

# Kernvorlesung Multimediale Systeme (VL 375, SS2002)

Mathis, Marc, marc.mathis@access.unizh.ch, 99-713-414

## A-57: Verbale Beschreibung der Kodierungs- und Dekodierungstechnik nach Huffman

Beim Huffman- Verfahren werden die Zeichen individuell codiert. Häufig vorkommende Zeichen werden mit wenigen Bits, selten vorkommende Zeichen mit mehr Bits codiert. Dies ist massiv weniger als im Vergleich dazu, der ASCII- Code, ein in der Datenverarbeitung sehr weit verbreiteter Standard, der für jedes Zeichen 8 Bit benötigt. Zur Verdeutlichung wird später noch ein Beispiel aufgeführt.

Zur Festlegung, welches Zeichen mit wie vielen Bits codiert werden soll, müssen also Informationen über die Zeichenhäufigkeiten vorhanden sein. Hierfür existieren in der Praxis drei Möglichkeiten:

- **statisch:** Die Zeichenhäufigkeiten werden vorher festgelegten Tabellen entnommen. Der Vorteil hierbei ist die schnelle Verarbeitung, andererseits entfällt eine hinreichend präzise Voraussage von Zeichenhäufigkeiten.
- **dynamisch:** Die Daten werden jeweils einmal für jedes Dokument ganz gelesen, um die dort geltenden Häufigkeiten zu bestimmen. Da das Kodierungsschema sich erst im Laufe der Kompression ergibt, muss man neben den eigentlichen Daten auch die dazugehörige Code-Tabelle übertragen.
- **adaptierend:** Das adaptive Verfahren versucht die Vorteile der beiden obigen Verfahren miteinander zu verbinden. Sie verändern das Codierungsschema im Verlauf der Kompression anhand der gelesenen Daten. So verwenden sie zu Beginn der Kompression einen festen Code (z.B.: alle Zeichen treten gleich oft auf, oder das e ist das häufigste Zeichen, usw..) und bestimmen dann jeweils nach dem Verarbeiten einer vorgegebenen Anzahl von Bytes anhand der inzwischen ermittelten Häufigkeiten einen Neuen.

Zum Beispiel wollen wir das Wort „**astatama**“ nach Huffman codieren.

Die dynamische Bestimmung der Zeichenhäufigkeit ergibt:

- 4 x "a"    - 2 x "t"    - 1 x "s"    - 1 x "m"

Stehen die Häufigkeiten der vorkommenden Zeichen fest, können die Bitkombinationen in passenden Längen verteilt werden. Zu Beginn wählt man die beiden seltensten Zeichen. Sie bekommen eine 0 beziehungsweise 1 codiert.

- **s:** 0 (Häufigkeit = 1)
- **m:** 1 (Häufigkeit = 1)

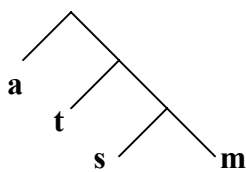
Nun fasst man beide zusammen und ersetzt sie in der Aufstellung der Häufigkeiten durch ihre Summe. Dann werden wieder die beiden seltensten gesucht, eins um 0 das andere um 1 für die Codierung vorne erweitert und die beiden zusammengefasst.

- **t**: 0 (Häufigkeit = 2)
- **s**: 10 ; **m**: 11 (Häufigkeit 1 x "s" und 1 x "m" = 2)

Dieses Verfahren endet, wenn nur noch ein Zeichen übrig ist. Dieses Zeichen erhält eine 0 und die anderen Werte eine 1.

- **a**: 0
- **t**: 10 ; **s**: 110 ; **m**: 111 (Häufigkeit alle zusammen = 4)

Man kann die Ermittlungen des Codes auch in einem Huffman- Baum darstellen:



Von der Wurzel ausgesehen (oben):  
Nach *links* bedeutet es wird eine **0** geschrieben.  
Nach *rechts* bedeutet es wird eine **1** geschrieben.

Das ergibt „**01101001001110**“ als Bitfolge.

Dieser Baum dient zum Kodieren und Dekodieren nach Huffman.

Beim **Kodieren** geht man vom Zeichen aus zur Wurzel und erhält so den Binärcode, der für das Zeichen stehen soll. Die Länge des Codes ist die Länge des zurückgelegten Weges.

Das **Dekodieren** beginnt an der Wurzel mit dem Einlesen eines Bits. Je nach Inhalt verzweigt man im Baum nach der einen **0** oder anderen **1** Seite, liest das nächste Bit und folgt wieder entsprechend der nächsten Verzweigung. Am Ende, gewissermassen auf einem Blatt des Baumes, steht das gesuchte Zeichen.

Im obigen Beispiel ergibt das „**0-110-10-0-10-0-111-0**“ wenn man Bit für Bit einliest bis man zu einem Blatt des Baumes kommt, was dem gesuchtem Wort „**astatama**“ entspricht.

Wichtig bei dieser Vorgehensweise ist, dass man den Codierbaum oder Übersetzungstabelle mitliefern muss, um aus dem Code wieder die ursprünglichen Symbole zu dekodieren.

Das Wort „**astatama**“ wird, wenn man so vorgeht, mit der Bitfolge „**01101001001110**“ codiert, entspricht 14 Bit. Im Vergleich dazu benötigt der ASCII- Code mit 8 Bit pro Zeichen beim obigen Wort 64 Bit (8 x 8). Für vier verschiedene Zeichen genügen eigentlich 2 Bit, was bei einer Wortlänge von acht Zeichen insgesamt 16 Bit verbrauchen würde, also immer noch schlechter als die Huffman- Codierung.

Das Verfahren nach Huffman sorgt nicht nur für eine *eindeutige* und *verlustfreie* Codierung, sondern ist auch *optimal*, d.h., die Länge des codierten Textes wird minimal.