Technische Hochschule Rosenheim

Vergleich verlustfreier Datenkompressionsverfahren auf Bilddaten

Vergleich **verlustfreier** Datenkompressionsverfahren für Bilddaten

Technische Hochschule **Rosenheim**

- Wieso relevant?
  Speichernutzung, Datenübertragung über Netzwerk

Theoretisch: Informationstheorie/ Grundlagen Datenkompression + Erwartungen
Praktisch: Praktischer Versuch (werden Erwartungen bestätigt?)

Algorithmen:

- Run Length Encoding (RLE)
- Huffman Encoding (Huffman)
- Lempel-Ziv 1977 (LZ77)
- Portable Network Graphics (PNG)
- Verschiedene Kombinationen

Zielsetzung

Technische Hochschule Rosenheim

Warum sind bestimmte verlustfreie Datenkompressionsverfahren für Bilddaten **besser geeignet als andere**?

Wie sind **Bilddaten** aufgebaut und welche Besonderheiten in ihrer Struktur können zur Datenkompression genutzt werden?

(Sind Strukturen bildspezifisch?)

Grundlagen

Technische Hochschule Rosenheim

Information (Claude Shannon): Maß für den Informationsgehalt einer Nachricht

Entropie: Minimale durchschnittliche Menge an Bits, die für die Kodierung einer Nachricht erforderlich ist (untere Schranke)

Redundanz: Information die in Daten mehrfach vorhanden ist (Wiederholungen oder vorhersehbare Strukturen = Überflüssige Information)

Informationstheorie

Technische Hochschule Rosenheim

- Daten vs. Information (Daten nicht gleich Informativ)

Quellencodierungstheorem/ Source Coding Theorem:

- Daten nicht unbegrenzt komprimierbar
- Entropie ist untere Schranke für maximale Kompression (ohne Informationsverlust)

L: mittlere Codierungslänge pro Symbol einer Nachricht

H: Entropie

L >= H

Kolmogorov Komplexität:

- Länge des kürzesten Programms das eine Zeichenfolge erzeugen kann
- Bsp. ”AAAAAAAAABBBBBCCCCCC” → ”9A5B6C”
- Kolmogorov Komplexität für allgemeine Zeichenfolge wegen des Halteproblem nicht praktisch berechenbar → Abschätzung

Datenanforderungen und Komprimierbarkeit

- Strukturierte vs. Unstrukturierte Daten

- Vorverarbeitung der Daten
  (Unstrukturierte Daten → Strukturierten Daten → bessere Kompression der Daten)

- Limitationen: Maximal unstrukturierte Daten (gleichverteilte Zufallszahlen)

  Bsp. „The Random Compression Challenge“ von Mark Nelson

Bilddaten + Wahl der Testbilder

Technische Hochschule Rosenheim

RGB-Format
(8-Bit Farbkanäle)

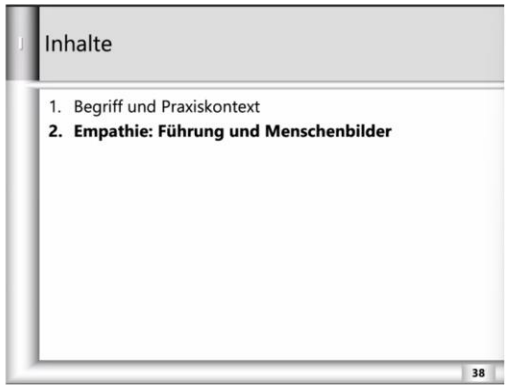
Flagge (290, 174)

Fade 1 (318, 159)

Fade 2 (1670, 954)

Frau (512, 512)

Katzen (1289, 720)



Folie (945, 718)

