

Kompression

Vortrag

Jonathan Balack

Betreuerin: Kira Duwe

—
Fachbereich Informatik

Fakultät für Mathematik, Informatik und Naturwissenschaften

Universität Hamburg

29.01.2019

Gliederung

- 1 Motivation
- 2 Redundanz
- 3 Verlustfreie Kompression
- 4 Verlustbehaftete Kompression
- 5 Zusammenfassung
- 6 Quellen



Motivation

- Effizientere Nutzung von Speicher- und Übertragungsinfrastruktur
- Reduzierung der Datenmenge
 - ohne den Informationsgehalt (stark) zu verringern
- Daten beinhalten oft Redundanzen

Redundanz

- Verteilungsredundanz
 - Unterschiedliche Wahrscheinlichkeit für jedes Zeichen
- Bindungsredundanz
 - Wahrscheinlichkeit des Auftreten, bedingt durch vorherigen Zeichen
- Redundanz in der Wahrnehmung
 - Qualität der Daten, die für Information nicht relevant ist

Morsecode

- Entwickelt um 1837
- Länge jedes Zeichens invers zur Häufigkeit
- nicht variabel
- nicht präfixfrei

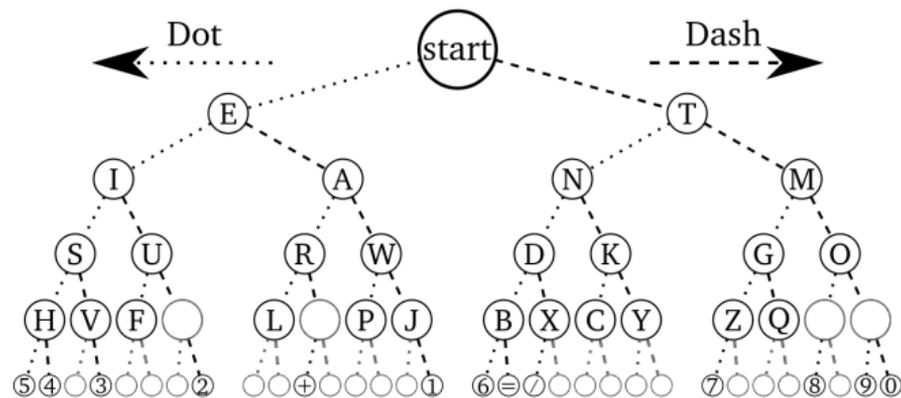


Abbildung 1: Binärbaum des Morsecodes

commons.wikimedia.org

Huffman-Kodierung

- 1952 veröffentlicht von David Huffman
- Binärbaum von unten nach oben aufgebaut
- optimal
- präfixfrei und variabel
- adaptive Methode möglich

Huffman-Kodierung

- Beispielwort: ROKOKOKOKOTTEN
- Sortierung der Zeichen nach Häufigkeit
 - 5*O, 4*K, 2*T, 1*E, 1*R, 1*N

