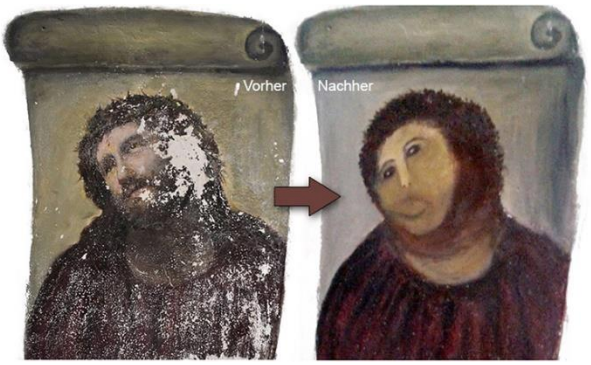
C_BILDER_KOMPRIMIEREN

In einem früheren Kapitel haben wir die verlustlose Komprimierung behandelt. Nun folgt die bei Bild und Ton unverzichtbare verlustbehaftete Variante. Diese bietet zwar wesentlich mehr Potential zur Datenreduktion, hat aber auch den Nachteil, dass bei der Komprimierung und Dekomprimierung Daten verloren gehen und das Resultat nicht mehr ganz dem Original entspricht.



Nicht ganz so wie es vorher war, aber so ähnlich:

In Spanien hat eine Amateur-Restauratorin versucht, ein Jesus-Fresko aufzufrischen und die alte Kirchenmalerei bis zur Unkenntlichkeit überpinselt.

Man versucht mit geeigneten Verfahren in einer Datei den Informationsgehalt derart zu reduzieren, dass das subjektive Empfinden des Datei-Konsumenten die Datenreduktion möglichst nicht wahrnimmt. Dabei gibt es verschiedene Ansätze und oft auch Kombinationen davon.

Die naheliegenden Techniken zur Datenreduzierung:

Die mathematisch weniger aufwändigen Verfahren reduzieren z.B. die Bildauflösung, die Farbtiefe oder die Anzahl der gezeigten Bilder:

- **Bildgrösse reduzieren**: Reduziert man bei einem Original-Bild von 1000 Pixel x 1000 Pixel Grösse sowohl Länge als auch Breite um die Hälfte, erhält man eine viermal kleinere Datei.
- **Farbauflösung reduzieren**: Reduziert man zum Beispiel bei einem TrueColor-Bild die Farbauflösung von ursprünglich pro Grundfarben (Rot,Grün,Blau) 8 Bit bzw. 256 Abstufungen auf 4 Bit bzw. 16 Abstufungen), ist man zwar mit nur noch 4'096 unterschiedlichen Farbe weit entfernt von TrueColor mit 16'777'216 unterschiedlichen Farbe, hat die Dateigrösse damit aber halbiert.
- **Bildwiederholrate reduzieren**: Beim Kino wird seit den Anfängen mit 24 Bilder pro Sekunde gerechnet. Aber schon ab 12 Bilder pro Sekunde würde eine Bewegung einigermassen flüssig zu erkenne sein. Ersparnis: 50%
- **Samplingrate reduzieren**: Wird ein analoges Signal wie z.B. das elektrische Signal eines Mikrophons digitalisiert, läuft das wie folgt ab: In regelmässigen Abständen wird die Amplitude des elektrischen Signals gemessen und in seinen digitale Wert umgewandelt. Bei Audioaufnahmen sind übliche Werte 44.1kHz. Gemäss Abtasttheorem kann damit ein analoges Audiosignal bis 22kHz, was der theoretischen oberen Grenzfrequenz des menschlichen Gehörs entspricht, aufgelöst werden. Reduziert man nun die Samplingrate auf z.B. 8kHz, scherbelts zwar gehörig aus dem Lautsprecher, einem Gespräch kann man aber immer noch folgen, wie die historische Telefongrenzfrequenz von 4kHz beweist. Theoretische Ersparnis bei einer Samplingrate-Reduktion von 44.1kHz auf 8kHz: Datei wir ca. 5x kleiner. Man könne nun auch noch an der Auflösung der Amplitude herumschrauben und anstatt z.B. 16 Bit nur 8 Bit auflösen, was die Dateigrösse nochmals halbieren würde.

- **Eine Farbtabelle benutzen**: Anstatt jedes einzelne Pixel mit den drei Grundfarben RGB in 3x8Bit zu beschreiben, kann man die Farbinformation eines Pixels auch in einer Farbtabelle referenzieren. Das lohnt sich allerdings nur, wenn das reduzierte Farbangebot keine Rolle spielt. Mit Dithering kann man zwar etwas tricksen, indem man benachbarte Pixel mit unterschiedlichen Farbinformationen versieht, die dann bei der Betrachtung aus einer gewissen Distanz zu einer imaginären Drittfarbe verschmelzen.
  Bsp.: GIF mit 8 Bit Farbauflösung (ergibt 256 Farben).

