

Ansätze zur Kompression von Bildern

Codierung der Farbwerte

Mit 24 Bit pro Pixel können $2^{24} = 16.777.216$ Farben unterschieden werden. Die Bilder A bis D verwenden jeweils deutlich weniger Farben, so dass sich diese auch mit deutlich weniger als 24 Bit unterscheiden lassen. Die Anzahl der unterscheidbaren Farben hängt von der Anzahl der zur Codierung verwendeten Bits ab. Mit 1 Bit pro Pixel lassen sich zwei Farben unterscheiden, mit 2 Bit lassen sich 4 Farben unterscheiden usw. Da die Codierung dann nicht mehr der Darstellung im RGB-Modell entspricht, muss zusätzlich festgelegt werden, welche Bitfolge welche Farbe codiert.¹

Aufgabe 1:

- Bestimmen Sie für die Bilder A bis D, wie viele Bits jeweils benötigt werden, um alle Farben eindeutig zu codieren.
- Legen Sie für jedes Bild eine Codetabelle an, die jeder Farbe einen Binärcode zuordnet. Bei welchen Bildern wäre eine Umrechnung ins RGB-Modell auch ohne Farbtabelle möglich?
Hinweis: Auf der nächsten Seite finden Sie eine Vorlage für die Tabellen.
- Berechnen Sie für jedes Bild den Speicherplatz, der benötigt wird, wenn für die Codierung der Pixel Ihre Farbtabelle aus b) verwendet wird. Vergleichen Sie mit der Codierung, die 24 Bit pro Pixel verwendet. Geben Sie dazu jeweils das *Kompressionsverhältnis* und die *prozentuale Datenersparnis* an.

Hinweise:

- Das *Kompressionsverhältnis* ist das Verhältnis der Größe der komprimierten Daten zur Größe der Originaldaten: $\text{Kompressionsverhältnis} = \text{Größe der komprimierten Daten} / \text{Größe der Originaldaten}$
- Die *prozentuale Datenersparnis* gibt an, wie viel Prozent an Speicherplatz durch die Kompression eingespart wurde:
 $\text{Prozentuale Datenersparnis} = (1 - \text{Kompressionsverhältnis}) \cdot 100 \%$
- Der Speicherplatz für die Farbtabelle soll hier nicht berücksichtigt werden.

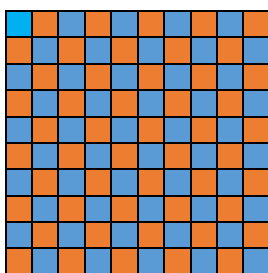


Bild A

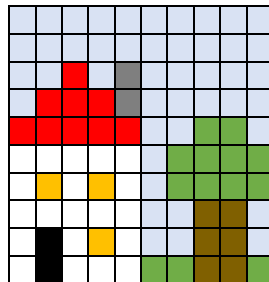


Bild B

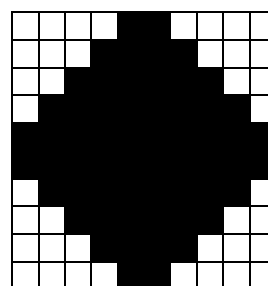


Bild C

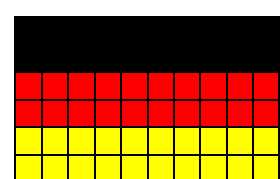


Bild D

¹ Bei der Anzahl der zur Codierung verwendeten Bits spricht man auch von **Farbtiefe**. Häufig bezieht sich die Farbtiefe dann auf die Anzahl der Bits, die jeweils pro Farbkanal, also im RGB-Modell für den Rot-, den Grün- und den Blauwert zur Verfügung stehen.

Bild A

Farbe	RGB-Wert	Code
Blau	91, 155, 213	
Orange	237, 125, 49	

Bild B

Farbe	RGB-Wert	Code
Schwarz	0, 0, 0	
Rot	255, 0, 0	
Gelb	255, 192, 0	
Grün	112, 173, 71	
Hellblau	217, 226, 243	
Braun	128, 96, 0	
Grau	127, 127, 127	
Weiß	255, 255, 255	

Bild C

Farbe	RGB-Wert	Code
Schwarz	0, 0, 0	
Weiß	255, 255, 255	

Bild D

Farbe	RGB-Wert	Code
Schwarz	0, 0, 0	
Rot	255, 0, 0	
Gelb	255, 255, 0	

Vorlage für Codetabellen zu Aufgabe 1b

Lauflängencodierung

Bei manchen Bildern lässt sich die Datenersparnis zusätzlich erhöhen, indem aufeinanderfolgende gleichfarbige Pixel zusammengefasst werden. Legt man für Bild C fest, dass die Bildgröße 10 x 10 Pixel beträgt und das Bild zeilenweise codiert wird, so lässt sich das Bild wie folgt beschreiben: 4 x weiß, 2 x schwarz, 7 x weiß, 4 x schwarz usw.