Huffman-Kodierung

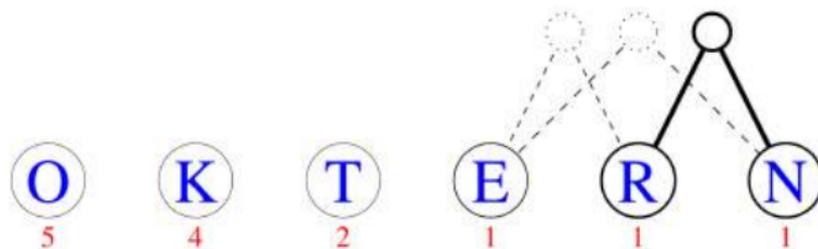


Abbildung 2: Beispiel eines Binärbaums nach Huffman

Studienarbeit von Oliver Schmid | Universität Mannheim

Huffman-Kodierung

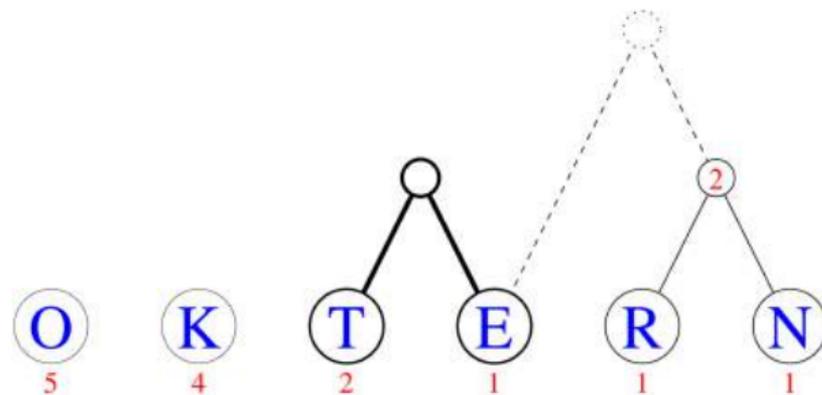


Abbildung 2: Beispiel eines Binärbaums nach Huffman

Studienarbeit von Oliver Schmid | Universität Mannheim

Huffman-Kodierung

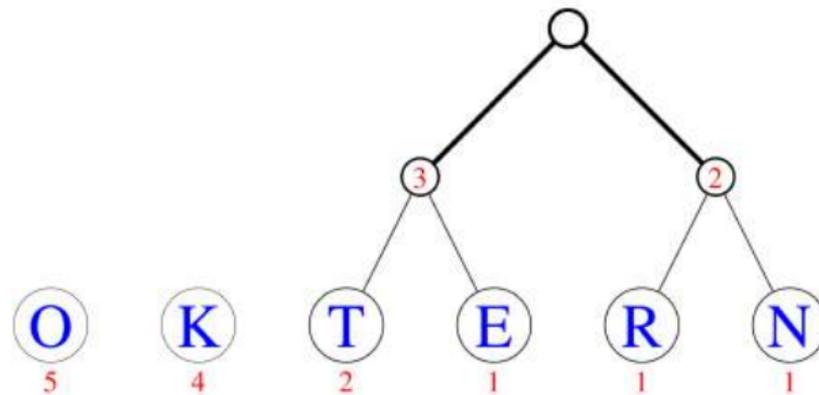


Abbildung 2: Beispiel eines Binärbaums nach Huffman

Studienarbeit von Oliver Schmid | Universität Mannheim

Huffman-Kodierung

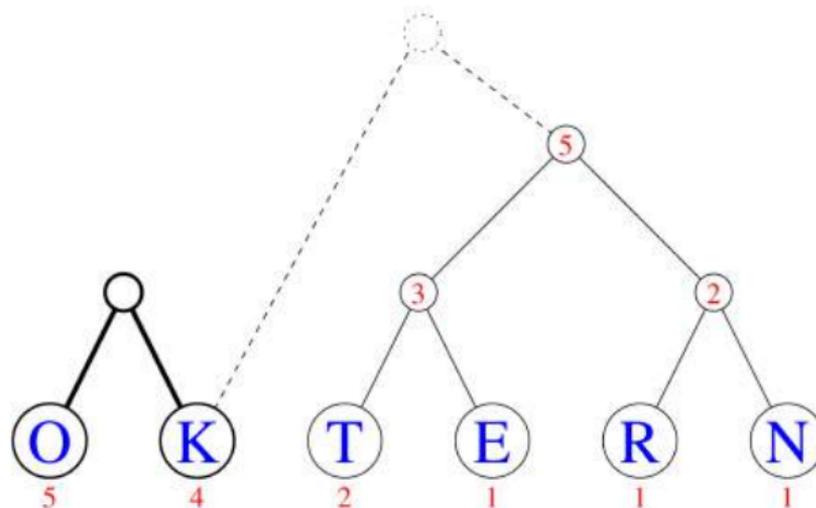


Abbildung 2: Beispiel eines Binärbaums nach Huffman

Studienarbeit von Oliver Schmid | Universität Mannheim

Huffman-Kodierung

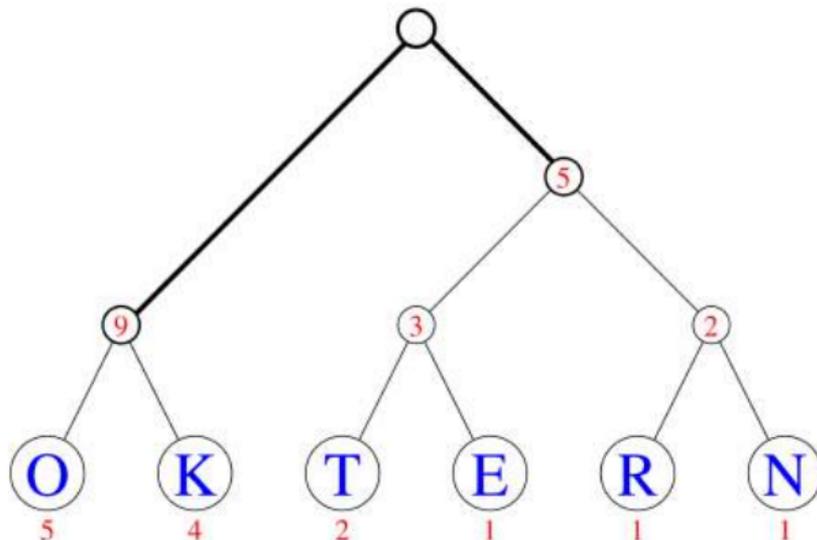


Abbildung 2: Beispiel eines Binärbaums nach Huffman

Studienarbeit von Oliver Schmid | Universität Mannheim

Huffman-Kodierung

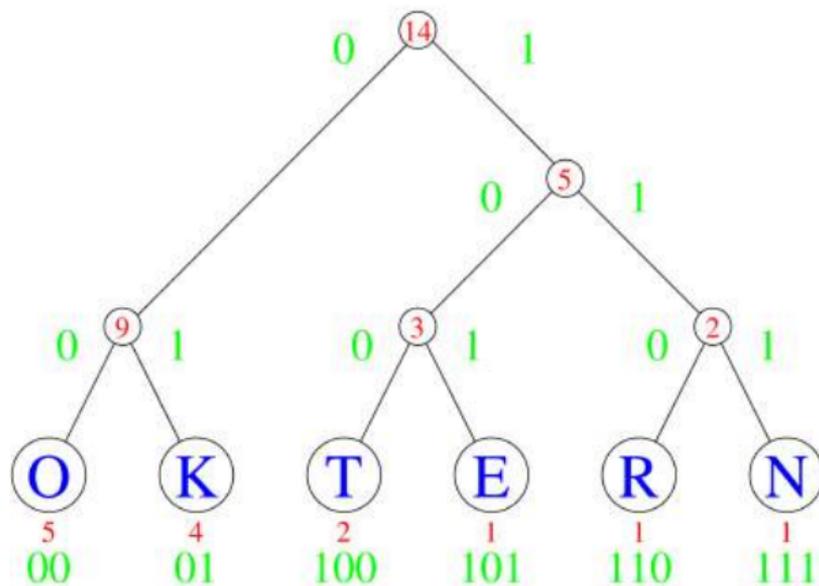


Abbildung 2: Beispiel eines Binärbaums nach Huffman

Studienarbeit von Oliver Schmid | Universität Mannheim

Huffman-Kodierung

R	O	K	O	K	O	K	O	K	O	T	T	E	N
110	00	01	00	01	00	01	00	01	00	100	100	101	111