Bereits etwas trickreicher ist das folgende Verfahren, dass sich Subsampling oder Farbunterabtastung nennt.

Subsampling (Farbunterabtastung)

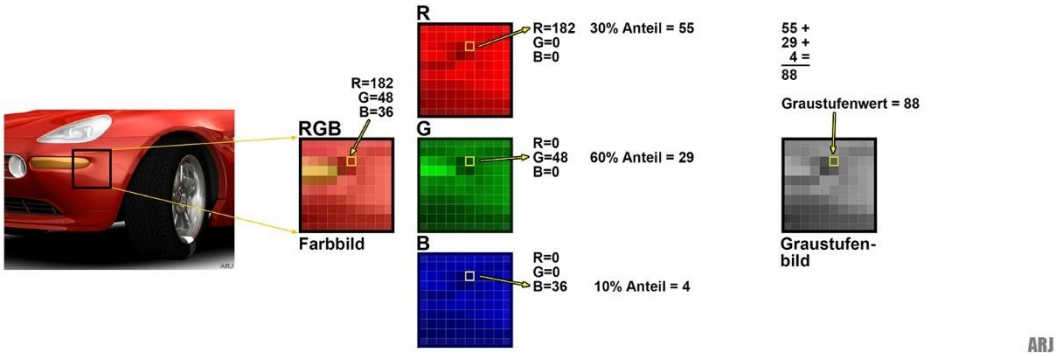
Im Folgenden werden wir über Farbbilder und Graustufenbilder sprechen. Daher zuerst ein kleiner Exkurs in die Biologie.

Die Farbwahrnehmung des Menschen:
Unser Auge ist nicht für alle Lichteindrücke gleichermassen empfänglich. Für Hell/Dunkel (Graustufen) verfügt unser Auge über ca. 100 Millionen helligkeitsempfindliche Stäbchen, wohingegen nur ca. 7 Millionen Zäpfchen der Farbempfindlichkeit dienen und diese dazu noch bei unterschiedlichen Wellenlängen (Farben) unterschiedlich stark reagieren.

Mit diesem Wissen soll nun aufgezeigt werden, wie ein Farbbild in ein Graustufenbild umgewandelt wird. In bestimmten Fällen ist es nämlich nötig, das Farbbild in ein Graustufenbild umzuwandeln. Zum Beispiel bei dem im nächsten Abschnitt behandelten Farbmodel YC_bC_r.

Bei der Umwandlung werden die drei Farbanteile RGB verschieden gewichtet. Dies könnte man mit folgendem historischem Hintergrund erklären: Der frühzeitliche Mensch als Jäger und Sammler war vor allem auf eine hohe Auflösung im Grünbereich (Wälder, Wiesen etc.) angewiesen, um Beute oder herannahende Gefahr besser erkennen zu können. Die Farbe Blau war da eher seltener und darum weniger wichtig. Da diese Aspekte des Farbensehens des menschlichen Auges berücksichtigt werden müssen. So wird beispielsweise Grün heller wahrgenommen als Rot, dieses wiederum heller als Blau.

Diese unterschiedliche Gewichtung wird in folgender Umrechnungsformel berücksichtigt:
Luminanz (Y) = 0.3 x Rot + 0.6 x Grün + 0.1 x Blau

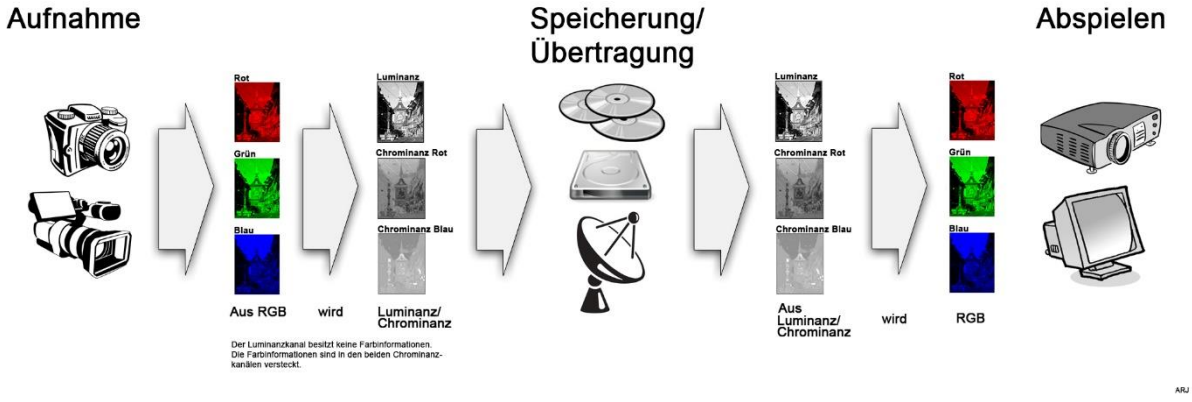


- Unser Auge ist für Grün am empfindlichsten, darum hat Grün im Graustufenbild den grössten Anteil von 60%.
- Weniger empfindlich ist man für Rot. Darum hat Rot nur 30% Anteil.
- Am wenigsten empfindlich ist man für Blau. Sein Anteil ist 10%.

