

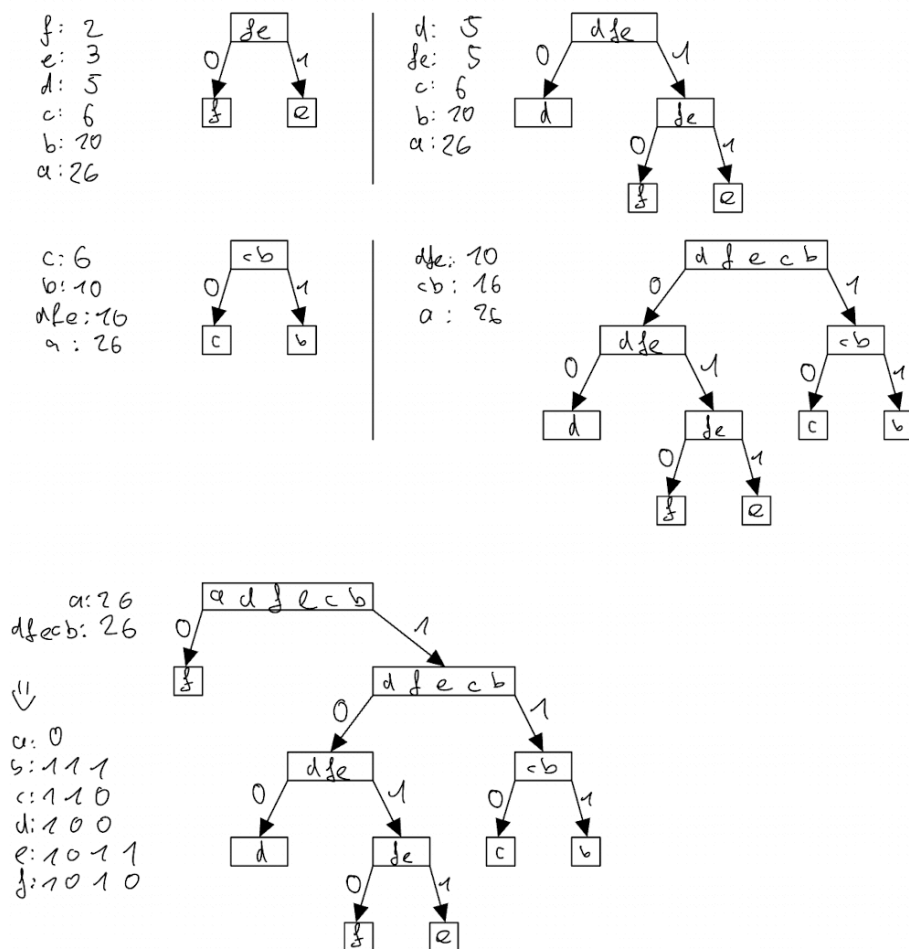
1.7 JPEG

a.) Erläutern Sie wie bei der Anpassung an die menschliche Wahrnehmung (Schritt 1 der jpeg Kompression) eine Kompression von 50% erreicht wird

- Reduktion des Datenvolumens in den Farbkanälen
- Helligkeitskanal bleibt unverändert
- Schritte
 - o Konvertierung von RGB nach YCbCr
 - o Berechnung des Mittelwerts von 4 benachbarter Pixel -> Reduktion auf $\frac{1}{4}$
 - o Speicherbedarf insg: $1 \cdot \#Y + \frac{1}{4} \cdot \#Cb + \frac{1}{4} \cdot \#Cr \rightarrow \frac{1}{2} \cdot (\#Y + \#Cb + \#Cr)$
- Es gibt verschiedene Abtastverhältnisse wie man die Werte mittelt. Üblich 4:2:0
- Dies ist sinnvoll da die menschliche Farbwahrnehmung ist geringer aufgelöst als das Helligkeitsempfinden

b.) Gegeben sind folgende Häufigkeiten, berechnen Sie die Huffman Codierung

Huffman-Codierung für 6 Buchstaben. Gewichtung war gegeben mit: 26, 10, 6, 5, 3, 2



c.) Erläutern Sie das Grundprinzip der Run-Length Encoding (RLE) und geben Sie ein Beispiel an