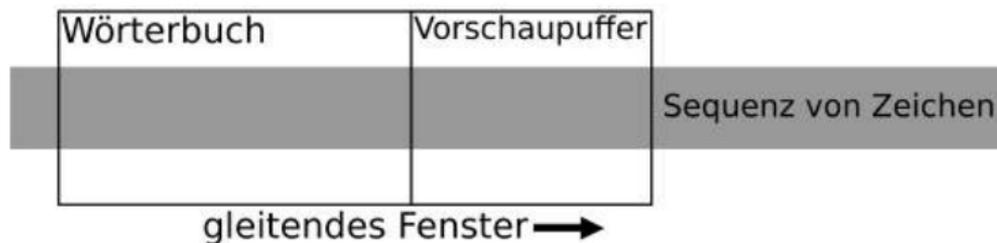
- Huffman-Kodierung: 33bit + Huffmanbaum
- ASCII: $14 * 8\text{bit} = 112\text{bit}$

LZ-Familie

- LZ77 1977 veröffentlicht
- LZ78 1978 veröffentlicht
- von Abraham Lempel und Jacob Ziv
- variabel und präfixfrei
- Ausnutzung von sich wiederholenden Sequenzen
- viele Weiterentwicklungen, wie LZW, LZSS, LZMA und DEFLATE

LZ77

- Fenster gleitet über Sequenz
- Sucht im Wörterbuch nach Sequenz aus Puffer
- Wörterbuch in Praxis mehrere Tausend Zeichen groß
- Vorschau-puffer etwa 100 Zeichen lang
- Ausgabe besteht aus Tripel in der Form: (pos,len,next)



LZ77

11	10	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	Ausgabe
											r	o	k	o	k	o	k	o	t	t	(0,0,r)

Beispielablauf der LZ77-Kompression
Beispielwort: rokokokotter

LZ77

11	10	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	Ausgabe
											r	o	k	o	k	o	k	o	t	t	(0,0,r)
									r	o	k	o	k	o	k	o	t	t	e		(0,0,o)

Beispielablauf der LZ77-Kompression

Beispielwort: rokokotter

LZ77

11	10	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	Ausgabe
											r	o	k	o	k	o	k	o	t	t	(0,0,r)