Bei der Farbunterabtastung möchte man nun Daten derart reduzieren, dass die Luminanz (Helligkeit) möglichst unangetastet bleibt und man bei den für unsere Augen weniger relevanten Farben Abstriche macht. Das gelingt allerdings bei RGB nicht. Wir benötigen einen anders definierten Farbraum.

Das Helligkeits-Farbigkeits-Modell oder Luminanz/Chrominanz-Modell.

Die Kamera zeichnet das Bild in RGB auf. Abgespeichert und datenreduziert wird es im Luminanz/Chrominanz-Farbmodell. Damit das Bild wieder angezeigt werden kann, muss es dekomprimiert und vom Luminanz/Chrominanz-Farbmodell in RGB zurückgerechnet werden.



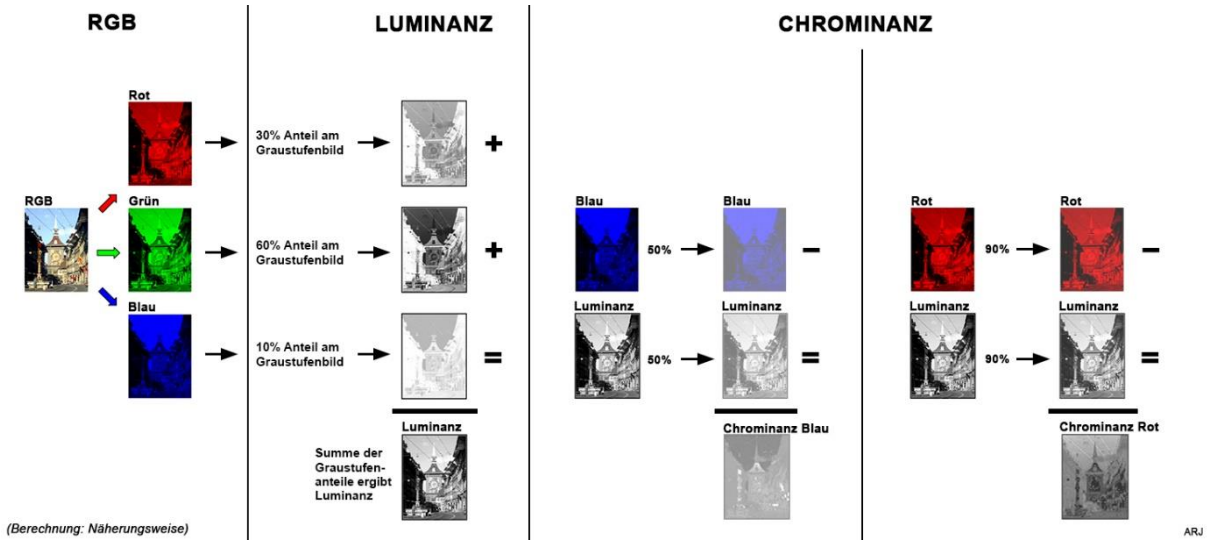
Die Umwandlung RGB zu Luminanz/Chrominanz (YC_bC_r):

Die drei Farbkanäle Rot, Grün und Blau werden in die drei Kanäle Luminanz (Y), Chrominanz Blau (C_b) und Chrominanz Rot (C_r) umgewandelt.

Bei der Umwandlung in den Luminanzkanal (Graustufenbild), hat jeder Farbkanal den Anteil, dem ihm gemäss Physiologie des Auges zusteht: Grün 60%, Rot 30%, Blau 10%).