- Jede Sequenz von identischen Symbolen wird durch deren Anzahl ersetzt
- Bei fester Bit-Breite ist **kein Trennzeichen nötig**
- man muss definieren mit welchem Symbol man beginnt
- Beispiel: 00000000111100000011
 - o 8 4 6 1 \rightarrow 1000 0100 0110 0001

d.) Erläutern Sie das Grundprinzip der Huffman-Codierung. Nennen und erklären Sie die wesentliche Eigenschaft der Codierung, die eine Einsparung von Platz überhaupt erst möglich macht.

- Idee: Codiere die auftretenden Symbole gemäß ihrer Häufigkeit. Häufige Symbole erhalten kürzere Codes, seltene Symbole längere Codes. Je ungleichmäßiger die Verteilung desto besser die Komprimierung.
- Eigenschaften:
 - Zuordnung ist Injektiv
 - Präfix-Frei, ein Teil Code am Anfang für ein Symbol darf nicht ein vollständiger Code eines anderen Symbols sein. Nur so kann die Decodierung zuverlässig funktionieren

e.) Aufgaben zur Matrix nach der Quantisierung:

a. 12 AC-Komponente markieren

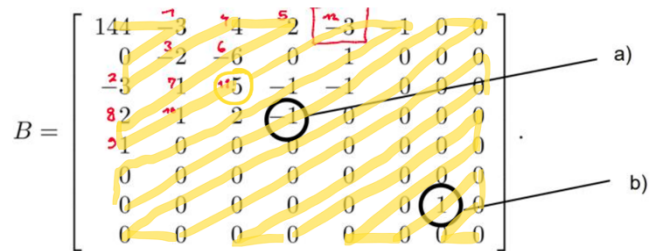
- (0,3)(5)

b. Tupel für a angeben

- (5,1)(-1)

c. Tupel für b angeben

- (15,0)(15,0)(1,1)(-1)



f.) Wieso wird bei der Codierung der DC- und AC-Komponenten nach der Erstellung der Tupel nur das jeweils erste Symbol (SIZE bzw. RUNLENGTH,SIZE) mit Huffman codiert, nicht aber der zweite Teil (AMPLITUDE)

- Da dieser beliebige Werte enthalten kann und deshalb es keine extremen Häufungen geben wird. Huffman Codierung wird in dem Fall keinen Nutzen haben

g.) Erläutern Sie, welche Effekte bei der JPEG-Kompression durch das Bearbeiten in 8×8 Blöcken entstehen. (Mit Begründung, schwierige Aufgabe) und weitere Artefakte inkl. Begründung

- Blockbildung: sichtbare 8×8 Blöcke
 - Wenn man zu viel unterdrückt, dass nur noch der DC-Komponenten der Blöcke da ist. Da jeder Block für sich selbst komprimiert wird. Dann gibt es einen klaren Helligkeitswechsel von den Durchschnittshelligkeiten
- Rining: scharfen Kanten ergeben wellenartige Strukturen um Kanten herum
 - Entsteht weil zu niedrig frequentierte Schwingen scharfe Kanten abbilden sollen, dadurch entstehen Ober- Unterschwinger. Diese können nicht diese Strukturen abbilden wie Kanten und approximieren es schlecht