									r	o	k	o	k	o	k	o	t	t	e		(0,0,o)
								r	o	k	o	k	o	k	o	t	t	e	r		(0,0,k)

Beispielablauf der LZ77-Kompression

Beispielwort: rokokotter

LZ77

11	10	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	Ausgabe
											r	o	k	o	k	o	k	o	t	t	(0,0,r)
									r	o	k	o	k	o	k	o	t	t	e		(0,0,o)
								r	o	k	o	k	o	k	o	t	t	e	r		(0,0,k)
							r	o	k	o	k	o	k	o	t	t	e	r			(2,5,t)

Beispielablauf der LZ77-Kompression

Beispielwort: rokokotter

LZ77

11	10	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	Ausgabe	
											r	o	k	o	k	o	k	o	t	t	(0,0,r)	
									r	o	k	o	k	o	k	o	k	o	t	t	e	(0,0,o)
								r	o	k	o	k	o	k	o	k	o	t	t	e	r	(0,0,k)
							r	o	k	o	k	o	k	o	k	o	t	t	e	r		(2,5,t)
		r	o	k	o	k	o	k	o	t	t	e	r									(1,1,e)

Beispielablauf der LZ77-Kompression

Beispielwort: rokokotter

LZ77

11	10	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	Ausgabe	
											r	o	k	o	k	o	k	o	t	t	(0,0,r)	
									r	o	k	o	k	o	k	o	k	o	t	t	e	(0,0,o)
								r	o	k	o	k	o	k	o	k	o	t	t	e	r	(0,0,k)
							r	o	k	o	k	o	k	o	k	o	t	t	e	r		(2,5,t)
		r	o	k	o	k	o	k	o	t	t	e	r									(1,1,e)
r	o	k	o	k	o	k	o	t	t	e	r											(11,1,-)