Eine noch kürzere Schreibweise wäre z. B.: 4w2s7w4s...

Eine solche Codierung, in der die Anzahl aufeinanderfolgender gleicher Werte angegeben wird, anstatt den Wert entsprechend oft zu wiederholen, bezeichnet man als Lauflängencodierung.

Aufgabe 2:

- a) Setzen Sie die Lauflängencodierung für das Bild C fort.
- b) Geben Sie eine Lauflängencodierung für die Bilder B und D an.

Aufgabe 3: Die Lauflängencodierung aus Aufgabe 2a) für Bild C soll nun binär dargestellt werden.

- a) Begründen Sie, dass in Bild C fünf Bit ausreichen, um jeweils die Anzahl der gleichfarbigen Pixel zu codieren. Wie viel Bit werden allgemein für ein Bild mit 100 Pixeln benötigt, um die Anzahlen zu codieren?
- b) Wie könnten 20 gleichfarbige aufeinanderfolgende Pixel codiert werden, wenn nur vier Bit für die Codierung der Anzahl zur Verfügung stehen?
- c) Die Lauflängencodierung für Bild C beginnt mit 4 x weiß, 2 x schwarz, 7 x weiß, 4 x schwarz. Geben Sie die entsprechende binäre Darstellung an, wenn die Farbe Schwarz mit 0 und die Farbe Weiß mit 1 codiert werden.
- d) Untersuchen Sie, ob bei einem einzelnen Pixel einer Farbe die 1 als Anzahl angegeben werden muss.
- e) Berechnen Sie, wie viele Bits für die binäre Darstellung der Lauflängencodierung von Bild C benötigt werden.
- f) Begründen Sie: Wenn bei einem Schwarz-Weiß-Bild festgelegt wird, dass immer mit der Anzahl schwarzer Pixel begonnen wird (das sind ggf. 0), kann auf die Codierung der Farbe verzichtet werden.
- g) Berechnen Sie, wie viele Bits nach dem Verfahren in e) für die Lauflängencodierung des Bildes C benötigt werden.
- h) Berechnen Sie die prozentuale Datenersparnis für die Lauflängencodierung des Bildes C aus e) bzw. f) im Vergleich zu einer pixelweisen Codierung gemäß RGB-Modell mit 24 Bit. Vergleichen Sie die Lauflängencodierung des Bildes C auch mit einer pixelweisen Codierung mit Ihrer Farbtabelle aus Aufgabe 1.
- i) Konstruieren Sie ein Schwarz-Weiß-Bild mit 10 x 10 Pixeln, bei dem eine Lauflängencodierung zu einer besseren Kompressionsrate führt als die reine Reduktion der Farbcodierung je Pixel auf ein 1-Bit.

Aufgabe 4: Untersuchen Sie, für welche der Bilder A, B und D eine Lauflängencodierung sinnvoll erscheint. Berechnen Sie ggf., wie viele Bits benötigt werden und bestimmen Sie die prozentuale Datenersparnis bezogen auf die Codierung mit Ihrer Farbtabelle aus Aufgabe 1.

Aufgabe 5: Erläutern Sie allgemein, wie ein Bild aufgebaut sein muss, damit eine Lauflängencodierung eine hohe Kompression erzielt.

Huffman-Codierung

Für Bild B werden mindestens 3 Bit benötigt, um alle Farben codieren zu können. Da einige Farben deutlich häufiger verwendet werden als andere, kann hier eine zusätzliche Kompression dadurch erfolgen, dass häufige Farben kürzere Codewörter erhalten als seltene Farben. Da die Codewörter der einzelnen Farben dann keine feste Länge mehr haben, muss darauf geachtet werden, dass die Codierung eines Bildes trotzdem eindeutig decodiert werden kann. Das erreicht man z. B., indem man einen Code konstruiert, der **präfixfrei** ist. Präfixfrei bedeutet, dass kein Codewort der Anfang eines anderen Codeworts ist.

Aufgabe 6: Tabelle 1 zeigt vier verschiedene Codes für fünf Farbtöne.

- Untersuchen Sie für jeden Code, ob er präfixfrei ist und ob sich eine Codierung mit diesem Code jeweils eindeutig decodieren lässt.
- Begründen Sie, dass eine Codierung, die mit einem präfixfreien Code erzeugt wurde, immer eindeutig decodierbar ist.