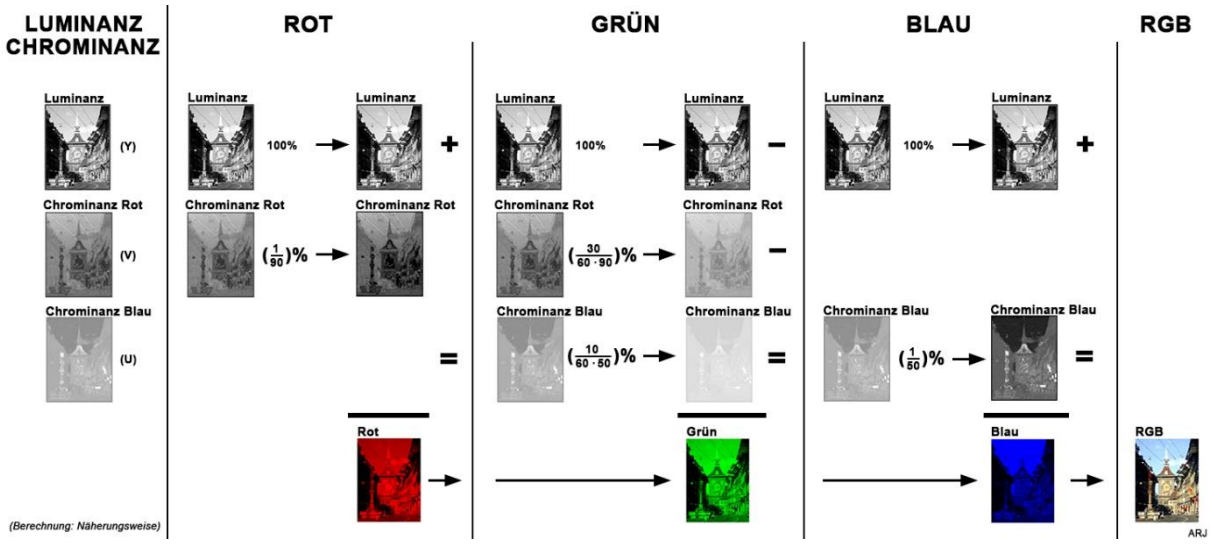
Hier folgen Aufgaben zum Thema. Siehe separates Aufgabenblatt.

Bei der Berechnung der Chrominanzkanälen kommt reine Algebra zur Anwendung.



Die Zurückwandlung Luminanz/Chrominanz (YC_bC_r) zu RGB:

Der Vollständigkeitshalber sei hier auch gezeigt, wie der Weg zurück erfolgt.



Zusammenfassung:

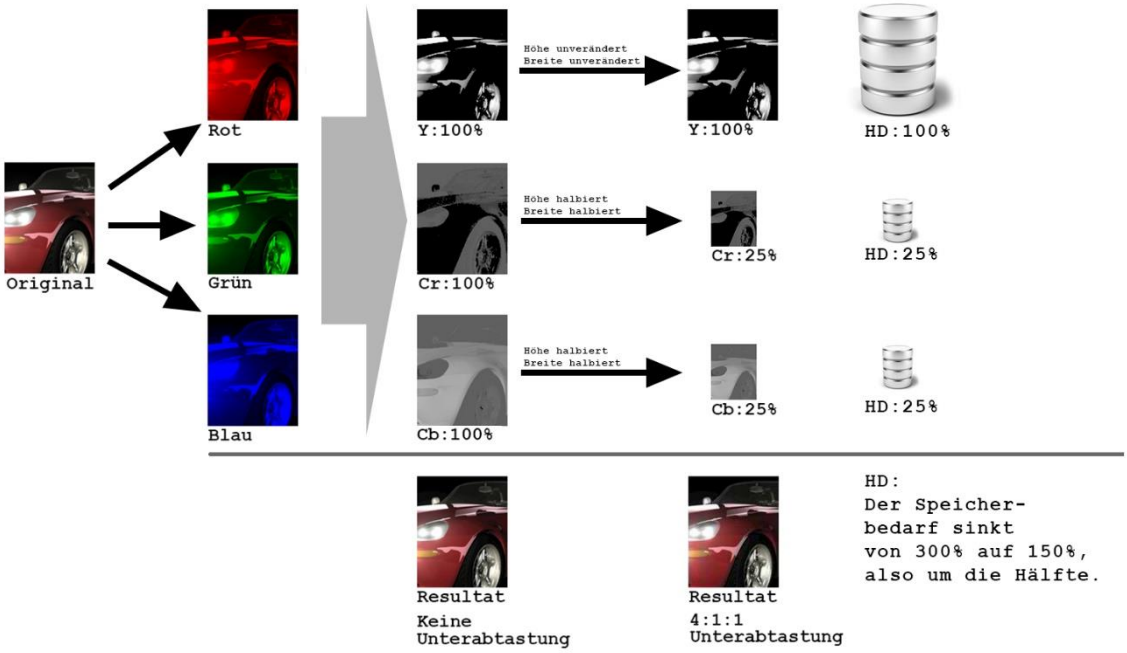
- Das **YC_bC_r-Helligkeits-Farbigkeits-Modell** (gemäss CCIR-601 bzw. IEC 601-Standard) wurde für das Digitalfernsehen entwickelt. Ausserdem wird es für digitale Bild- und Videoaufzeichnung, bei JPG-Bildern, MPEG-Videos und damit auch bei DVDs, sowie den meisten anderen digitalen Videoformaten verwendet. Es teilt die Farbinformation in die (Grund-)Helligkeit Y und die Farbigkeit, bestehend aus den zwei Farbkomponenten C_b (Blue-Yellow Chrominance) und C_r (Red-Green Chrominance) auf.
- Die **Helligkeit** entspricht der Hellempfindlichkeit des Auges, die im grünen Spektralbereich am grössten ist. Chrominance oder kurz Chroma bedeutet Buntheit.
- Die **unterschiedliche Wahrnehmung** von Y gegenüber den C_b- und C_r-Kanälen entspricht der Entwicklung der Farb- und Helligkeitsverteilung in der Natur: Im Laufe der Evolution hat sich der menschliche Sehsinn daran angepasst. Das Auge kann geringe Helligkeitsunterschiede besser erkennen als kleine Farbtonunterschiede, und diese wiederum besser als kleine Farbsättigungsunterschiede. So ist ein Text grau auf schwarz geschrieben gut zu lesen, blau auf rot geschrieben bei gleicher Grundhelligkeit jedoch nur sehr schlecht.
- Die **Analogie zum menschlichen Sehsinn** wird für einen grossen Vorteil von YC_bC_r genutzt: die Farbunterabtastung (engl. chroma subsampling). Dabei wird die Abtastrate und damit die Datenmenge der Chrominanz-Kanäle C_b und C_r gegenüber der Abtastrate des Luminanz-Kanals Y reduziert, ohne dass es zu einer spürbaren Qualitätsverringerung kommt. So kann man z. B. mit der JPEG-Komprimierung eine nicht unerhebliche Datenmenge einsparen.
- Das **YUV-Farbmodell** der analogen Fernsehtechnik wird manchmal fälschlicherweise mit YC_bC_r für digitale Darstellung von Farbvideosignalen gleichgesetzt. Einfachheitshalber bedienen wir uns für die folgenden Erklärungen bei dem YC_bC_r-ähnlichen YUV-Farbmodell. Wer es ganz genau wissen will, konsultiert die einschlägige Fachliteratur oder informiert sich bei den entsprechenden Beiträgen auf Wikipedia.