Beispielablauf der LZ77-Kompression

Beispielwort: rokokotter

Vergleich

Kompressionsprogramm	Kompression	Dekompression	kompr. Größe [B]	Verhältnis
memcpy	8657 MB/s	8891 MB/s	211947520	100.00
yalz77 -1	71 MB/s	358 MB/s	93952728	44.33
yalz77 -12	14 MB/s	344 MB/s	84050625	39.66
lzsse8 -12	6.08 MB/s	2842 MB/s	75464339	35.61
libdeflate -1	117 MB/s	570 MB/s	73318371	34.59
lzma 16.04 -9	1.55 MB/s	67 MB/s	48742901	23.00

Vergleich verschiedener Kompressionsalgorithmen

Testbedingungen: Ein Kern des Intel Core i5-4300U, Windows 10 64-bit.

Datensatz: Silesia compression corpus

github.com/inikep/lzbench

Grenzen

- Kolmogorow-Komplexität
 - Länge der kürzesten Anleitung für einen Datensatz
 - AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA → 25 Zeichen
 - *Schreibe 25 mal A* → 17 Zeichen
 - $A*25$ → 4 Zeichen
- Taubenschlagprinzip
 - 16 bits → $2^{16} = 65536$ mögliche Informationen
 - 15 bits → $2^{15} = 32768$ mögliche Informationen

Verlustbehaftete Kompression

- scheinbar 'unwichtige' Informationen gehen verloren
- oft an die menschliche Wahrnehmung angepasst
- Kompressionsgrenze frei wählbar

JPEG

- (meist) RGB-Farbraum ins YCbCr-Farbmodell
- Einteilung in 8×8 -Blöcke
- diskrete Kosinustransformation dieser Blöcke
- Quantisierung
- Huffman-Kodierung



Abbildung 3: Aufteilung in die drei Kanäle des YCbCr-Modell

commons.wikimedia.org

Diskrete Kosinustransformation

- Darstellung durch Kosinusfunktion
- Kombination aus 64 Funktionen
- Ausgabe ist der jeweilige Anteil jeder Kosinusfunktion zur Gesamtfunktion

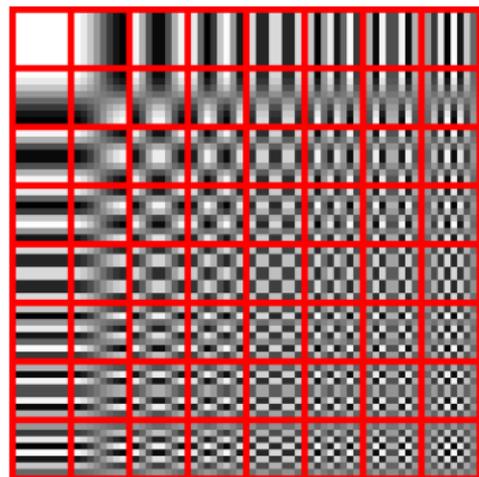


Abbildung 4: Darstellung der 64 Kosinusfunktionen für die DCT

commons.wikimedia.org

Quantisierung

- Tabelle gewichtet Kosinusfunktionen
- $F_{0,0}^Q = \text{round}\left(\frac{782,91}{10}\right) = 78$
- Je nach Komprimierungsstufe verschiedene Tabellen
- Huffman-Kodierung auf der Ausgabe der Quantisierung

$$F = \begin{bmatrix} 782,91 & 44,93 & 172,52 & -35,28 & -20,58 & 35,93 & 2,88 & -3,85 \\ -122,35 & -75,46 & -7,52 & 55,00 & 30,72 & -17,73 & 8,29 & 1,97 \\ -2,99 & -32,77 & -57,18 & -30,07 & 1,76 & 17,63 & 12,23 & -13,57 \\ -7,98 & 0,66 & 2,41 & -21,28 & -31,07 & -17,20 & -9,68 & 16,94 \\ 3,87 & 7,07 & 0,56 & 5,13 & -2,47 & -15,09 & -17,70 & -3,76 \\ -3,77 & 0,80 & -1,46 & -3,50 & 1,48 & 4,13 & -6,32 & -18,47 \\ 1,78 & 3,28 & 4,63 & 3,27 & 2,39 & -2,31 & 5,21 & 11,77 \\ -1,75 & 0,43 & -2,72 & -3,05 & 3,95 & -1,83 & 1,98 & 3,87 \end{bmatrix}$$

