen sehr sehr kleinen Ausschnittes des Bildes dar, in der Regel ist dort nur eine Kontur oder leichte Farbänderung zu sehen. Ein solch kleiner Ausschnitt lässt sich gut als eine Kombination aus Cosinus-Funktionen darstellen.

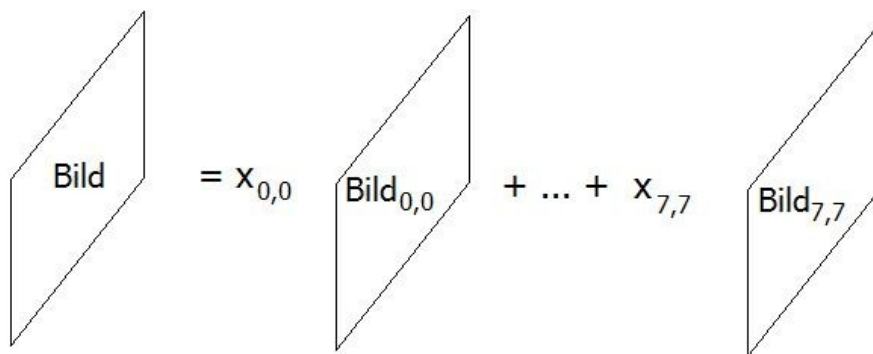
Wie wird nun eine solche Codierung durchgeführt? Es gibt 64 Basisbilder die man in der folgenden Darstellung sieht.



Jedes Bild stellt eine Cosinus-Funktion dar, dunkle Bereiche stehen für besonders hohe Werte und helle Bereiche für besonders niedrige. In der ersten Zeile sehen wir eine horizontale, eindimensionale Cosinus-Funktion, das erste Bild stellt eine Schwingung von niedrig nach hoch dar (weiß nach schwarz). Alle zwei Felder weiter wird die Frequenz verdoppelt. In der ersten Spalte finden wir das gleiche mit einer vertikalen, eindimensionalen Funktion. Das aller erste Feld ist der so genannte "Grauanteil"(oder auch "Grundfarbe") er gibt die ungefähre Farbtiefe des betreffenden Blockes an. Alle anderen Bilder sind Kombinationen der jeweiligen horizontalen und vertikalen Funktionen, d.h. Bild(4/5) ist eine Kombination aus dem vierten Bild der ersten Zeile und dem fünften Bild der ersten Spalte usw.

Ein 8*8-Block wird nun als Kombination der Basisbilder dargestellt. Ein Bildblock in dem beispielsweise ein Übergang von hell nach dunkel zu sehen ist, ließe sich sehr leicht als eine Kombination der ersten Zeile mit ein paar wenigen andern Basisbilder

darstellen. Umso hochfrequenter ein Bild selbst in solch kleinen Ausschnitten ist, umso mehr Basisbilder sind zur Darstellung zu kombinieren. Die letztendliche Codierung ist eine ebenfalls 8*8 große Matrix, deren Werte den jeweiligen Anteil des Basisbildes darstellen, d.h. steht in der Matrix an Stelle (4/5) eine 7, dann ist das Originalbild eine Linearkombination in der das Basisbild(4/5) mit dem Faktor 7 eingeht. Folgende Grafik macht dies nochmals deutlich:



Im JPEG-Format stellt dem zufolge eine 8*8 Matrix einen 8*8 Pixel-Bereich des Originals dar. Nun stellt sich die Frage wo liegt hier die eigentlich Einsparung? Schließlich werden nur Farbwerte von Pixeln in Werte einer Linearkombination umgewandelt, somit entstehen aus 8*8 Zahlen bloß wieder 8*8 Zahlen mit einer anderen Bedeutung. Die eigentliche Komprimierung findet in den Folgenden Schritten statt. Nach der DCT folgt eine Quantisierung, bei der man nun die um strukturierten Daten nutzen kann. Man quantisiert die hochfrequenten Anteile möglichst grob (z.B. mit 99), sodass aus dem Bereich meist nur drei Werte resultieren. Diese lassen sich dann sehr gut mit Hilfe der zu Beginn vorgestellten Entropie Verfahren komprimieren.