- a) Erklären Sie wie die Quantisierung funktioniert? (2. Schritt) Quantisierungsmatrix erklären, warum nicht einfach Schwellwert?
- Die Quantisierungsmatrix versucht die Koeffizientenmatrix in den 8x8 Blöcken zu quantisieren (Datenreduktion) und unrelevante Koeffizienten zu verwerfen, indem die auf 0 gesetzt werden.
 - Dabei wird die Koeffizienten Matrix durch die Quantisierungsmatrix geteilt.
 - $F'(u, v) = \text{round}\left(\frac{F(u, v)}{Q(u, v)}\right)$
 - Die Quantisierungsmatrix orientiert sich an der menschlichen Wahrnehmung und versucht Koeffizienten, die für hohe Frequenzen stehen härter zu "bestrafen" indem durch höhere Werte geteilt wird.
Also kleinere Werte in der linken oberen Hälfte und größere Werte in der rechten unteren Hälfte
 - **Es wird keine Schwellwertoperation durchgeführt, weil dieser viel unflexibler ist und nicht so gut steuerbar ist wie stark man einzelne Koeffizienten reduziert werden sollen**
- b) Erläutern Sie, wie genau die AC- und DC-Komponenten bei der JPEG-Kompression mit der RLE jeweils weiterverarbeitet werden. Erläutern Sie, welche Vorteile die jeweilige Vorgehensweise bietet Ergänzen Sie Ihre Erklärung mit einer Skizze.
- Bei DC wird nicht RLE gemacht
 - Bei AC sinnvoll, weil die Koeffizienten Matrix sehr viele Einträge mit 0 hat
 - Besonders ist, dass die Matrix zick-zack förmig iteriert wird, weil oben links eher Koeffizienten $\neq 0$ und unten recht Einträge = 0 liegen. Dadurch wird gewährleistet, dass RLE dann möglichst lange 0er-Ketten abbildet
- c) DC und AC-Komponenten sehr ausführlich erklären mit Codierung und was warum wie codiert wird warum? (Size), (Runlength, size)
- DC ($f = 0$) Komponenten ist der erste Eintrag in den 8x8 Blöcken, die für die Durchschnittshelligkeit stehen
 - DC wird durch 2 Symbole dargestellt und codiert (SIZE, AMPLITUDE).
 - **SIZE** gibt Größe der Amplitude, also Koeffizient an und wird mit Huffman codiert
 - **AMPLITUDE** wird mit **Differential Pulse Code Modulation (DPCM)** also z.B. die Folge 150, 149, 155 140 wird so codiert 150, -1, 5, -15
 - AC ($f > 0$) Komponenten sind die restlichen Koeffizienten-Einträge in den 8x8 Blöcken
 - AC wird durch 3 Symbole dargestellt und codiert (RUNLENGTH, SIZE) (AMPLITUDE).
 - **RUNLENGTH** und **SIZE** sind 4 Bit groß, also zusammen 1 Byte.
 - **RUNLENGTH** gib die Anzahl an 0en vor einer nicht 0 AC Komponente an. Hier wird **Run Length Encoding RLE** gemacht und **RUNLENGTH & SIZE** wird mit Huffman codiert.
 - Die **AMPLITUDE** wird nicht mit Huffman codiert, da es nicht von einer ungleichmäßigen Verteilung auszugehen ist
 - **SIZE** wird bei DC und AC zusätzlich **RUNLENGTH** mit Huffman codiert, da sehr häufig die gleichen Werte auftauchen.
 - **RLE** bietet sich an, weil vor allem bei der rechten unteren Hälfte Blöcke sehr viele 0en zu erwarten sind.
 - **DPCM** bei DC bietet sich an, weil die die Amplituden sehr ähnliche Werte haben und nur leicht voneinander abweichen
- h.) Huffman bei DC und AC-Komponente warum? (SIZE), (RUNLENGTH, SIZE)
- Da dort die Verteilung der Werte ungleichmäßig ist und häufig die gleichen Werte auftauchen. Diese Symbole bekommen dann kurze Codes und führt zu einer starken Kompression
- d) Warum geht man nicht im Zickzack-Muster und schneidet einfach nach einem Wert ab?
- Damit mit hoher Wahrscheinlichkeit eine lange Null-Kette entsteht.