Abbildung 5: Bilddaten nach der DCT

de.wikipedia.org

Quantisierung

- Tabelle gewichtet Kosinusfunktionen
- $F_{0,0}^Q = \text{round}\left(\frac{782,91}{10}\right) = 78$
- Je nach Komprimierungsstufe verschiedene Tabellen
- Huffman-Kodierung auf der Ausgabe der Quantisierung

$$Q = \begin{bmatrix} 10 & 15 & 25 & 37 & 51 & 66 & 82 & 100 \\ 15 & 19 & 28 & 39 & 52 & 67 & 83 & 101 \\ 25 & 28 & 35 & 45 & 58 & 72 & 88 & 105 \\ 37 & 39 & 45 & 54 & 66 & 79 & 94 & 111 \\ 51 & 52 & 58 & 66 & 76 & 89 & 103 & 119 \\ 66 & 67 & 72 & 79 & 89 & 101 & 114 & 130 \\ 82 & 83 & 88 & 94 & 103 & 114 & 127 & 142 \\ 100 & 101 & 105 & 111 & 119 & 130 & 142 & 156 \end{bmatrix}$$

Abbildung 5: Quantisierungsmatrix

de.wikipedia.org

Quantisierung

- Tabelle gewichtet Kosinusfunktionen
- $F_{0,0}^Q = \text{round}\left(\frac{782,91}{10}\right) = 78$
- Je nach Komprimierungsstufe verschiedene Tabellen
- Huffman-Kodierung auf der Ausgabe der Quantisierung

$$F^Q = \begin{bmatrix} 78 & 3 & 7 & -1 & 0 & 1 & 0 & 0 \\ -8 & -4 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Abbildung 5: Bilddaten nach der Quantisierung