2.2.1.2 DCT mathematisch

Sei $F(x, y)$ ein 8*8 Pixel großes Bild, mit den entsprechenden Einträgen zu den Farbwerten an den jeweiligen Positionen. Die zweidimensionale DCT ist durch folgende Formel definiert:

$$f(k, n) = \frac{C(k) \cdot C(n)}{2} \sum_{x=0}^7 \sum_{y=0}^7 F(x, y) \cos\left(\frac{\pi(2x+1)k}{16}\right) \cos\left(\frac{\pi(2y+1)n}{16}\right)$$

Um die Transformation rückgängig zu machen ist folgende inverse DCT (auch IDCT) Formel definiert:

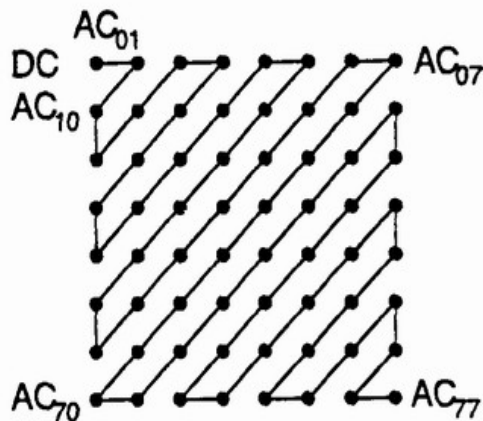
$$F(x, y) = \sum_{k=0}^7 \sum_{n=0}^7 \frac{C(k) \cdot C(n)}{2} f(k, n) \cos\left(\frac{\pi(2x+1)k}{16}\right) \cos\left(\frac{\pi(2y+1)n}{16}\right)$$

Für beide Formeln gilt $k, n \in \{0, \dots, 7\}$ und $c(z) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{für } z=0 \\ 1, & \text{für } z>0 \end{cases}$.

Bei der oben beschriebenen Transformation fasst man das aus 8×8 Bildpunkten bestehende Ausgangsbild $F(x, y)$ als Matrix in $M_8(\mathbb{R})$ auf und stellt es bezüglich einer anderen Basis $B_{k,n}$ von $M_8(\mathbb{R})$ dar. Die 64 Elemente dieser neuen Basis $B_{k,n}$ sind die vorher beschriebenen *Basisbilder*.

2.2.1.3 Codierungs-Struktur eines Bildblocks

Man betrachtet die DCT-Koeffizienten in einer Zickzack-Reihenfolge, somit wird eine Ordnung eingeführt, die ungefähr der schlechter werdenden Sichtbarkeit und steigenden Frequenz der zugehörigen Basisbilder entspricht. Bei der Codierung eines Bildblocks kombiniert man erst die niedrigfrequenten Bilder zu dem Originalbild das Bild das die Kombination der beiden Basisbilder mit der jeweils höchsten Frequenz (AC7,7) wird als letztes betrachtet. Man erinnere sich an dieser Stelle wieder an das Ziel der DCT, die hochfrequenten Anteile zu eliminieren.



2.2.2 Quantisierung

Die nun vorliegenden DCT-Koeffizienten werden mit ungleichmäßigen Intervallen quantisiert. Dabei wird jeder Koeffizient $X[k,l]$ durch einen ihm zugeordneten Quantisierungswert $Q_{k,l}$ dividiert. Dabei bietet der JPEG-Standard bereits eine Vorschlag für die Quantisierungswerte, welche bereits unter der Berücksichtigung der menschlichen Wahrnehmung ermittelt wurden. Dadurch erhält man für jeden Koeffizienten einen Quantisierungswert $q[k,l]$:

$$q[k, l] = \left[\frac{X[k, l]}{Q_{k, l}} + 0,5 \cdot \text{sgn}(X[k, l]) \right]$$

Zum Decodieren multipliziert man die Codierung mit den Quantisierungswerten:

$$X'[k, l] = q[k, l] \cdot Q_{k, l}$$

Da die Quantisierung ein verlustbehaftetes Verfahren ist, handelt es sich bei X' nicht um die gleiche Koeffizientenmatrix wie X . Folgendes Bild zeigt ein Beispiel einer DCT-Koeffizientenmatrix vor und nach der Quantisierung:

Nach DCT:

	0	1	2	3	4	5	6	7
0	8	17	18	19	21	23	25	27
1	17	18	19	21	23	25	27	28
2	20	21	22	23	24	26	28	30
3	21	22	23	24	26	28	30	32
4	22	23	24	26	28	30	32	35
5	23	24	26	28	30	32	35	38
6	25	26	28	30	32	35	38	41
7	27	28	30	32	35	38	41	45

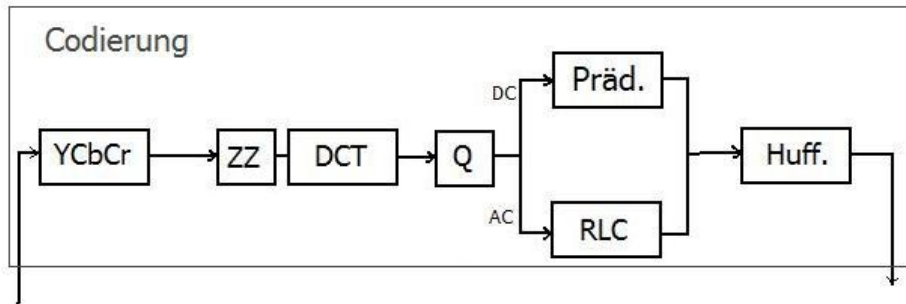
Nach DCT und nach Quantisierung:

	0	1	2	3	4	5	6	7
0	5	0	-1	0	0	0	0	0
1	3	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0
5	2	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	2	0	0	0	0	0	0	0

Verfahren der Entropie Codierung können nur sehr gute Komprimierungen erzielen, da sehr viele 0 Werte und nur wenige verschiedene Werte auftreten.

2.2.3 JPEG Codierung im Überblick

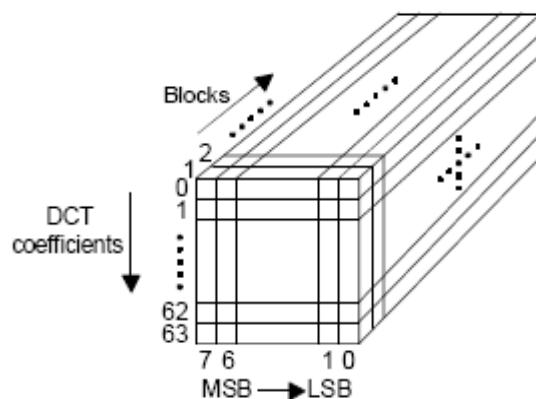
Folgende Abbildung zeigt nochmal die wesentlichen Schritte der JPEG-Codierung. Die Decodierung erfolgt in genau umgekehrter Reihenfolge.



Zunächst erfolgt eine Umwandlung in den YCbCr Farbraum, bei dem bereits die Eigenschaften des menschlichen Auges berücksichtigt werden. Im Anschluss erfolgt die Umstrukturierung der Daten in 8*8 Blöcke und Zickzack Anordnung. Auf die nun vorliegende Datenstruktur wird der DCT angewendet und anschließend quantisiert. Der DC-Koeffizient (Grauanteil) eignet sich besonders um mit einem Prädiktions Verfahren komprimiert zu werden, während die anderen Koeffizienten per RLC komprimiert werden. Im Anschluss darauf wird nochmal der Huffman-Algorithmus angewandt.

2.3 Verarbeitungsmethoden

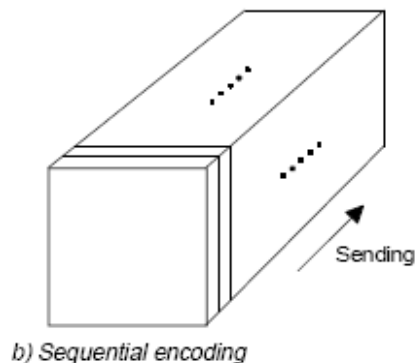
Der JPEG-Standard unterscheidet drei wesentliche Verarbeitungsmethoden, die *sequentielle Verarbeitung*, die *progressive Verarbeitung* und die *hierarchische Verarbeitung*. Die Struktur der zu codierenden Bildblöcke gleicht der nachstehenden Grafik:



Man betrachte ein oben dargestelltes 64×8 großes Rechteck. Eine Zeile steht für die 8-Bit große Zahl eines Eintrags in der Matrix des Bildblocks. Jeder Bildblock hat 64 Einträge die in den 64 Zeilen repräsentiert werden. In der erste Zeile ist die Grundfarbe eingetragen, ihr folgen gemäß der Zickzack Reihenfolge die anderen Einträge nach Frequenzen aufsteigend sortiert, somit bildet AC7,7 die letzte und AC0,1 die zweite Zeile. Die verschiedenen Verarbeitungsmethoden werden in den folgenden Abschnitten beschrieben.

2.3.1 Sequentielle Verarbeitung

Bei der sequentiellen Verarbeitung wird Block für Block von links oben nach rechts unten jede Komponente nur einmal gelesen und direkt codiert. Da die Blöcke jeweils unabhängig voneinander betrachtet werden, wird während der Codierung nur wenig Speicherplatz benötigt und der Codierungsprozess erfolgt ziemlich schnell. Bei der Decodierung wird dementsprechend das Bild Block für Block aufgebaut. Nachstehende Grafik illustriert dieses Vorgehen:

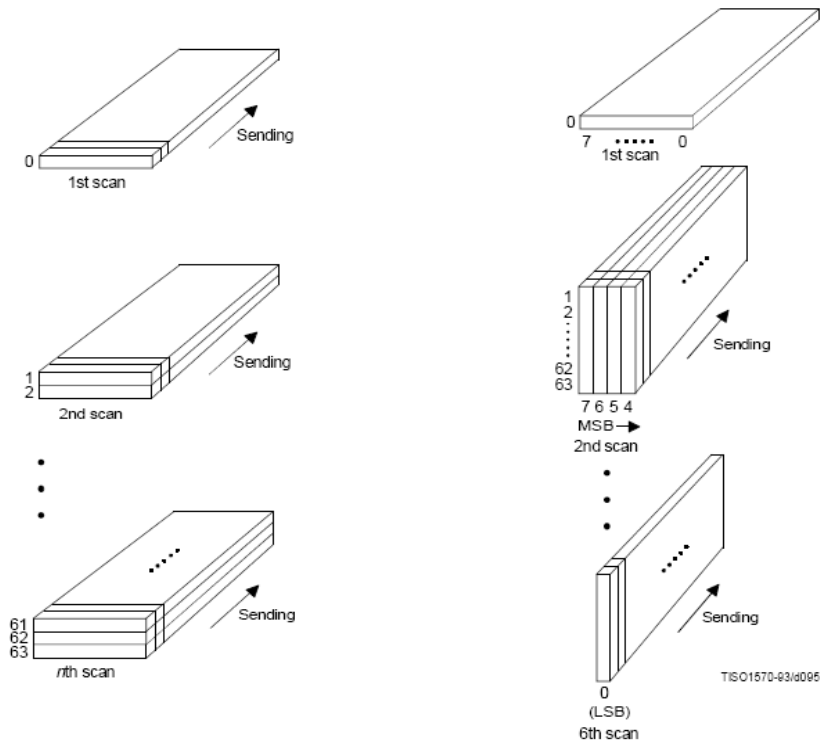


Neben der bereits beschriebenen sequentiellen Verarbeitung gibt es noch eine *verlustfreie sequentielle Verarbeitung*, bei dieser Methode wird die DCT-basierte Kompression weggelassen.

2.3.2 Progressive Verarbeitung

Bei der progressiven Codierung werden die Bildinformationen Schritt für Schritt codiert, zu Beginn wird bloß die grobe Auflösung des Bildes codiert. Die darauf folgenden Durchläufe codieren die feineren Informationen des Bildes nach und nach. Es werden zwei Arten der progressiven Codierung unterschieden:

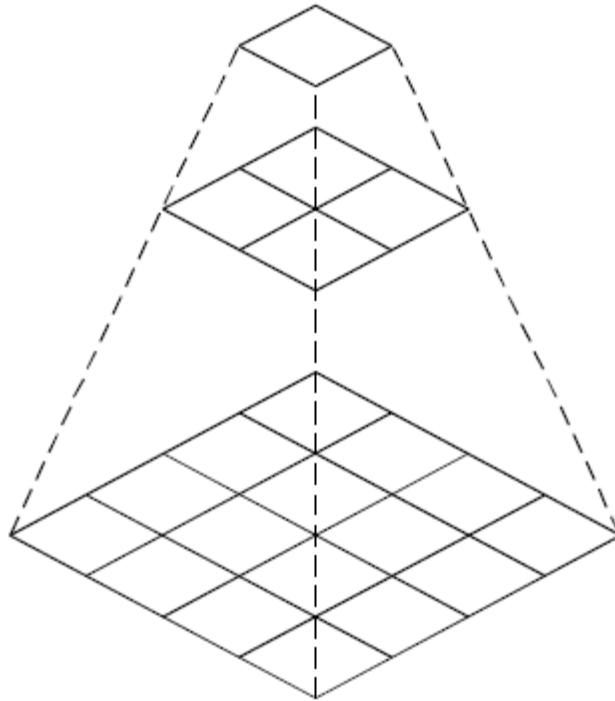
Spektrale Selektion. Bei der spektralen Selektion werden die tieffrequenten vor den hochfrequenten Koeffizienten übertragen. Dieser Prozess wird in der linken Hälfte der unteren Grafik veranschaulicht.



Schrittweise Approximation. Die schrittweise Approximation (*successive approximation*) überträgt erst die DC-Koeffizienten und anschließend von allen Koeffizienten die obersten Bits. In den darauf folgenden Schritten werden die weiteren Bitebenen übertragen. Dieser Prozess ist in der rechten Hälfte der oberen Grafik dargestellt.

2.3.3 Hierarchische Verarbeitung

Die hierarchische Verarbeitung ist eine Art der progressiven Codierung, es wird zunächst ein Bild in niedrigster Auflösung übertragen, aus dem dann die höheren Auflösungen erzeugt werden. Diese „Auflösungspyramide“ wird in der nachstehenden Grafik illustriert.



Beim Codiervorgang wird das Bild in maximaler Auflösung mit jeweils halber Abtastrate mehrfach unterabgetastet. Bei der Decodierung wird das vorliegende Bild, also die niedrigste Auflösung, überabgetastet, bilinear interpoliert und als Prädiktor für die nachfolgenden Bildebenen verwendet.

2.4 Prozessausführungen

Der JPEG-Standard definiert verschiedene Codierungsprozesse.

Baseline-Prozess. Der Baseline-Prozess ist das Kernstück der JPEG-Codierung. Jede Software, die gemäß des JPEG-Standards codieren und decodieren will, muss mindestens die Operationen des Baseline-Prozesses implementieren. Dies erfordert die Prinzipien der DCT-Kompression, die Huffman-Codierung mit maximal zwei Codetabellen und die Verarbeitung von 8-Bit pro Abtastwert, bei sequentieller Codierung.

Erweiterter DCT-basierter Prozess. Im erweiterten DCT-basierten Prozess können über den Baseline-Prozess hinaus auch 12 Bit pro Bildpunkt codiert werden und neben der sequentiellen steht auch die progressive Codierung zur Verfügung. In diesem Prozess kann man darüber hinaus die Huffman-Codierung durch arithmetische Codierung ersetzen, auf die ich aber nicht weiter eingegangen bin.

Verlustfreier Prozess. Des Weiteren gibt es den verlustfreien Prozess in dem ebenfalls nur sequentiell erfolgt mit einer maximalen Anzahl von vier Codetabellen. Es sind 2 Bit bis 16 Bit pro Pixel möglich und um verlustfrei codieren zu können ersetzt man die DCT-Kompression durch eine prädiktive Verarbeitung.

Hierarchischer Prozess. Im hierarchischen Prozess geht man wie in 2.3.3 beschrieben vor, dabei kann wahlweise eine DCT-basierte Codierung oder eine verlustfreie Codierung verwendet werden.

3 Zusammenfassung und Bewertung

JPEG ist ein Kompressions Standard der dem Anwender die Wahl zwischen verschiedenen Qualitätsstufen und Codierungs Prozessen lässt. Somit ist der Standard für verschiedenste Anforderungen zu gebrauchen und sehr anpassungsfähig.