Messbarkeit

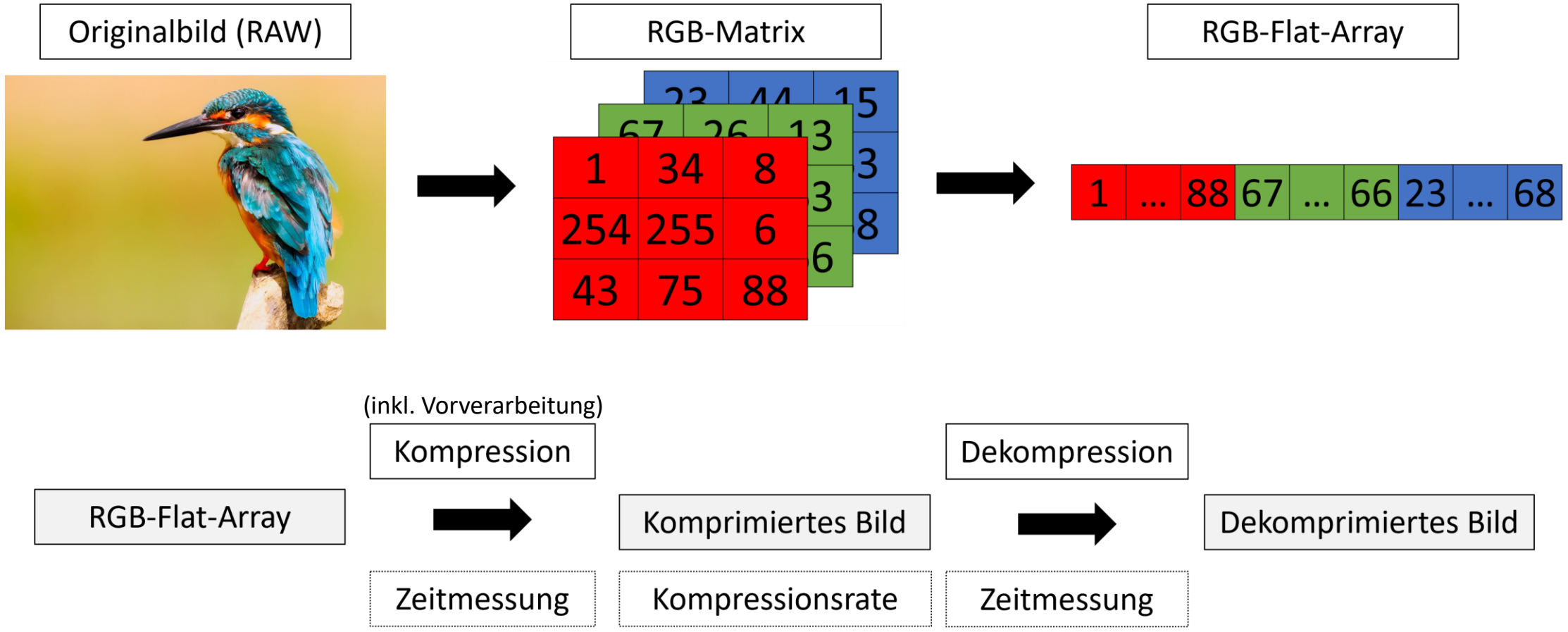
- Kompressionsrate

$$\text{Kompressionsrate} = \left(1 - \frac{\text{Größe komp. Daten}}{\text{Größe unkomp. Daten}}\right) \times 100\%$$

- Kompressionszeit

- Dekompressionszeit

Praktischer Versuch



Bildvorverarbeitung: Filtern

Technische Hochschule Rosenheim

Filtern:

- Vorverarbeitungsschritt des PNG Algorithmus
- Filtern ist umkehrbar
- Ziel unstrukturierte Daten zu strukturierten Daten
  → Redundanz besser erkennbar machen
- Lokale Korrelationen aufzeigen + Wiederholungen erzeugen

Funktionsweise:

- Bild wird zeilenweise gefiltert
- Fünf verschiedene Filtertypen: None, Sub, Up, Average und Paeth
- Heuristic wählt den wahrscheinlich „bestpassensten“ Filter für die jeweilige Zeile aus

Laufzeitkomplexität:

- Filtern: $O(5 * n)$
- Rückgängig machen: $O(m)$

Algorithmen

Technische Hochschule Rosenheim

Algorithmus	Implementierungsdetails	Komp.	Dekomp.
RLE	Reservierte Lauflängenbits: maximale Lauflänge	O(n)	O(m)
Huffman	-	O(n ∗ log n)	O(m)
LZ77	Sliding Window s = 100	O(n ∗ s)	O(m ∗ s)
Deflate	LZ77 + Huffman	O(n ∗ s)	O(m ∗ s)
PNG	• Bildvorverarbeitung: Filter + Deflate • Kein CRC • Sliding Window s = 100, normal bis max. 32768	O(n ∗ s)	O(m ∗ s)
Filter + Huffman	PNG Filter + Huffman	O(n ∗ log n)	O(m)
Filter + Deflate	PNG Filter + Deflate	O(n ∗ s)	O(m ∗ s)

Anmerkung: Implementierungen in Python, nicht optimiert

Code auf GitHub: https://github.com/NickStudRepo/STI-Repo

Erwartungen

- RLE bei Wiederholungen top
- Fade 1 und 2 schwierig für RLE
- Vorverarbeitungsschritt Filtern sollte Verbesserung bringen
- Natürliche Bilder am „schwersten“ komprimierbar (PNG am besten?)