de.wikipedia.org

JPEG Vergleich

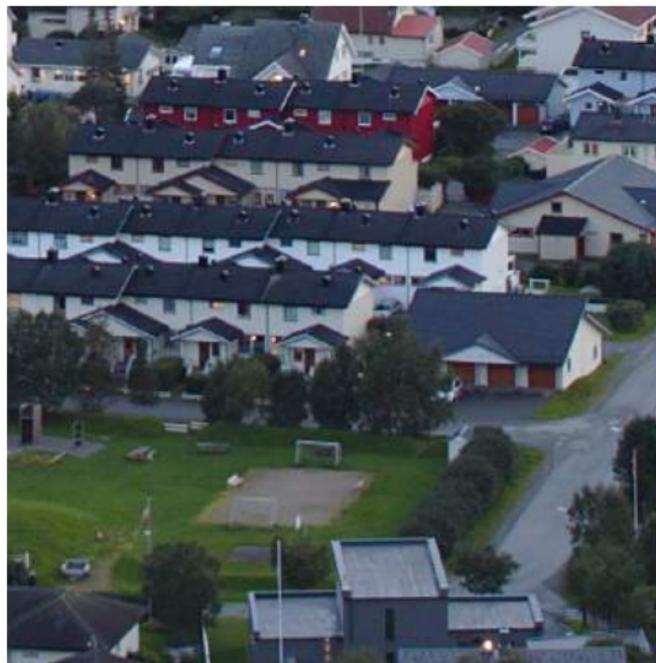


Abbildung 6: JPEG-Beispiel: Qualität 100 | 579.789 B

JPEG Vergleich

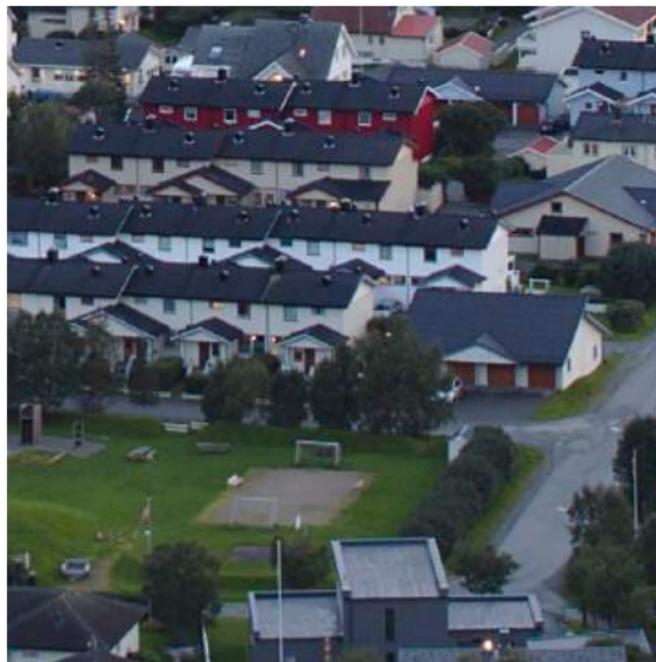


Abbildung 6: JPEG-Beispiel: Qualität 90 | 158.904 B

JPEG Vergleich

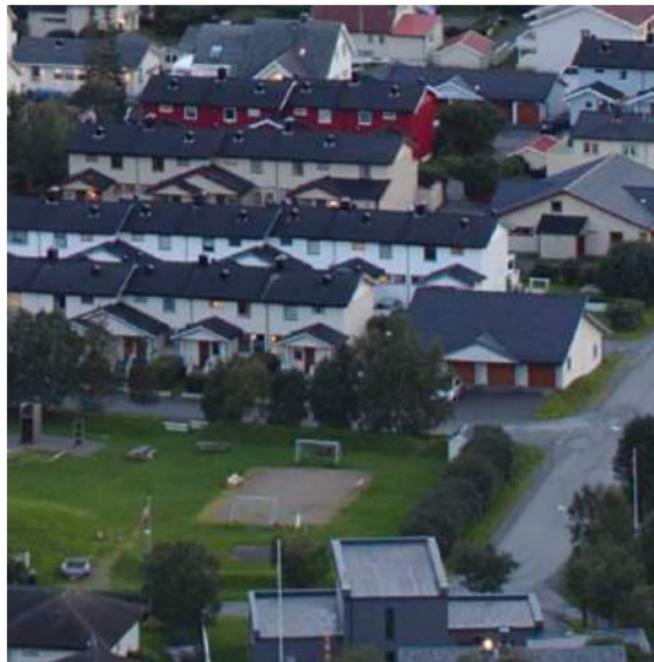


Abbildung 6: JPEG-Beispiel: Qualität 50 | 90.214 B

JPEG Vergleich

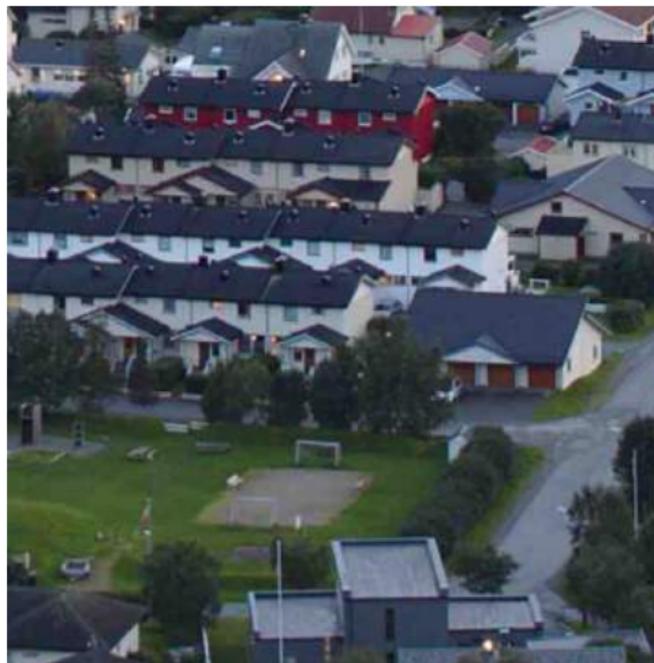


Abbildung 6: JPEG-Beispiel: Qualität 25 | 75.655 B

JPEG Vergleich

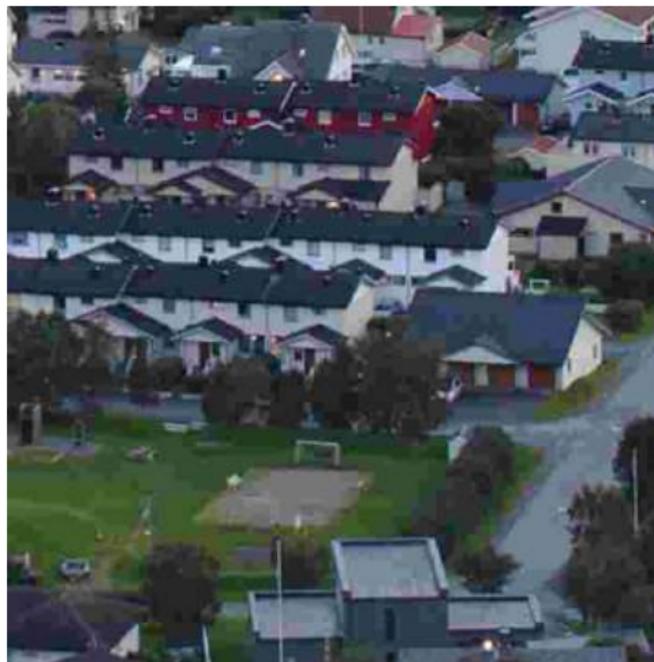


Abbildung 6: JPEG-Beispiel: Qualität 10 | 65.252 B

JPEG Vergleich



Abbildung 6: JPEG-Beispiel: Qualität 5 | 60.587 B

JPEG Vergleich

Format	Kompressionsstufe	kompr. Größe [B]
TIFF 8-bit	0	1.149.702
JPEG 8-bit	100	579.789
JPEG 8-bit	90	158.904
JPEG 8-bit	50	90.214
JPEG 8-bit	25	75.655
JPEG 8-bit	10	65.252
JPEG 8-bit	5	60.587

Größenvergleich verschiedener Bildformate
Programm: darktable

Zusammenfassung

- Reduzierung der Datenmenge