Farbe	Code 1	Code 2	Code 3	Code 4
Weiß	0	0	01	10
Rot	1	10	10	100
Gelb	00	110	001	1000
Grün	01	101	110	10000
Blau	10	1110	0001	100000

Tabelle 1: verschiedene Codes für Aufgabe 6

Einen präfixfreien Code, der die Häufigkeiten der zu codierenden Informationen berücksichtigt, erhält man durch die Konstruktion eines **Huffman-Baums**. Um einen Huffman-Baum für die in Bild B verwendeten Farben zu erstellen, muss zunächst gezählt werden, wie häufig die einzelnen Farben verwendet wurden:

Farbe	RGB-Wert	Anzahl	Huffman-Code
Weiß	255, 255, 255	20	
Rot	255, 0, 0	9	
Gelb	255, 192, 0	3	
Grün	112, 173, 71	13	
Hellblau	217, 226, 243	45	
Braun	128, 96, 0	6	
Grau	127, 127, 127	2	
Schwarz	0, 0, 0	2	

Tabelle 2: In Bild B verwendete Farben mit ihrer jeweiligen Anzahl

Ein Huffman-Baum ist ein Binärbaum, der beginnend bei den Blättern bis hin zur Wurzel aufgebaut wird. Für jede Farbe wird ein Blatt erstellt. Das Blatt erhält die Anzahl, wie oft die Farbe auftritt, als Wert. Nun erhalten immer die beiden freien Knoten (Knoten, die noch keinen Vater haben) mit dem geringsten Wert einen gemeinsamen Vaterknoten. Der Vaterknoten erhält als Wert die Summe der

beiden Kindknoten. Gibt es mehrere freie Knoten mit kleinstem Wert, wird zufällig einer ausgewählt. Das Erzeugen von Vaterknoten wird so lange fortgesetzt, bis der Baum einen eindeutigen Wurzelknoten erhält und alle Knoten verbunden sind. Die Teilbäume werden also nach und nach zu einem Baum zusammengesetzt. Die Kanten von jedem Vater zu seinen zwei Kindern werden schließlich mit 0 bzw. 1 beschriftet, z. B. alle linken Kanten mit 0 und alle rechten Kanten mit 1. Abbildung 1 zeigt den Huffman-Baum, der so für Bild B entsteht.

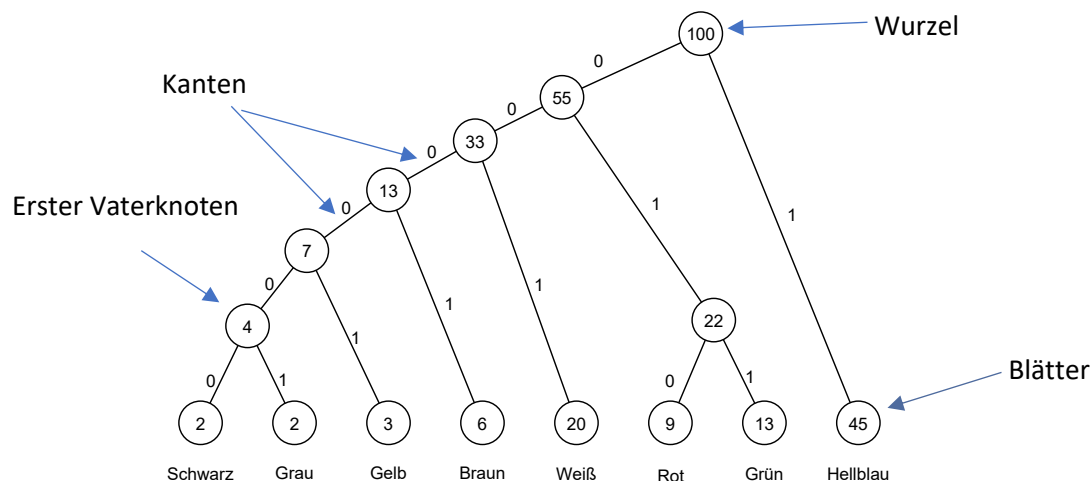


Abbildung 1: Huffman-Baum zu Bild B

Die Pfade von der Wurzel bis zum Blatt geben nun die Codierung der jeweiligen Farbe an. Hellblau erhält z. B. den Code 1, Rot den Code 010 und Grau den Code 000001

Der Algorithmus für die Konstruktion des Huffman-Baums sorgt dafür, dass häufig auftretende Farben einen kurzen Code und seltene Farbwerte einen längeren Code erhalten. Da der Code-Baum für jedes Bild individuell erstellt wird und ggf. Zufallsentscheidungen enthält, muss der Baum der Codierung des Bildes hinzugefügt werden.

Huffman-Bäume können auch für die Kompression anderer Daten, z. B. die Zeichen eines Textes verwendet werden.

Aufgabe 7: Ergänzen Sie in der Tabelle 2 in der rechten Spalte den Huffman-Code der einzelnen Farben entsprechend dem Huffman-Baum in Abbildung 1.