3.1 Stärken und Schwächen

Schwächen zeigen sich bei der JPEG-Codierung immer dort, wo digitale Strichzeichnungen, Bilder mit viel Text wie beispielsweise Zeitungsdruck oder generell Bilder mit vielen hochfrequenten Anteilen zu codieren sind. Hierbei treten viele verschwommene bzw. unscharfe Bereiche auf, da bei der Rückquantisierung große Schwankungen im Bereich der hochfrequenten Anteile verloren gehen.

Vor allem im Bereich der verlustfreien Komprimierung gibt es deutlich bessere Verfahren. Eine mir bekannte Untersuchung hat die Komprimierung eines exemplarischen Beispielbildes mit einer Vielzahl von Komprimierungsverfahren dokumentiert. JPEG gelang eine verlustfreie Komprimierung um 3,5% während das Packprogramm "Stuffit 14" eine Einsparung von 24,21% erzielen konnte.

Stärken sind wiederum "natürlich"-gerasterte Bilder wie Fotografien und computergenerierte Bilder. Hier lassen sich bei geringem Qualitätsverlust gute Ergebnisse erzielen, im Vergleich zu einem 24 Bit RGB Bild ist sogar eine 35fache Komprimierung möglich. Eine weitere Stärke ist die progressive Verarbeitung im Bereich des Internets. Die Vorschau des zu Übertragen Bildes eröffnet dem Betrachter die Möglichkeit die Übertragung vorzeitig zu beenden falls das Bild

uninteressant sein sollte, was vor allem bei besonders hochauflösenden Bildern eine große Zeiteinsparung bedeutet. Allerdings lässt sich beobachten, dass diese Funktion kaum genutzt wird.

Abschließend lässt sich sagen, dass sich vor allem aufgrund der verschiedenen Optionen und der guten Ergebnisse bei Fotos etc. JPEG so weit verbreitet hat und wahrscheinlich auch in naher Zukunft noch Standard bleiben wird.

Quellen:

Strutz, Tilo: Bilddatenkompression. Grundlagen, Codierung, Wavelets, JPEG, MPEG, H.264. 4.Auflage. Wiesbaden: Viewew + Teubner, 2009

Matzer, Michael/Lohse Hartwig: Dateiformate. ODF, DOCX, PSD, SMIL, WAV & Co. Einsatz und Konvertierung. Siegen: Software & Support Verlag GmbH, 2007

Milde, Torsten: Videokompressionsverfahren. Im Vergleich. 1. Auflage. Heidelberg: Verlag für digitale Technologien GmbH, 1995

„JPEG – Wikipedia“. http://de.wikipedia.org/wiki/JPEG#Progressives_JPEG : 04.01.2010

„Lossless jpg/jpeg compression test“. <http://www.maximumcompression.com/data/jpg.php> : 04.01.2010

„Das PCX-Format“. <http://www.weblearn.hs-bremen.de/risse/CNAPS/referate/pcx/pcx.htm> : 04.01.2010

„Flask_DCT_bsp.png“. http://freenet-homepage.de/videostation/Flaskmpeg/DCT/Flask_DCT_bsp.png : 04.01.2010

Abbildungen:

Seite 6 Abbildung 1:

Thomas Moser: Image Similarity
http://www.tmoser.ch/typo3/uploads/pics/dct_2d_block_basis.png

Seite 7 Abbildung 1:

Milde, Torsten: Videokompressionsverfahren. Im Vergleich. 1. Auflage. Heidelberg: Verlag für digitale Technologien GmbH, 1995
Seite 20, Abbildung 2.2

Seite 8 Abbildung, sowie Seite 10 Abbildung 2, Seite 11 Abbildung 1, Seite 12 Abbildung 1 und Seite 13 Abbildung 1:
CCIT, T.81, Terminal equipment and protocols for telematic services. 1992

Seite 9 Abbildung 1:

Offizielles deutsches flaskmpeg & dvdtoogm board

http://freenet-homepage.de/videostation/Flaskmpeg/DCT/Flask_DCT_bsp.png