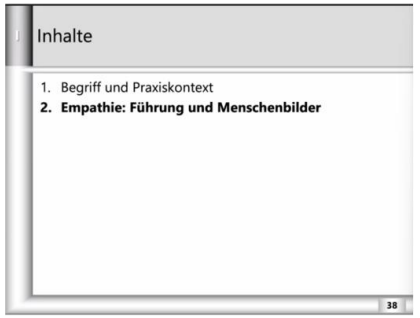
Flagge (290, 174)

Fade 1 (318, 159)

Fade 2 (1670, 954)

Frau (512, 512)

Katzen (1289, 720)



Folie (945, 718)

Kompressionsraten

Technische Hochschule Rosenheim

TABLE I
KOMPRESSIONSRATEN

Name	Auflösung	RLE	Huffman	LZ77	Deflate	Filter + Huffman	Filter + LZ77	PNG
Flagge	(290, 174)	99,98	77,85	78,85	91,70	87,38	84,69	91,80
Fade 1	(318, 159)	45,73	14,30	25,87	31,90	84,94	78,13	80,69
Frau	(512, 512)	-39,19	2,74	-10,96	-9,51	38,22	-7,41	3,38
Folie	(945, 718)	84,57	56,87	73,48	76,62	84,90	81,66	87,03
Katzen	(1289, 720)	-42,85	2,11	-11,39	-9,31	25,49	-10,42	-2,45
Fade 2	(1670, 954)	78,95	10,95	62,13	66,95	85,74	80,59	83,74
Durchschnitt		37,87	27,47	36,33	41,40	67,78	51,21	57,37

[1] Angaben in Prozent, gerundet auf zwei Nachkommastellen
[2] Bilder nach Größe aufsteigend sortiert

Fazit Kompressionsraten

Technische Hochschule Rosenheim

- Filtern bringt durchwegs Verbesserung
- Filtern erzeugt entfernbare Redundanz
- PNG schlechter als erwartet, Sliding Window s = 100
- RLE ist gut für PC-generierte Bilder mit Wiederholungen
- Filter + Huffman überraschenderweise am besten
- Komprimierbarkeit hängt stark von den Bildern selbst ab
- Kein Kompressionsverfahren für alle Bilder am besten → Vergleichen

Kompressionszeiten

Technische Hochschule Rosenheim

TABLE II
KOMPRESSIONSZEITEN

Name	Auflösung	RLE	Huffman	LZ77	Deflate	Filter + Huffman	Filter + LZ77	PNG	Filter
Flagge	(290, 174)	0,015	0,017	2,3	2,3	1,3	3,3	3,3	1,3
Fade 1	(318, 159)	0,028	0,018	14	14	1,6	5	5	1,6
Frau	(512, 512)	0,21	0,09	139	139	9	131	131	9
Folie	(945, 718)	0,47	0,24	63	63	22	62	62	22
Katzen	(1289, 720)	0,77	0,33	530	530	32	540	540	32
Fade 2	(1670, 954)	0,61	0,55	200	200	51	140	140	51

[1] Angaben in Sekunden
[2] Bilder nach Größe aufsteigend sortiert
[3] Die Spalte Filter gibt die Filterzeit an

Dekompressionszeiten

Technische Hochschule Rosenheim

TABLE III
DEKOMPRESSIONSZEITEN

Name	Auflösung	RLE	Huffman	LZ77	Deflate	Filter + Huffman	Filter + LZ77	PNG	Revert Filter
Flagge	(290, 174)	0,003	0,012	0,120	0,124	0,124	0,255	0,255	0,120
Fade 1	(318, 159)	0,032	0,048	3,5	3,5	0,008	0,500	0,550	0,220
Frau	(512, 512)	0,395	0,293	308	308	1,4	230	230	1,4
Folie	(945, 718)	0,769	0,321	82	82	3	35	35	2,9
Katzen	(1289, 720)	1,4	1	1200	1200	5,1	1205	1205	5
Fade 2	(1670, 954)	0,480	1,5	732	732	7	184	184	7

[1] Angaben in Sekunden
[2] Bilder nach Größe aufsteigend sortiert
[3] Die Spalte Revert Filter gibt die Zeit an, in der die Filterung rückgängig gemacht wird

Fazit Zeitmessungen

Technische Hochschule Rosenheim

- Einfach/ Effektiv implementierbar: RLE, Huffman, Filter
- Deflate und PNG ungeeignet wegen schlechter Performance des LZ77-Algorithmus (Sliding Window deutlich vergrößern)

Fazit

Technische Hochschule Rosenheim

- Ergebnisse ungleich Erwartungen
- Filtern verbessert Kompressionsrate
- Filter + Huffman nach den Kriterien am besten
- RLE ist je nach Bild eine sinnvolle Alternative
- Kein Verfahren ist für alle Bilder am besten (unterschiedliche Stärken)