Hier folgen Aufgaben zum Thema. Siehe separates Aufgabenblatt.

So funktioniert die eigentliche Farbunterabtastung / Subsampling:

Beim Subsampling oder Farbunterabtastung wird die Bildauflösung reduziert. Allerdings nicht in RGB. Dazu ein Beispiel: Reduziert man ein HD1080-TV-Bild von 1920x1080 auf 960x540 hat man zwar die Dateigrösse geviertelt, möchte man das Bild aber wieder auf einem HD1080-Projektor anschauen, wirkt es unscharf, weil es von 960x540 wieder auf 1920x1080 gedehnt werden muss. Bei der Unterabtastung wählt man einen anderen Weg: Man wandelt das RGB-Bild in YC_bC_r (Luminanz/ChrominanzBlau/ChrominanzRot) um. Anstatt wie beim vorangegangenen Beispiel alle drei Kanäle (RGB) in der Pixelauflösung zu reduzieren, wird dies bei Chroma-Subsampling nur in den beiden Chrominanz-Kanälen getan. Damit erhält man zwar eine etwas ungenauere Einfärbung, die Bildschärfe (Luminanz) bleibt aber vollständig erhalten. Es sind folgende Chroma-Subsamplings vorgesehen: 4:4:4, 4:2:2, 4:1:1, 4:2:0. Und was wird damit eingespart? Bei 4:1:1 bedeutet dies Y=100%, C_b=25%, C_r=25% und somit gegenüber dem Original (300%) Dateigrösse halbiert.

Beispiel zu Subsampling:



Anspruchsvollere Verfahren wie DCT, verwendet bei JPG

Wirklich schwierig zu verstehen sind erst die Komprimierungsverfahren, die auf höherer Mathematik beruhen. Hier brauchts schon fast einen Hochschulabschluss, um die Technik dahinter zu verstehen wie z.B. bei der DCT (Discrete Cosine Transformation) für JPG-Komprimierung, wo in Kombination zwar einige bereits besprochene Verfahren wie Huffman, RLC, Chroma-Subsampling zur Anwendung kommen, aber eben auch die diskrete Kosinustransformation, wo uns die mathematischen Kenntnisse dazu fehlen. Wer's trotzdem wissen will, hier ein vereinfachter Erklärungsversuch:

Die DCT-Verarbeitungsschritte

1. Als erster Schritt wird das **RGB-Bild in ein YC_bC_r-Bild** umgewandelt. Danach wird die Auflösung in den beiden Chrominanzkanälen reduziert. (**Chroma-Subsampling**)
Die drei Kanäle (1xLuminanz, 2xCrominanz) werden ab hier separat verarbeitet. Jeder Kanal wird nun in 8x8 Pixelblöcke aufgeteilt.