- Ausnutzung von Redundanz
- Verlustfrei
 - Erhalt aller Informationen
 - Einsparung begrenzt
- Verlustbehaftet
 - Reduzierung der Informationen/Qualität
 - Einsparung quasi beliebig

Quellen

- <https://de.wikipedia.org/wiki/Datenkompression>
- <https://pi4.informatik.uni-mannheim.de/pi4.data/content/animations/losslesscompression/studarbeit.pdf>
- <https://www.youtube.com/watch?v=Jqc418tQDkg>
- <https://de.wikipedia.org/wiki/JPEG>
- <https://de.wikipedia.org/wiki/LZ77>
- <https://www.youtube.com/watch?v=Q2aEzeMDHMA>
- <https://de.wikipedia.org/wiki/Huffman-Kodierung>

Bildquellen

- Abb.1 <https://commons.wikimedia.org/wiki/File:Morse-code-tree.svg>
- Abb.2 <https://pi4.informatik.uni-mannheim.de/pi4.data/content/animations/losslesscompression/studarbeit.pdf>
- Abb.3 https://commons.wikimedia.org/wiki/File:Barns_grand_tetons_YCbCr_separation.jpg
- Abb.4 <https://commons.wikimedia.org/wiki/File:Dctjpeg.png>
- Abb.5 <https://de.wikipedia.org/wiki/JPEG>