Aufgabe 8:

- Die Bitfolge 010010010010010110110111 ist die Huffman-Codierung einer Zeile des Bildes B. Decodieren Sie die Farben mithilfe des Huffman-Baums in Abbildung 1, um herauszufinden, um welche Zeile es sich handelt.
- Berechnen Sie die prozentuale Datenersparnis, die sich ergibt, wenn das Bild B mit dem Huffman-Code aus Abbildung 1 statt mit 3 Bit pro Farbe bzw. Pixel codiert wird.

Aufgabe 9: Beim Erstellen des Huffman-Baums in Abbildung 1 wurde an einer Stelle eine Zufallsentscheidung getroffen. Identifizieren Sie die Stelle und geben Sie an, wie der Huffman-Baum aussieht, wenn an dieser Stelle anders entschieden wird.

Aufgabe 10: Ist die Erstellung eines Huffman-Baums auch für die Bilder A, C und D auf Seite 1 sinnvoll? Begründen Sie jeweils.

Aufgabe 11:

- Erstellen Sie für das Bild in Abbildung 2 einen Huffman-Baum.
- Erstellen Sie die Huffman-Codierung der zweiten Zeile des Bildes.
- Berechnen Sie die prozentuale Datenersparnis, die sich ergibt, wenn das Bild in Abbildung 2 mit Ihrem Huffman-Code statt mit 3 Bit pro Farbe bzw. Pixel codiert wird.
- Begründen Sie, dass der Huffman-Baum, den Sie für das Bild in Abbildung 2 erstellt haben, ungeeignet ist, um das Bild in Abbildung 3 effizient zu codieren.

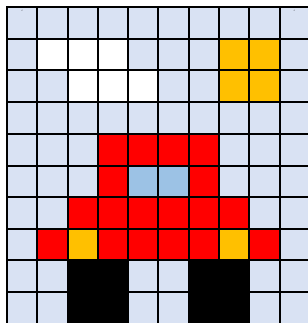


Abbildung 2: Bild zu Aufgabe 11a bis c

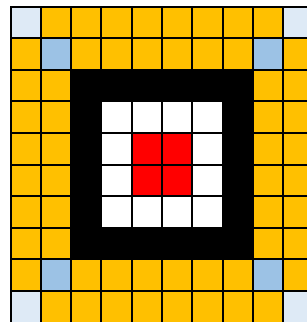


Abbildung 3: Bild zu Aufgabe 11d

Aufgabe 12: Begründen Sie, dass es sich bei den hier vorgestellten Verfahren um verlustfreie Kompressionsverfahren handelt.

Dateiformate für Bilder

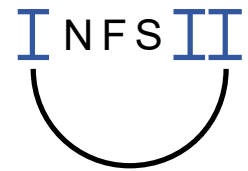
Aufgabe 13: Sammeln Sie Informationen, die in einer Bilddatei gespeichert sein müssen. Welche Aufgabe hat in diesem Zusammenhang das Dateiformat?

Gängige Dateiformate für Bilder sind z. B. *jpg*, *png*, *gif* oder *bmp*. Je nachdem welches Dateiformat beim Speichern eines Bildes verwendet wird, erfolgt eine mehr oder weniger starke Kompression der Daten. Häufig werden dabei verschiedene Verfahren, sowohl verlustfreie als auch verlustbehaftete, kombiniert. Meist spielt die Reduktion der Farbtiefe dabei eine Rolle. Manche Bildbearbeitungsprogramme bieten dem Anwender an, die Farbtiefe beim Speichern eines Bildes zu wählen. Das Format *gif* verwendet grundsätzlich eine maximale Farbtiefe von 8 Bit, wobei die 8 Bit den Farben anhand einer Tabelle zugeordnet werden.

Die Effektivität der Kompression hängt nicht nur von dem verwendeten Verfahren, sondern auch von der Art des Bildes ab. Für digitale Fotos eignen sich andere Verfahren als für Grafiken.

Aufgabe 14:

- Der Ordner *Bilder_Aufgabe14* enthält drei verschiedene Bilder in unterschiedlichen Dateiformaten. Vergleichen Sie den Speicherbedarf der Bilder in den verschiedenen Formaten. Welches Dateiformat erzielt für das jeweilige Bild die beste Kompression?
- Wählen Sie ein eigenes Bild aus und speichern Sie es mithilfe eines Bildbearbeitungsprogramms in verschiedenen Dateiformaten ab. Für welche Dateiformate können Sie die Farbtiefe wählen?
- Beim Abspeichern eines Bildes im Format *jpg* kann häufig die Qualität in Prozent gewählt werden. Speichern Sie ein Bild, z. B. Bild 2 aus dem Ordner *Bilder_Aufgabe14* in verschiedenen Qualitäten ab. In welchen Fällen können Sie den Qualitätsverlust mit bloßem Auge wahrnehmen?



Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Von der Lizenz ausgenommen ist das InfSII-Logo.