2. Auf jeden dieser 8x8 Pixelblöcke wird die diskrete **Kosinustransformation** angewandt. Vereinfacht ausgedrückt bedeutet dies, dass die 8x8 Bildwertematrix vom Bildbereich in den sogenannten Frequenzbereich transformiert wird. Im Bildbereich prägen Unterschiede in den Helligkeitswerten die Wertematrix, wohingegen im Frequenzbereich die Schnelligkeit der Helligkeitsänderungen entscheidend sind. (Etwas vereinfacht ausgedrückt: Scharfe Bilder ergeben schnelle Helligkeitsänderungen und damit viele hohe und unterschiedliche DCT-Werte, unscharfe langsamere und somit tiefere und weniger unterschiedliche DCT-Werte.)

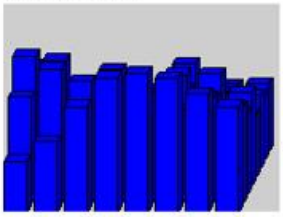
Originalwerte im 8x8 Block

114	115	117	114	109	117	113	115
120	114	117	117	121	113	120	110
118	119	121	114	117	112	119	125
163	149	141	115	130	193	143	105
204	163	133	155	141	163	191	159
236	228	178	163	168	204	163	138
180	228	208	227	186	157	179	163
86	119	179	228	236	227	200	179

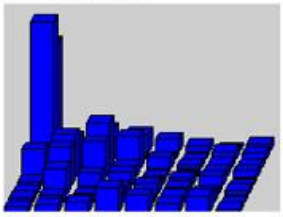
DCT-transformierte Werte

188	15	-22	32	-23	12	-6	-14
-239	10	55	4	3	-2	16	13
-11	-54	-81	-57	20	-8	-9	7
50	74	62	19	-3	2	-11	-12
-8	-67	-22	24	-4	15	17	-2
7	46	2	-14	-20	-13	1	-8
-11	-14	-3	16	-2	-13	18	-1
-9	-13	18	-43	29	15	-28	6

3D-Visualisierung



3D-Visualisierung



3. Die DCT-transformierten Werte werden nun **quantisiert**. Unter Quantisierung ist etwa dasselbe zu verstehen, wie das Notenrunden bei Schulprüfungen: Wenn die Schulnote mit einer Genauigkeit von 1/10 festgehalten wird, hat man mehr mögliche Notenwerte, als wenn die Note auf 0.5 gerundet wird. Allerdings verliert man mit der gröberen, weil ungenaueren 0.5-er Notenskala auch an Aussagekraft. Dasselbe gilt für die quantisierten DCT-Werte. Bis jetzt hat man allerdings noch keine eigentliche Datenreduktion erreicht. Abhängig von der Stärke der Quantisierung der DCT-Werte (im Extremfall werden die meisten Werte zu 0) erhält die 8x8-Matrix eine geeignete Form, für RLC und VLC.

DCT - Matrix:

188	15	-22	32	-23	12	-6	-14
-239	10	55	4	3	-2	16	13
-11	-54	-81	-57	20	-8	-9	7
50	74	62	19	-3	2	-11	-12
-8	-67	-22	24	-4	15	17	-2
7	46	2	-14	-20	-13	1	-8
-11	-14	-3	16	-2	-13	18	-1
-9	-13	18	-43	29	15	-28	6

Hohe Komprimierung

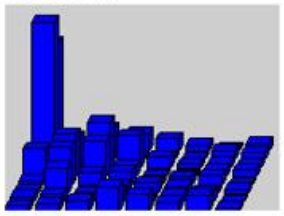
Quantisierungsmatrix:

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Quantisierte Matrix:

188	15	-22	32	-23	12	-6	-14
-239	10	55	4	3	-2	16	13
-11	-54	-81	-57	20	-8	-9	7
50	74	62	19	-3	2	-11	-12
-8	-67	-22	24	-4	15	17	-2
7	46	2	-14	-20	-13	1	-8
-11	-14	-3	16	-2	-13	18	-1
-9	-13	18	-43	29	15	-28	6

Visualisierung:



Mittlere Komprimierung

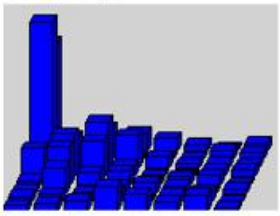
Quantisierungsmatrix:

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Quantisierte Matrix:

12	1	-2	2	-1	0	0	0
-20	1	4	0	0	0	0	0
-1	-4	-5	-2	0	0	0	0
4	4	3	1	0	0	0	0
0	-3	-1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Visualisierung:



Niedrige Komprimierung

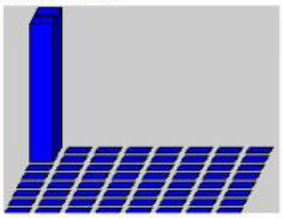
Quantisierungsmatrix:

255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Quantisierte Matrix:

1	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Visualisierung:

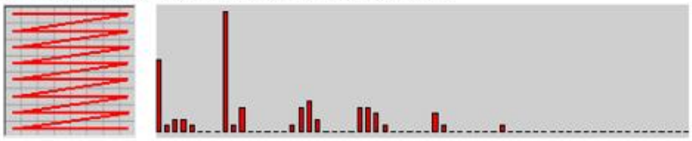


4. Nun folgt eine Zick-Zack-Scan.

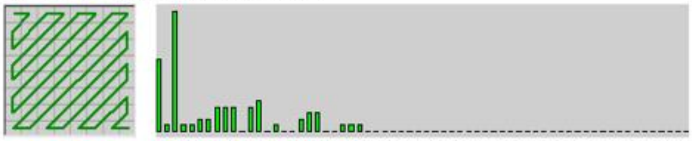
Quantisierte Matrix:

12	1	-2	2	-1	0	0	0
-20	1	4	0	0	0	0	0
-1	-4	-5	-2	0	0	0	0
4	4	3	1	0	0	0	0
0	-3	-1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Matrixwerte von links nach rechts und von oben nach unten:

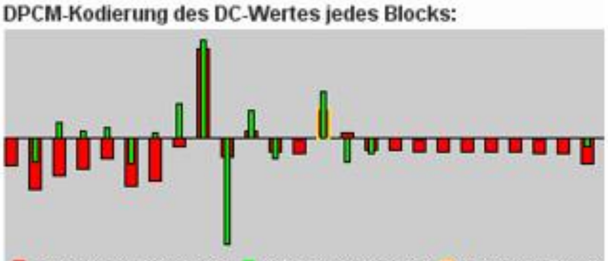


Matrixwerte mit Zick-Zack-Scan Methode:



5. Abgesehen vom Subsampling werden erst jetzt mit RLC Daten reduziert.

DPCM-Kodierung des DC-Wertes jedes Blocks:



RLE- und Variable-Length-Integer-Kodierung der AC-Werte:

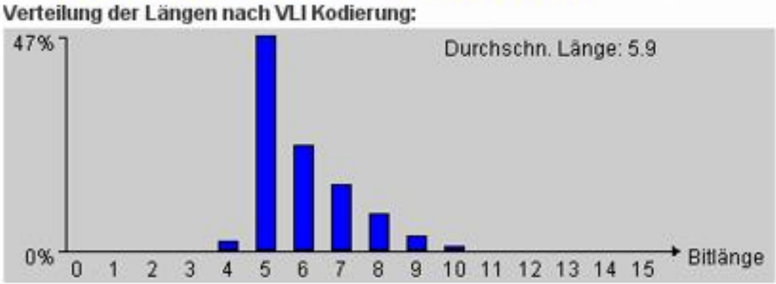
Vor Run Length-Encoding:

1 -20 -1 1 -2 2 4 -4 4 0 4 -5 0 -1 0 0 -2 3 -3 0 0 1 -1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

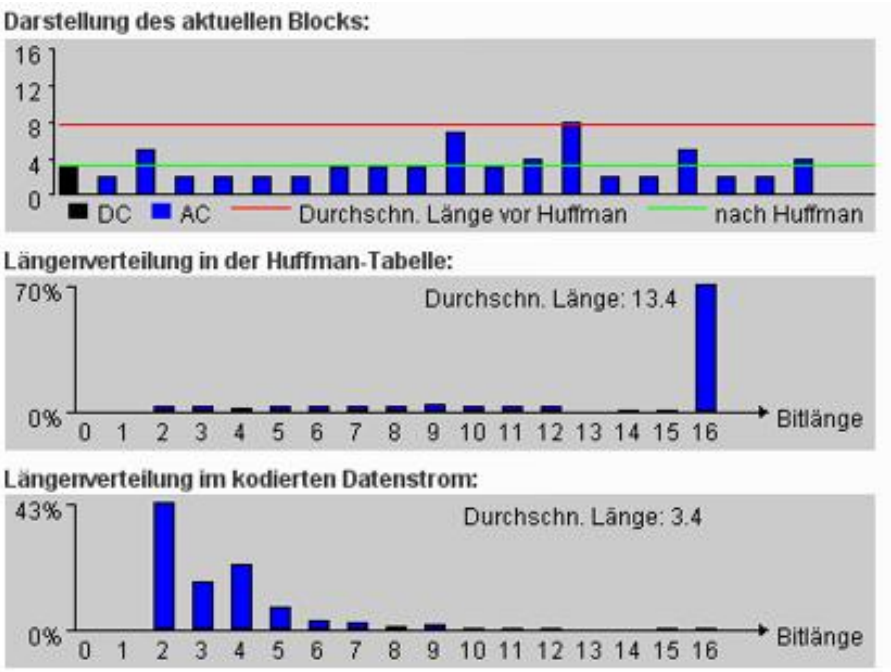
Nach Run Length-Encoding:

1 -20 -1 1 -2 2 4 -4 4 1 4 -5 1 -1 2 -2 3 -3 2 1 -1 1 39

Verteilung der Längen nach VLI Kodierung:

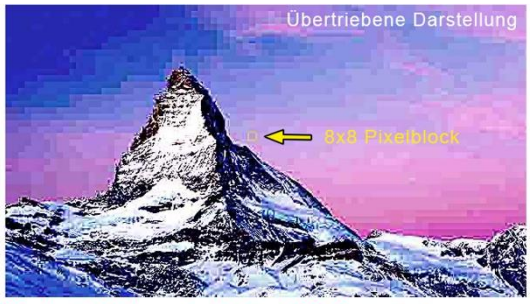


6. Darauf folgt ein VLC in Form von Huffman



Schlussfolgerung zu DCT: Wenn bei der DCT zu stark quantisiert wird, gleichen sich die Farben der 8x8 Pixelblöcke einander an und es entsteht eine Blockstruktur. Man spricht dann von **Blocking-Artefakten**:

Eine starke JPG-Komprimierung führt zu unsauberen Textkonturen. (Zur besseren Sichtbarmachung des Effekts wurde das Bild nachträglich gammakorrigiert)



Blockartefakte nach starker JGP-Komprimierung. Führt von der 8x8-Blockbildung bei DCT.

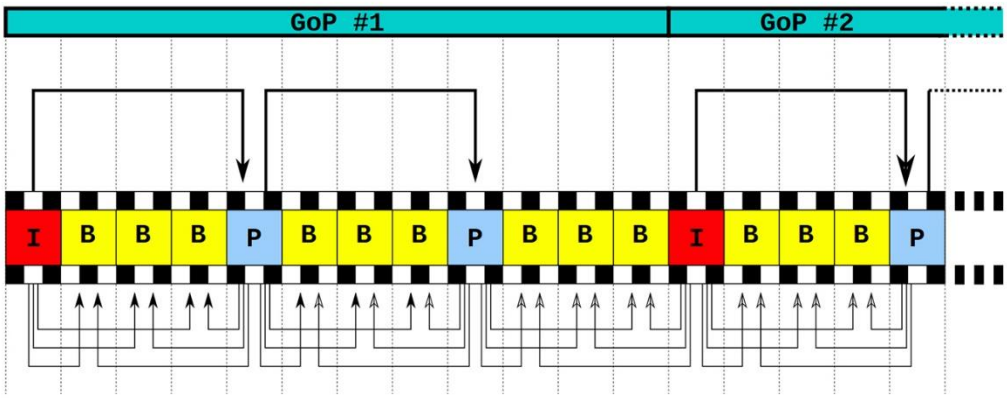
Ein kleiner Ausblick in die Audiokomprimierung (MP3)

MP3 ist eigentlich MPEG-1/2 Audio Layer 3 und damit ein Verfahren zur verlustbehafteten Kompression digital gespeicherter Audiodaten. Zur Komprimierung, die gegenüber einer Original-WAV-Datei durchaus bis zu über 85% betragen kann, werden psychoakustische Effekte der menschlichen Wahrnehmung von Tönen und Geräuschen ausgenutzt. Von MP3-Playern unterstützt werden Datenraten zwischen 8 kb/s und 320 kb/s. Wer mehr dazu erfahren will, dem sei z.B. der Wikipedia-Eintrag zu MP3 empfohlen.

Intraframe und Interframe

Bisher haben wir ausschliesslich Verfahren betrachtet, die sich auf ein **einzelnes Bild** beziehen. Man bezeichnet dies auch **Intraframe-Komprimierung**. Es ist aber auch möglich innerhalb einer **Bildsequenz** Bandbreite oder Speicher zu sparen. Dies nennt man dann **Interframe-Komprimierung**. Und das geht so: Anstatt in einem Videostream pro Sekunde 24 komplette Bilder zu verschicken (oder speichern), könnte man ja nach einem Vollbild (i-Frame) nur noch die Änderung des Bildinhalts zum vorangegangenen Bild übermitteln.

Allerdings darf man das nicht übertreiben. In regelmässigen Abständen sollte ein Vollbild verfügbar sein, damit die Filmwiedergabe bei einem Übermittlungsfehler neu synchronisieren kann. Man spricht dabei von einer **GOP** (Group-of-Pictures). Je nach Dynamik der Filmszene gibt das eine immense Einsparung an Daten. Filmt man 10 Minuten lang seine geschlossene Haustüre ohne weitere Action, würde dies bei GOP25 einer Datenreduktion von so etwa 96% bedeuten. BTW: Ein ähnliches Verfahren kennen wir beim Backup, wo zwischen Fullbackup und inkrementellem Backup unterschieden wird.



Hier folgen Aufgaben zum Thema. Siehe separates Aufgabenblatt.